Rank Deficiency Condition of the Multiple View Matrix for Mixed Point and Line Features

Yi Ma, Jana Košcká and Kun Huang

Abstract—Geometric relationships governing multiple images of points and lines and associated algorithms have been studied to a large extent separately in multiple view geometry. In this paper we present a universal rank condition on the so-called multiple view matrix \( M \) comprised of arbitrarily combined point and line features across multiple views. The proposed formulation is shown to be equivalent (but superior) to the multilocus constraints based approach. For the first time, it allows us to carry out global geometric analysis for multiple images, as well as systematically characterize all degenerate configurations, without breaking image sequence into pairwise or triplicate ones. One motivation behind this formulation is that it allows to utilize all incidence conditions that govern all features in all images simultaneously for a consistent recovery of motion and structure from multiple views. Simulation results are presented to validate the multiple view matrix based approach.

Keywords—multiple view matrix, rank condition, mixed features

I. INTRODUCTION

CHARACTERIZATION of the existing geometric constraints has a long history both in computer vision and photogrammetry and has important implications for a variety of applications. The geometric relationships governing observable feature primitives in multiple views provide a starting point from which one can determine the choice of primitives to represent a 3-D scene and consequently formulate and solve the problem of motion and structure recovery from multiple views.

The basic formulation of the geometric constraints governing perspective projections of point features in two views originated in photogrammetry which could be traced back to the beginning of last century [8] and then was revived later in the computer vision community in early eighties [10]. Natural extensions (of theoretical importance and with profound practical implications) had been those considering multiple views and different feature primitives. In the computer vision literature, fundamental and structure independent relationships between image features and camera displacements were first described by the so-called multilinear matching constraints [4], [14], [7]. Most of the previous work focused on the algebraic aspects of these multilinear constraints, along with the algorithms which followed from the same formulation. This line of work culminated recently in publication of two monographs on this topic [6], [2].

The constraints among multiple views and associated algorithms were mostly developed separately for point and line features and for different number of views. Such a development relied on the use of (often difficult to follow) tensorial notation, which was preceded by algebraic elimination of some of the unknowns to render otherwise intrinsically nonlinear relationships as linear ones. A distinguished role in this development was the use of the so-called trilinear constraints and their associated trilinear tensors. Trilinear constraints revealed certain geometric relationships between point and line features among three views [13], [12], [3] and were used extensively for feature matching, point-line transfer to a new view, and motion and structure recovery among three views. In order to apply the trilinear constraints to more than three views, however, one typically resorted to certain cascading scheme [1]. Given that the choice of cascading is by no means unique and many degenerate configurations may occur among the chosen triple-wise views, it was difficult to draw any consistent conclusions on the global geometry for the multiple views altogether.

The main contribution of our work here is the derivation of a new general rank deficiency condition on a formal multiple view matrix \( M \), which combines measurements from multiple views of point and line features. This condition generalizes recently proposed rank deficiency conditions developed separately for point, line features and features on a plane [11]. It certainly complements previous efforts to use both line and point features for structure from motion [9], [5], [13]. The rank condition of the new multiple view matrix \( M \) clearly reveals the relationship among all previously known or even some unknown multilinear constraints. Furthermore, the matrix \( M \) generalizes previously studied trilinear constraints involving mixed point and line features to a multiple view setting, and it allows a geometrically meaningful global analysis of arbitrarily many images with arbitrarily mixed features, with no need to cascade pairwise, triple-wise or quadruple-wise images. Its linear structure directly facilitates feature matching, feature transfer across multiple views and motion and structure recovery. An additional appeal of this approach is the sole use of linear algebraic techniques, with no need to introduce tensorial notation, or projective geometry.

Overview of the paper: Section II we introduction notation used in this paper and basic concepts and equations in the formulation of multiple view geometry. In Section III, we give (without proof) a rank condition on some formal multiple view matrix \( M \), from which all multiple view constraints among points and lines can be instantiated. The geometric interpretation of the rank condition of the matrix \( M \) is given in Section IV. In Section V, we outline ideas how to use the multiple view matrix of mixed features to incorporate all incidence conditions in a scene for a consistent motion and structure recovery. Simulation results in Section VII will demonstrate the benefits of the
proposed approach.

II. MULTIPLE VIEWS OF A POINT ON A LINE

An image \( \mathbf{x}(t) = [x(t), y(t), 1]^T \in \mathbb{R}^3 \) of a point \( p \in \mathbb{R}^3 \), with coordinates \( \mathbf{X} = [X, Y, Z, 1]^T \in \mathbb{R}^4 \) relative to a fixed world coordinate frame, taken by a moving camera satisfies the following relationship:

\[
\lambda(t) \mathbf{x}(t) = A(t) P g(t) \mathbf{X}
\]

where \( \lambda(t) \in \mathbb{R}_+ \) is the (unknown) depth of the point \( p \) relative to the camera frame, \( A(t) \in \mathbb{R}^{3 \times 3} \) is the camera calibration matrix (at time \( t \)), \( P = [I, 0] \in \mathbb{R}^{3 \times 4} \) is the constant projection matrix and \( g(t) \in \mathbb{S} \mathbb{E}(3) \) is the coordinate transformation from the world frame to the camera frame at time \( t \). In the above equation, all \( \mathbf{x}, \mathbf{X} \) and \( g \) are in **homogeneous representation**. Now suppose that \( p \) is lying on a straight line \( L \subset \mathbb{R}^3 \), defined by \( L = \{ \mathbf{Y} | \mathbf{Y} = \mathbf{X} + \alpha \mathbf{v} \} \), where \( \mathbf{v} = [v_1, v_2, v_3]^T \in \mathbb{R}^3 \) is a non-zero vector indicating the direction of the line, and \( \alpha \in \mathbb{R} \). An image \( \mathbf{I}(t) = [a(t), b(t), c(t)]^T \in \mathbb{R}^3 \) of \( L \) taken by the moving camera then satisfies the following equation:

\[
\mathbf{I}(t)^T \mathbf{y}(t) = \mathbf{I}(t)^T A(t) P g(t) \mathbf{Y} = 0
\]

for the image \( \mathbf{y}(t) \) of any point on the line \( L \). In a realistic situation, we usually only obtain "sampled" images of \( \mathbf{x}(t) \) or \( \mathbf{I}(t) \) at some time instances: \( t_1, t_2, \ldots, t_m \). For simplicity we denote

\[
\lambda_i = \lambda(t_i), \quad \mathbf{x}_i = \mathbf{x}(t_i), \quad \mathbf{l}_i = \mathbf{I}(t_i), \quad \Pi_i = A(t_i) P g(t_i).
\]

We then have the following system of equations:

\[
\begin{align*}
\lambda_i \mathbf{x}_i & = \Pi_i \mathbf{X} \quad \lambda_i \mathbf{x}_i = \Pi_i \mathbf{X} \quad (4) \\
\mathbf{l}_i^T \mathbf{x}_i & = \mathbf{l}_i^T \Pi_i \mathbf{Y} = \mathbf{l}_i^T \Pi_i \mathbf{v} = 0 \quad (5)
\end{align*}
\]

for \( i = 1, \ldots, m \). We first observe that, among the unknowns, \( \lambda, \mathbf{X}, \mathbf{Y} \) and \( \mathbf{v} \) all encode information about location of the point \( p \) on the line \( L \), which are not intrinsically available from the images. Hence it is natural to eliminate them from these equations if needed. The remaining relationships would be between \( \mathbf{x}, \mathbf{l} \) and \( \Pi \) only, i.e., between the images and the camera configuration. Of course there are many different, but algebraically equivalent, ways that one can eliminate these unknowns. This has in fact resulted in all kinds of multilinear (or multibjective) constraints that exist in the computer vision literature. They have very much made the study of multiple view geometry seem unnecessarily difficult. We here provide a more systematic way of eliminating all the unknowns and characterizing all the constraints. The resulting set of constraints or conditions would also be **complete**. That is, as we will soon see, any known or even some unknown (multifocal) relationships can be trivially deduced from our results.

III. MULTIPLE VIEW RANK CONDITIONS

Without loss of generality, we may assume that the first camera frame is chosen to be the reference frame. That gives the projection matrices \( \Pi_i, i = 1, \ldots, m \) the general form:

\[
\Pi_i = [I, 0], \quad i = 1, \ldots, m = [R_i, T_i] \in \mathbb{R}^{3 \times 4},
\]

where \( R_i \in \mathbb{R}^{3 \times 3}, i = 2, \ldots, m \) is the first three columns of \( \Pi_i \) and \( T_i \in \mathbb{R}^{3}, i = 2, \ldots, m \) is the fourth column of \( \Pi_i \). Although we have used the suggestive notation \( (R_i, T_i) \) here, they are not necessarily the actual rotation and translation. \( R_i \) could be an arbitrary \( 3 \times 3 \) matrix. Only in the case when the camera is perfectly calibrated does \( R_i \) correspond to actual camera rotation and translation \( T_i \) to translation.

For the \( m \) images \( x_1, \ldots, x_m \) of a point \( p \) on a line \( L \) with its \( m \) images \( l_1, \ldots, l_m \), we define the following set of matrices formally:

\[
\begin{align*}
D_i & = [x_i]_x \in \mathbb{R}^{3 \times 3} \quad \text{or} \quad I_i^T I_i^T \mathbf{Y} = 0 \\
D_i^+ & = [x_i]_x \in \mathbb{R}^{3} \quad \text{or} \quad [l_i]_Y^T \in \mathbb{R}^{3 \times x}
\end{align*}
\]

where the transpose on \([l_i]_Y^T\) is purely stylistic. Then, depending on whether the available (or chosen) measurement from the \( i^{th} \) image is the point feature \( x_i \) or the line feature \( l_i \), the \( D_i \) matrix chooses a corresponding value. That choice is completely independent of the other \( D_j \)'s for \( j \neq i \). The "dual" matrix \( D_i^+ \) can be viewed as the orthogonal supplement to \( D_i \) since for all \( u \in \mathbb{R}^3 \), the row vectors of \([u]_x \) are orthogonal to \( u \). Using the above definition of \( D_i \) and \( D_i^+ \), we now also formally define a **universal multiple view matrix**:

\[
M = \begin{bmatrix}
D_2 R_2 D_2^+ & D_2 T_2 \\
D_3 R_3 D_3^+ & D_3 T_3 \\
\vdots & \vdots \\
D_m R_m D_m^+ & D_m T_m
\end{bmatrix}
\]

Depending on the particular choice for each \( D_i \) or \( D_i^+ \), the dimension of the matrix \( M \) may vary. But no matter what the choice for each individual \( D_i \) or \( D_i^+ \) is, \( M \) will always be a valid matrix of certain dimension. Then after elimination of the unknowns \( \lambda, \mathbf{X}, \mathbf{Y} \) and \( \mathbf{v} \) in the system of equations in (4) and (5), we obtain:

**Theorem 1** (Multiple view rank conditions) Consider a point \( p \) lying on a line \( L \) and their images \( x_1, \ldots, x_m \in \mathbb{R}^3 \) and \( l_1, \ldots, l_m \in \mathbb{R}^3 \) relative to \( m \) camera frames whose relative configuration is given by \((R_i, T_i) \) for \( i = 2, \ldots, m \). Then for any choice of \( D_i \) and \( D_i^+ \) in the definition of the.
multiple view matrix $M$, the rank of the resulting $M$ belongs to and only belongs to the following two cases:
1. If $D_i^1 = [I_i]_{x,y}$ and $D_i = [x_i]_x$ for some $i \geq 2$, then
$$1 \leq \text{rank}(M) \leq 2$$
(8)

2. Otherwise,
$$0 \leq \text{rank}(M) \leq 1.$$  
(9)

A complete proof of this theorem can be found in [11]. Essentially, the above theorem gives a universal description of the incidence condition between a point and a line in terms of their $m$ images seen from $m$ vantage points.

As a result of Theorem 1, any previously known or unknown constraints among multiple images of point or line features are simply certain instantiations of the Theorem 1. It is worth noting that the rank description is far more general and universal than these special constraints, since restricting the constraints to triple-wise views may introduce certain artificial degeneracies.\footnote{For example, some three views may form a degenerate configuration among themselves but no longer so after putting them together with many other views.} In any case, Theorem 1 implies that there would be no further relationship among quadruple-wise views, even in the mixed feature scenario.\footnote{In fact, this is quite expected: While the rank condition geometrically corresponds to the incidence condition that lines intersect at a point and that planes intersect at a line, incidence condition that three-dimensional subspaces intersect at a plane is a void condition in $\mathbb{P}^3$.}

Therefore, quadrilinear constraints and quadrilinear tensors do not really exist. For clarity, we demonstrate using the following examples how to obtain different types of constraints by instantiating $M$:

**Example 1** (Point-point-point constraints) Let us choose $D_i^1 = x_1, D_2 = [x_2]_x, D_3 = [x_3]_x$. Then we get a multiple view matrix:
$$M = \begin{bmatrix} [x_2]_x & R_2 x_1 & [x_2]_x & T_2 \\ [x_3]_x & R_3 x_1 & [x_3]_x & T_3 \end{bmatrix} \in \mathbb{R}^{6 \times 2}. \quad (10)$$

Then rank($M$) $\leq 1$ gives:
$$[[x_2]_x \times R_2 x_1][[x_3]_x \times T_3] - [[x_3]_x \times R_3 x_1][[x_2]_x \times T_2]^T = 0 \in \mathbb{R}^{3 \times 3}. \quad (11)$$

**Example 2** (Line-line-line constraints) Let us choose $D_i^1 = [I_i]_{x,y}, D_2 = [I_2]_x, D_3 = [I_3]_x$. Then we get a multiple view matrix:
$$M = \begin{bmatrix} [I_2]_y & R_2 I_1 & [I_2]_y & T_2 \\ [I_3]_y & R_3 I_1 & [I_3]_y & T_3 \end{bmatrix} \in \mathbb{R}^{2 \times 4}. \quad (12)$$

Then rank($M$) $\leq 1$ gives:
$$[[I_2]_y \times R_2 I_1][[I_3]_y \times T_3] - [[I_3]_y \times R_3 I_1][[I_2]_y \times T_2] = 0 \in \mathbb{R}^3. \quad (13)$$

**Example 3** (Point-line-line constraints) Let us choose $D_i^1 = x_1, D_2 = [I_2]_y, D_3 = [I_3]_y$. Then we get a multiple view matrix:
$$M = \begin{bmatrix} [I_2]_y & R_2 x_1 & [I_2]_y & T_2 \\ [I_3]_y & R_3 x_1 & [I_3]_y & T_3 \end{bmatrix} \in \mathbb{R}^{2 \times 2}. \quad (14)$$

Then rank($M$) $\leq 1$ gives:
$$[[I_2]_y \times R_2 x_1][[I_3]_y \times T_3] - [[I_3]_y \times R_3 x_1][[I_2]_y \times T_2] = 0 \in \mathbb{R}. \quad (15)$$

Then rank($M$) $\leq 2$ implies that all $3 \times 3$ submatrices of $M$ have determinant zero. They are the line-point-point type of constraints on three images. Similarly, other choices of $D_i$ and $D_j$ will give rise to all possible types of constraints among any number of views with point and line features arbitrarily mixed. In fact, other incidence conditions such as all features belong to a plane in $\mathbb{P}^3$ can also be expressed in terms of the same rank condition:

**Corollary 1** (Coplanar features) In the case that all features are from a plane with their coordinates $X$ satisfy $\pi^T X = 0$ for a vector $\pi \in \mathbb{R}^4$. Denote $\pi = [\pi^1, \pi^2]^T$ with $\pi^1 \in \mathbb{R}^3, \pi^2 \in \mathbb{R}$. Then simply apply
$$[\pi^1 D_i^1 \pi^2] \quad (16)$$
to the matrix $M$ in its formal definition (7). The rank condition on $M$ remains the same as Theorem 1.

The rank condition on the new $M$ matrix then implies all constraints among multiple images of planar features, including the homographic constraints studied in [3] (see [11] for details).

**Remark 1** (Features at infinity) In Theorem 1, if the point $p$ and line $L$ are in the plane at infinity $\mathbb{P}_p \setminus \mathbb{P}_L$, the rank condition on the multiple view matrix $M$ is just the same. Hence the rank condition extends to multiple view geometry of the entire projective space $\mathbb{P}_3$, and it does not discriminate Euclidean, affine or projective assumption on the underlying space.

**Remark 2** (Occlusion) If any feature is occluded in a particular image, the corresponding row (or a group of rows) is simply omitted from $M$; or if only the point is occluded but not the entire line(s) on which the point lies, then simply replace the missing image of the point by the corresponding image(s) of the line(s). In either case, the overall rank condition on $M$ remains unaffected. In fact, the rank condition on $M$ gives a very effective criterion to tell whether or not a set of (mixed) features indeed correspond to one or another. If the features are miss-matched, either due to occlusion or errors during establishing correspondence, the rank condition will be violated.

IV. GEOMETRIC INTERPRETATION

For the first time, the multiple view matrix provides a tool which allows us to carry out global geometrical analysis for multiple images simultaneously, without breaking them into pairwise or triple-wise ones. Since there are practically infinitely many possible instantiations of the multiple view matrix for arbitrarily many views, it is impossible to provide a geometric description to each of them. Instead, we are going to discuss one class of them which will give
the reader a clear idea how the rank condition works geometrically. Understanding these cases would be sufficient for the reader to carry out a similar analysis to any other case.

Let us consider multiple view matrices arise from the case 2 in Theorem 1. In this case, we have $0 \leq \text{rank}(M) \leq 1$. So there are only two interesting sub-cases depending on the value of the rank of $M$:

1. $\text{rank}(M) = 1$, and 2. $\text{rank}(M) = 0$. \hspace{1cm} (15)

When the rank of $M$ is 1, it corresponds to the generic cases that, regardless of the particular choice of features in $M$, all these features satisfy the incidence condition, i.e. all the point features (if more than 2 present in $M$) are from a unique 3-D point $p$, lines features (if more than 3 present in $M$) are from a unique 3-D line $L$, and if both point and line features are present, the point $p$ must lie on the line $L$ in 3-D. This is illustrated in Figure 1. But one should be careful if there is not enough point or line features present in $M$? For example, in $M_{pl}$, there is only one point feature $x_1$ present. Still $\text{rank}(M_{pl}) = 1$ means that a line $L$ is uniquely determined by $l_2, \ldots, l_m$ and the point $p$ is consequently determined by the $L$ and its first image $x_1$. On the other hand, if there is only one line features present in some $M$ but more than two point features in $M$, $L$ can then be a family of lines (on a plane in fact) passing through the point $p$. In any case, if a point or line is under-determined in the case $\text{rank}(M) = 1$, it is only because there is not enough data in the given images, not because the configuration is degenerate.

When the rank of $M$ is 0, it means all the entries of $M$ are zeros. It is easy to verify that this corresponds to a set of degenerate cases when the 3-D location of the point or the line cannot be uniquely determined from their multiple images (no matter how many), or the incidence condition between the point $p$ and the line $L$ no longer holds. In these cases, the best we can do is: 1. When there are more than two point features present in $M$, the 3-D location of the point $p$ can be determined up to a line which connects all camera centers (related to these point features); 2. When there are more than three line features are present in $M$, the 3-D location of the line $L$ can be determined up to the plane on which all related camera centers must lie; 3. When both point and line features are present in $M$, we can usually determine the point $p$ up to a line (connecting all camera centers related to the point features) which is lying on the same plane on which the rest of the camera centers (related to the line features) and the line $L$ must lie.

Let us demonstrate this on a concrete example. Suppose the number of views is $m = 6$ and we choose the matrix $M$ to be:

$$M = \begin{bmatrix} l_1^T R_{x_1} & l_2^T T_2 \\ l_1^T R_{x_1} & l_3^T T_3 \\ l_4^T R_{x_1} & l_5^T T_4 \\ [x_5] \times R_0 x_1 & [x_5] \times T_5 \\ [x_6] \times R_0 x_1 & [x_6] \times T_6 \end{bmatrix} \in \mathbb{R}^{9 \times 2}. \hspace{1cm} (16)$$

Geometric configuration of the point and line features corresponding to the condition $\text{rank}(M) = 0$ is illustrated in Figure 2.

Similar geometric analysis can be performed on the case 1 in Theorem 1. One should notice that there are also only two sub-cases there since the rank of $M$ can only be either 2 or 1. Similarly, the upper bound 2 corresponds to generic configurations but the lower bound 1 corresponds to all degenerate ones. For details, the reader can refer to [11].

As a summary of the above discussion, we see that the rank condition without doubt extends previous methods which use multilinear constraints and can only analyze up to three views at a time.\footnote{Analysis using quadrifocal tensors would simply be void.} Since there is yet no systematic way to extend triple-wise analysis to multiple views, the multiple view matrix seems to be a more natural tool for multiple-view analysis. Notice that, from examples in the preceding section, the rank condition simply implies all previously known multilinear constraints, but not vice versa. Hence the use of algebraic equations may introduce certain artificial degeneracy that makes a global analysis much more complicated and sometimes even intractable. On the other hand, the rank condition has no such problem: All the degenerate cases simply correspond to a further drop of rank for the multiple view matrix.
V. MOTION AND STRUCTURE RECOVERY

The unified formulation of the rank condition may allow us to solve the problem of motion and structure recovery from multiple views using both point and line features. There are certain advantages for using point and line features together. Incidence constraints among points and lines can now be explicitly taken into account when a global estimation of motion and structure takes place. To demonstrate how this works, let us consider an image of a cube as shown in Figure 3. For the $j^{th}$ corner $p$, it lies on the three edges $L_i^{j}$, $L_2^{j}$, and $L_3^{j}$, $j = 1, \ldots, 8$. From three images of the cube, we have the multiple view matrix $M^j$ for $p$:

$$M^j = \begin{bmatrix} x_1^j \times R_2 x_1^j & x_2^j \times R_1 x_2^j \\ R_2 x_1^j \times R_3 x_1^j & R_1 x_2^j \times R_3 x_2^j \end{bmatrix} \in \mathbb{R}^{12 \times 2} \tag{17}$$

where $x_i^j$ means the image of the $j^{th}$ corner in the $i^{th}$ view and $1_{ij}^T$ means the image of the $k^{th}$ edge associated to the $j^{th}$ corner in the $i^{th}$ view. It is direct to verify that $\alpha^j = [\lambda_1^j, 1]^T \in \mathbb{R}^2$ is in the kernel of $M^j$. In addition to the multiple images $x_1^j, x_2^j, x_3^j$ of the $k^{th}$ corner $p$, the extra rows associated to the line features $1_{ij}^k, i, k = 1, 2, 3$ also help to determine the depth scale $\lambda_1^j$.

We can already see one advantage about the rank condition: It can simultaneously handle multiple incidence conditions associated to the same feature.\footnote{In fact, any algorithm extracting point feature essentially relies on exploiting local incidence condition on multiple edge features. The structure of the $M$ matrix simply reveals a similar fact within a larger scale.} In principle, using Corollary 1, one can even further take into account that the four vertices and edges on each side are coplanar. Since such incidence conditions between points and lines occur frequently in practice, especially for man-made objects such as buildings and houses, the use of multiple view matrix for mixed features is going to improve the quality of overall reconstruction by explicitly taking into account all the geometric relationships among features of various types and with multiple measurements.

In order to estimate $\alpha^j$ we need to know the matrix $M^j$, i.e. we need to know the motion $(R_2, T_2)$ and $(R_3, T_3)$. From the geometric meaning of $\alpha^j = [\lambda_1^j, 1]^T$, $\alpha^j$ can be solved already if we know only the motion $(R_2, T_2)$ between the first two views, which can be estimated using the standard 8 point algorithm. Knowing $\alpha^j$'s, the equations

$$M^j \alpha^j = 0, \ j = 1, \ldots, 8 \tag{18}$$

become linear in $(R_2, T_2)$ and $(R_3, T_3)$. We can use them to solve for the motions (again). Define the vectors $\tilde{R}_i = [r_1, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}]^T \in \mathbb{R}^9$ and $\tilde{T}_i = T_i \in \mathbb{R}^3$, $i = 2, 3$. It is then equivalent to solve the following equations for $i = 2, 3$:

$$P_i \begin{bmatrix} \tilde{R}_i \\ \tilde{T}_i \end{bmatrix} = \begin{bmatrix} \lambda_1^j [x_i^j] \times [x_i^j]^T & \lambda_1^j [x_i^j] \times [x_i^j]^T \\ \lambda_1^j [x_i^j] \times [x_i^j]^T & \lambda_1^j [x_i^j] \times [x_i^j]^T \end{bmatrix} \begin{bmatrix} \lambda_1^j \end{bmatrix} = 0 \in \mathbb{R}^{18}, \tag{19}$$

where $A \ast B$ is the Kronecker product of $A$ and $B$. In general, if we have more than 6 feature points (here we have 8) or equivalently 12 feature lines, the rank of the matrix $P_i$ is 11 and there is a unique solution to $(\tilde{R}_i, \tilde{T}_i)$. Let $\tilde{T}_i \in \mathbb{R}^3$ and $\tilde{R}_i \in \mathbb{R}^{3 \times 3}$ be the (unique) solution of (19) in matrix form. Such a solution can be obtained numerically as the eigenvector of $P_i$ associated to the smallest singular value. Let $\tilde{R}_i = U_i V_i^T$ be the SVD of $\tilde{R}_i$. Then the solution of (19) in $\mathbb{R}^3 \times \text{SO}(3)$ is given by:

$$T_i = \frac{\text{sign}(\det(U_i V_i^T))}{\sqrt{\det(S_i)}} \tilde{T}_i \in \mathbb{R}^3, \tag{20}$$

$$R_i = \text{sign}(\det(U_i V_i^T)) U_i V_i^T \in \text{SO}(3). \tag{21}$$

We then have the following linear algorithm for motion and structure estimation from three views of a cube:

Algorithm 1 (Motion and structure from mixed features)

Given $m (= 3)$ images $x_1, \ldots, x_m$ of $n (= 8)$ points $p_i$, $i = 1, \ldots, n$ (as the corners of a cube), and the images $1_{ij}^k$, $k = 1, 2, 3$ of the three edges intersecting at $p_i$, estimate the motions $(R_i, T_i)$, $i = 2, \ldots, m$, as follows:

1. Initialization: $s = 0$

(a) Compute $(R_2, T_2)$ using the 8 point algorithm for the first two views [10].

(b) Compute $\alpha^j_2 = [\lambda_2^j/\lambda_1^j, 1]^T$ where $\lambda_2^j$ is the depth of the $j^{th}$ point relative to the first camera frame.

2. Compute $(\tilde{R}_i, \tilde{T}_i)$ as the eigenvector associated to the smallest singular value of $P_i$, $i = 2, \ldots, m$.

3. Compute $(R_i, T_i)$ from (20) and (21) for $i = 2, \ldots, m$.

4. Compute the new $\alpha_{s+1} = \alpha^j$ from (18). Normalize so that $\lambda_1^s = 1$.

5. If $|\alpha_s - \alpha_{s+1}| > \epsilon$, for a pre-specified $\epsilon > 0$, then $s = s + 1$ and goto 2. Else stop.
The camera motion is then \((R_i, T_i), i = 2, \ldots, m\) and the structure of the points (with respect to the first camera frame) is given by the converged depth scalar \(\lambda_j, j = 1, \ldots, n\).

We have a few comments on the proposed algorithm:
1. The reason to set \(\lambda_{i+1} = 1\) is to fix the universal scale. It is equivalent to putting the first point at a relative distance of 1 to the first camera center.
2. Although the algorithm is based on the cube, considers only three views, and utilizes only one type of multiple view matrix, it can be easily generalized to any other objects and arbitrarily many views whenever incidence conditions among a set of point features and line features are present. One may also use the rank conditions on different types of multiple view matrix provided by Theorem 1. The reader may refer to [11] for the case when \(D_j^+\) is chosen to be \(\text{null}(A_j)\).
3. The above algorithm is a straightforward modification of the algorithm proposed for the pure point case [11]. All the measurements of line features directly contribute to the estimation of the camera motion and the structure of the points. Throughout the algorithm, there is no need to initialize or estimate the 3-D structure of lines.

VI. Simulations and Experiments

The simulation parameters are as follows: the camera’s field of view is 90°, image size is 500 × 500, everything is measured in units of the focal length of the camera, and features typically are suited with a depth variation from 100 to 400 units of focal length away from the camera center, i.e., they locate in the truncated pyramid specified by the given field of view and depth variation (see Fig. 4).

Camera motions are specified by their translation and rotation axes. For example, between a pair of frames, the symbol \(XY\) means that the translation is along the X-axis and rotation is along the Y-axis. If \(n\) such symbols are connected by hyphens, it specifies a sequence of consecutive motions. We always choose the amount of total motion such that all feature points will stay in the field of view for all frames. In all simulations, independent Gaussian noise with a standard deviation (std) given in pixels is added to each image point, and each image line is perturbed in a random direction of a random angle with a corresponding std given in degrees.  Error measure for rotation is \(\arccos\left(\frac{\text{tr}(A_j^+R)}{2}\right)\) in degrees where \(\hat{R}\) is an estimate of the true \(R\). Error measure for translation is the angle between \(T\) and \(\hat{T}\) in degrees where \(\hat{T}\) is an estimate of the true \(T\).

We run the algorithm for 1000 trials with the noise level on the point features from 0 pixel to 5 pixels and a corresponding noise level on the line features from 0 degree to 1 degrees. Relative to the given amount of translation, 5 pixels noise is rather high because we do want to compare how all the algorithms perform over a large range of noise levels. The results of the motion estimate errors are given in Fig. 6. The “Point feature only” algorithm is the one for pure point features proposed in [11] which essentially use the multiple view matrix \(M\) in (18) without all the rows associated to the line features; and the “Mixed features” algorithm uses essentially the same \(M\) as in (18). Both algorithms are initialized by the standard 8 point algorithm. The “Mixed features” algorithm gives a significant improvement in all the estimates as a result of the use of both point and line features in the recovery. Also notice that, at a high noise levels, even though the 8 point algorithm gives rather off initialization values, the two iterative algorithms manage to converge back to reasonable estimates. The structure estimate errors show a similar pattern as the errors for motion estimates.

B. Simulations on a random scene

Here we run the algorithm for 500 trials on a randomly chosen scene for each trial. Each random scene is generated
by randomly generating 24 points in the truncated pyramid as shown in Figure 4. They are then connected by 40 randomly chosen lines. The two consecutive X–X and Y–Y rotations with an incremental 10 degrees rotation and the translation is given by the so-called \( T/R \) ratio, which is the ratio between the magnitude of translation \( ||T|| \) and rotation angle \( \theta \) compared at the center of truncated pyramid (see Figure 4). In following simulations, the ratio is 2. Comparing to the motion with previous simulations on the cubes, here the amount of translation is much bigger. That results in improved estimates for translation as shown by Figure 7. And the structure estimates are similarly improved (data not shown). The improvement is expected.

Fig. 6. Motion estimates error versus level of noises. “Motion x–y” means the estimate for the motion between image frames x and y. Since the results are very much similar, we only plotted “Motion 1–2” and “Motion 1–4”.

Fig. 7. Motion estimates error versus level of noises for random scenes. “motion x–y” means the estimate for the motion between image frames x and y.

VII. DISCUSSIONS AND CONCLUSIONS

This paper has proposed a unified paradigm to synthesize results and experiences in the study of multiple views of point and line features. It is shown that all relationships among multiple images of a point on a line are captured through a single rank condition on certain multiple view matrix. All previously known constraints on multiple images simply become its instantiations. To a large extent, this condition simplifies and unifies multiple view geometry. In addition, we can now carry out meaningful geometric analysis for arbitrarily many images altogether without going through a pairwise, triple-wise and quadruple-wise analysis. Compared to conventional multiple view analysis based on trifocal tensors, the multiple view matrix based approach clearly separates meaningful geometric degeneracy from degeneracy which may be artificially introduced by the use of algebraic equations to describe constraints. In particular, as shown in this paper, any configurations which cause a further drop of rank in the multiple view matrix correspond exactly to certain geometric degeneracy. Combined with previous results on pure point, line and planar feature cases [11], results on the mixed feature case make a coherent set of theory and algorithms for multiple view geometry.

The proposed approach aims to provide people a new perspective to multiple view geometry. It will certainly have impact on both theoretical analysis and algorithm development. The linear algorithms given in this paper and others [11] only show a straight-forward way of using the rank condition. There are many other ways to improve them: 1. One can use any other error measures in the 3-D space or in the 2-D image to recover the motion and structure optimally subject to the rank condition; 2. Slight variations of the rank condition can handle occlusions; 3. Better numerical methods need to be investigated on how to impose the rank condition; and so on. While we are still in the process of investigating the full potential of this new approach, there are plenty of reasons for us to believe that we are still at a very early stage of understanding the full extent of multiple view geometry: either its theory or its practice.

REFERENCES


