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

Yi Ma, Jana Košecká 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 multilinear (or multifocal) 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 triple-wise sets of views. The additional advantage 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 can 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. A distinguished role in that development was the use of the so-called trilinear constraints and their associated trilinear tensors. The initial formulation of the constraints between three views of point and line features is due to [13]. Further developments and extensions to multiple views relied on the use of tensorial notation, where the multilinear constraints were obtained by algebraic elimination of some of the unknowns to render otherwise intrinsically nonlinear relationships as linear ones. Trilinear constraints revealed certain geometric relationships between point and line features among three views [12], [5] and were used extensively for feature matching, point-line transfer to a new view and motion and structure recovery from three views. In order to apply the trilinear constraints to more than three views, one had to typically resort to a cascading scheme as in [1]. Given that the choice of cascading is by no means unique and many degenerate configurations may occur among the chosen triplets, it was difficult to draw consistent conclusions on the global geometry for the multiple views altogether.

The main contribution of our work 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 points, lines and planar features [11]. Our treatment completes previous efforts to use both line and point features for structure from motion recovery from multiple views [9], [5], [13]. Furthermore 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 introduces notation used in this paper as well as basic concepts and equations for 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

Yi Ma and Kun Huang are with the Electrical & Computer Engineering Department of University of Illinois at Urbana-Champaign. Jana Košecká is with the Department of Computer Science of George Mason University. This work is supported by U.S. Army Research Office under Contract DAAD19-00-1-0466; the National Science Foundation KDI initiative under subcontract SBC-MIT-5710000330; the UIUC ECE dept. and George Mason Univ. CS dept. startup fund.

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 VI 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 $p \in \mathbb{E}^3$, 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:

$$\lambda(t)\mathbf{x}(t) = A(t)Pg(t)\mathbf{X} \tag{1}$$

where $\lambda(t) \in \mathbb{R}_+$ is the (unknown) depth of the point p relative to the camera frame, $A(t) \in SL(3)$ is the camera calibration matrix (at time t), $P = [I, 0] \in \mathbb{R}^{3\times 4}$ is the constant projection matrix and $g(t) \in SE(3)$ 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 $L \subset \mathbb{E}^3$, defined by $L = \{\mathbf{Y} \mid \mathbf{Y} = \mathbf{X} + \alpha v\}$, where $v = [v_1, v_2, v_3, 0]^T \in \mathbb{R}^4$ is a non-zero vector indicating the direction of the line, and $\alpha \in \mathbb{R}$. An image $\mathbf{l}(t) = [a(t), b(t), c(t)]^T \in \mathbb{R}^3$ of L taken by the moving camera then satisfies the following equation:

$$\mathbf{l}(t)^T\mathbf{y}(t) = \mathbf{l}(t)^T A(t)Pg(t)\mathbf{Y} = 0 \tag{2}$$

for the image $\mathbf{y}(t)$ of any point on the line L.[1] In a realistic situation, we usually only obtain "sampled" images of $\mathbf{x}(t)$ or $\mathbf{l}(t)$ at some time instances: $t_1, t_2, \ldots, t_m$. For simplicity we denote

$$\lambda_i = \lambda(t_i),\ \mathbf{x}_i = \mathbf{x}(t_i),\ \mathbf{l}_i = \mathbf{l}(t_i),\ \Pi_i = A(t_i)Pg(t_i). \tag{3}$$

We then have the following system of equations:

$$\lambda_i \mathbf{x}_i = \Pi_i \mathbf{X} \tag{4}$$
$$\mathbf{l}_i^T \mathbf{x}_i = \mathbf{l}_i^T \Pi_i \mathbf{Y} = \mathbf{l}_i^T \Pi_i v = 0 \tag{5}$$

for $i = 1, \ldots, m$. We first observe that the unknowns, λ, $\mathbf{X}$, $\mathbf{Y}$ and v, which encode the information about location of the point p or the line L in $\mathbb{R}^3$ are not intrinsically available from the images. Hence it is natural to eliminate them from these equations first. The remaining relationships would be between $\mathbf{x}, \mathbf{l}$ and Π 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 different kinds (or forms) of multilinear (or multifocal) constraints that exist in the computer vision literature. We here introduce a more *systematic* way of eliminating *all* the above unknowns that results in a *complete* set of conditions and a clear characterization of *all* constraints. Consequently, as we will soon see, all previously known and even some unknown relationships can be trivially deduced from our results.

[1] So defined $\mathbf{l}$ is in fact the vector orthogonal to the plane spanned by the images of points on the line. Strictly speaking, $\mathbf{l}$ should be called the "coimage" of the line.

III. Multiple view rank conditions

Without loss of generality, we may assume that the first camera frame is chosen to be the reference frame.[2] That gives the projection matrices $\Pi_i, i = 1, \ldots, m$ the general form:

$$\Pi_1 = [I, 0], \quad \ldots, \quad \Pi_m = [R_m, T_m] \quad \in \mathbb{R}^{3\times 4}, \tag{6}$$

where $R_i \in \mathbb{R}^{3\times 3}, i = 2, \ldots, m$ is the first three columns of Π_i and $T_i \in \mathbb{R}^3, i = 2, \ldots, m$ is the fourth column of Π_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 3×3 matrix. Only in the case when the camera is perfectly calibrated does R_i correspond to the actual camera rotation and T_i to the translation.

For the m images $\mathbf{x}_1, \ldots, \mathbf{x}_m$ of a point p on a line L with its m images $\mathbf{l}_1, \ldots, \mathbf{l}_m$, we define the following set of matrices formally:[3]

$$\begin{array}{rclcl} D_i & \doteq & [\mathbf{x}_i]_\times \in \mathbb{R}^{3\times 3} & \text{or} & \mathbf{l}_i^T \in \mathbb{R}^3, \\ D_i^\perp & \doteq & \mathbf{x}_i \in \mathbb{R}^3 & \text{or} & [\mathbf{l}_i]_\times^T \in \mathbb{R}^{3\times 3}, \end{array}$$

where the transpose on $[\mathbf{l}_i]_\times^T$ is purely stylistic. Then, depending on whether the available (or chosen) measurement from the i^{th} image is the point feature $\mathbf{x}_i$ or the line feature $\mathbf{l}_i$, 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 $D_i^\perp$ can be viewed as the *orthogonal supplement* to D_i since for all $u \in \mathbb{R}^3$, the row vectors of $[u]_\times$ are orthogonal to u.[4] Using the above definition of D_i and $D_i^\perp$, we now also formally define a *universal multiple view matrix:*

$$M \doteq \begin{bmatrix} D_2R_2D_1^\perp & D_2T_2 \\ D_3R_3D_1^\perp & D_3T_3 \\ \vdots & \vdots \\ D_mR_mD_1^\perp & D_mT_m \end{bmatrix}. \tag{7}$$

Depending on the particular choice for each D_i or $D_1^\perp$, the dimension of the matrix M may vary. But no matter what the choice for each individual D_i or $D_1^\perp$ is, M will always be a valid matrix of certain dimension. Then after elimination of the unknowns λ, $\mathbf{X}$, $\mathbf{Y}$ and 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 $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^3$ and $\mathbf{l}_1, \ldots, \mathbf{l}_m \in \mathbb{R}^3$ 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 $D_1^\perp$ in the definition of the*

[2] Depending on the context, the reference frame could be either a Euclidean, affine or projective reference frame. Without loss of generality the projection matrix for the first image becomes the standard projection matrix $[I,\ 0] \in \mathbb{R}^{3\times 4}$.

[3] For a three dimensional vector $u \in \mathbb{R}^3$, we use $[u]_\times \in \mathbb{R}^{3\times 3}$ to denote the skew symmetric matrix associated to u such that for any vector $v \in \mathbb{R}^3$, we have: $[u]_\times v = u \times v$. Notice that $[u]_\times$ is skew-symmetric, i.e. $[u]_\times^T = -[u]_\times$.

[4] In fact, there are many equivalent matrix representations for D_i and $D_i^\perp$. We choose $[\mathbf{x}_i]_\times$ and $[\mathbf{l}_i]_\times^T$ here because they are the simplest forms representing the orthogonal subspaces of $\mathbf{x}_i$ and $\mathbf{l}_i$ and also linear in $\mathbf{x}_i$ and $\mathbf{l}_i$ respectively.

multiple view matrix M, the rank of the resulting M belongs to and only belongs to the following two cases:

1. If $D_1^\perp = [\mathbf{l}_1]_\times^T$ and $D_i = [\mathbf{x}_i]_\times$ for some $i \geq 2$, then

$$\boxed{1 \leq rank(M) \leq 2.} \tag{8}$$

2. Otherwise,

$$\boxed{0 \leq rank(M) \leq 1.} \tag{9}$$

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 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 condition is far more general and universal than these special constraints, since restricting the constraints to triple-wise views may introduce certain artificial degeneracies.[5] Theorem 1 also implies that there would be *no* further relationship among quadruple-wise views, even in the mixed feature scenario.[6] Therefore, quadrilinear constraints and quadrilinear tensors *do not* really exist. To make a connection with existing work, we demonstrate by the following examples how to obtain different types of constraints by instantiating M.

Example 1 (Epipolar constraints) Let us choose $D_1^\perp = \mathbf{x}_1$ and $D_2 = [\mathbf{x}_2]_\times$, then $M = [[\mathbf{x}_2]_\times R_2\mathbf{x}_1 \;\; [\mathbf{x}_2]_\times T_2] \in \mathbb{R}^{3\times 2}$. rank$(M) = 1$ is exactly equivalent to the *epipolar constraint* $\mathbf{x}_2^T[T_2]_\times R_2\mathbf{x}_1 = 0$ between two views.

Example 2 (Trilinear constraints) Let us choose $D_1^\perp = \mathbf{x}_1, D_2 = [\mathbf{x}_2]_\times, D_3 = [\mathbf{x}_3]_\times$. Then we get a multiple view matrix:

$$M = \begin{bmatrix} [\mathbf{x}_2]_\times R_2\mathbf{x}_1 & [\mathbf{x}_2]_\times T_2 \\ [\mathbf{x}_3]_\times R_3\mathbf{x}_1 & [\mathbf{x}_3]_\times T_3 \end{bmatrix} \in \mathbb{R}^{6\times 2}. \tag{10}$$

Then rank condition rank$(M) \leq 1$ gives:

$$[[\mathbf{x}_2]_\times R_2\mathbf{x}_1][[\mathbf{x}_3]_\times T_3]^T - [[\mathbf{x}_3]_\times R_3\mathbf{x}_1][[\mathbf{x}_2]_\times T_2]^T = 0 \in \mathbb{R}^{3\times 3}.$$

This is the well known trilinear constraint among point features. Similarly, if we choose $D_1^\perp$ and D_i to be line features only, we get an M matrix of size 2×4, its rank condition is exactly the trilinear constraint for lines.

Example 3 (Point-line-line constraints) Let us choose $D_1^\perp = \mathbf{x}_1, D_2 = \mathbf{l}_2^T, D_3 = \mathbf{l}_3^T$. Then we get a multiple view matrix:

$$M = \begin{bmatrix} \mathbf{l}_2^T R_2\mathbf{x}_1 & \mathbf{l}_2^T T_2 \\ \mathbf{l}_3^T R_3\mathbf{x}_1 & \mathbf{l}_3^T T_3 \end{bmatrix} \in \mathbb{R}^{2\times 2}. \tag{11}$$

Then rank$(M) \leq 1$ condition :

$$[\mathbf{l}_2^T R_2\mathbf{x}_1][\mathbf{l}_3^T T_3] - [\mathbf{l}_3^T R_3\mathbf{x}_1][\mathbf{l}_2^T T_2] = 0 \quad \in \mathbb{R}.$$

gives the trilinear constraint in a mixed feature case.

Example 4 (Line-point-point constraints) Let us choose $D_1^\perp = [\mathbf{l}_1]_\times^T, D_2 = [\mathbf{x}_2]_\times, D_3 = [\mathbf{x}_3]_\times$. Then we get a multiple view matrix:

$$M = \begin{bmatrix} [\mathbf{x}_2]_\times R_2[\mathbf{l}_1]_\times^T & [\mathbf{x}_2]_\times T_2 \\ [\mathbf{x}_3]_\times R_3[\mathbf{l}_1]_\times^T & [\mathbf{x}_3]_\times T_3 \end{bmatrix} \in \mathbb{R}^{6\times 4}. \tag{12}$$

Then rank$(M) \leq 2$ implies that all 3×3 sub-matrices of M have determinant zero. They are the line-point-point type of constraints on three images.

Similarly, other choices of D_i and $D_1^\perp$ 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 belonging to a plane in $\mathbb{R}^3$ can also be expressed in terms of the same rank condition:

Corollary 1 (Planar features and homography) *Suppose that all features are in a plane and coordinates $\mathbf{X}$ of any point on it satisfy the equation $\pi^T\mathbf{X} = 0$ for some vector $\pi \in \mathbb{R}^4$. Denote $\pi = [\pi^1, \pi^2]$ with $\pi^1 \in \mathbb{R}^3, \pi^2 \in \mathbb{R}$. Then simply append the matrix*

$$\begin{bmatrix} \pi^1 D_1^\perp & \pi^2 \end{bmatrix} \tag{13}$$

to the matrix M in its formal definition (7). The rank condition on the new M remains exactly the same as Theorem 1.

The rank condition on the new M matrix then implies *all* constraints among multiple images of these planar features, including a special constraint previously studied as *homography* [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 $\mathbb{P}^3 \setminus \mathbb{E}^3$, 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 against 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

[5] For example, some three views may form a degenerate configuration but no longer so after putting them together with many other views.

[6] 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 $\mathbb{E}^3$.

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 arising from the case 2 in Theorem 1. In this case, we have $0 \leq \operatorname{rank}(M) \leq 1$. So there are only two interesting sub-cases depending on the value of the rank of M:

$$1.\ \operatorname{rank}(M) = 1, \quad \text{and} \quad 2.\ \operatorname{rank}(M) = 0. \tag{14}$$

The case of $\operatorname{rank}(M) = 1$ corresponds to the generic situations, when regardless of the particular choice of features in M, all these features satisfy the incidence condition. For example all the point features (if projections in more then 2 views are present in M) are form a unique 3-D point p, lines features (if in more than 3 views present in M) are from a unique 3-D line L. If both point and line features are present, the point p then must lie on the line L in 3-D. This is illustrated in Figure 1. But what happens in the

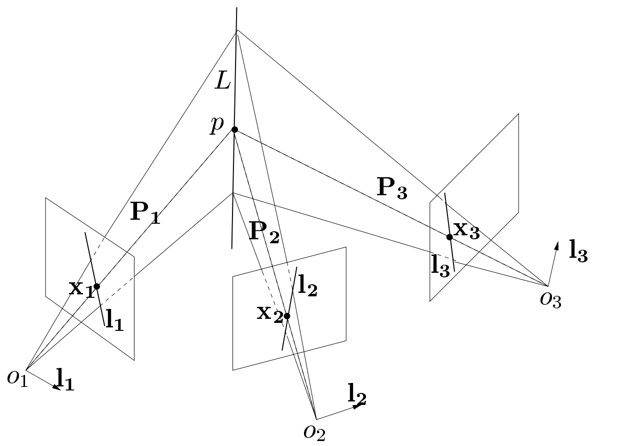

Fig. 1. Generic configuration for the case $\operatorname{rank}(M) = 1$. Planes extended from the images $\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_3}$ intersect at one line L in 3-D. Lines extended from the images $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ intersect at one point p. p must lie on L.

mixed case, where there is not enough point or line features present in M, but we have some mixture of them? Consider for example matrix M with only one point feature $\mathbf{x}_1$ present and the remaining are the line features. Still $\operatorname{rank}(M) = 1$ means that a line L is uniquely determined by $\mathbf{l}_2, \ldots, \mathbf{l}_m$ and the point p is consequently determined by the L and its first image $\mathbf{x}_1$. 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 $\operatorname{rank}(M) = 1$, it is only because there is not enough data in the give images, not because the configuration is degenerate.

The case when the $\operatorname{rank}(M) = 0$ 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:

$$M = \begin{bmatrix} \mathbf{l}_2^T R_2 \mathbf{x}_1 & \mathbf{l}_2^T T_2 \\ \mathbf{l}_3^T R_3 \mathbf{x}_1 & \mathbf{l}_3^T T_3 \\ \mathbf{l}_4^T R_4 \mathbf{x}_1 & \mathbf{l}_4^T T_4 \\ [\mathbf{x}_5]_\times R_5 \mathbf{x}_1 & [\mathbf{x}_5]_\times T_5 \\ [\mathbf{x}_6]_\times R_6 \mathbf{x}_1 & [\mathbf{x}_6]_\times T_6 \end{bmatrix} \in \mathbb{R}^{9 \times 2}. \tag{15}$$

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

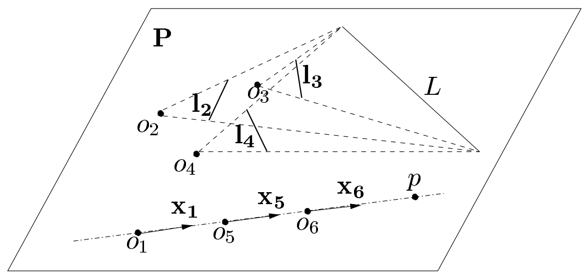

Fig. 2. A degenerate geometric configuration for the case $\operatorname{rank}(M) = 0$: a point-line-line-line-point-point scenario. From the given rank condition, the line L could be any where on the plane spanned by all the camera centers; the point p could be any where on the line through o_1, o_5, o_6.

Similar geometric analysis can be performed in the case 1 of Theorem 1. One should notice that there are 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 multifocal tensors but can only analyze up to three views at a time.[7] 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.

[7]Analysis using quadrifocal tensors would simply be void.

V. Motion and structure recovery

The unified formulation of contraints in terms of the rank condition allows 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 better than existing methods, let us consider an image of a cube as shown in Figure 3. For the j^{th} corner p^j, it is the inter-

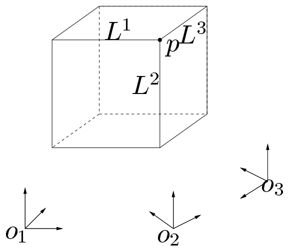

Fig. 3. A standard cube. The three edges L^1, L^2, L^3 intersect at the corner p. The three coordinates indicate that three images are taken at these vantage points.

section of the three edges L^{1j}, L^{2j} and L^{3j}, $j = 1, \dots, 8$. From three images of the cube, we have the multiple view matrix M^j associated to p^j:

$$M^j = \begin{bmatrix} [\mathbf{x}_2^j]_\times R_2 \mathbf{x}_1^j & [\mathbf{x}_2^j]_\times T_2 \\ \mathbf{l}_2^{1jT} R_2 \mathbf{x}_1^j & \mathbf{l}_2^{1jT} T_2 \\ \mathbf{l}_2^{2jT} R_2 \mathbf{x}_1^j & \mathbf{l}_2^{2jT} T_2 \\ \mathbf{l}_2^{3jT} R_2 \mathbf{x}_1^j & \mathbf{l}_2^{3jT} T_2 \\ [\mathbf{x}_3^j]_\times R_3 \mathbf{x}_1^j & [\mathbf{x}_3^j]_\times T_3 \\ \mathbf{l}_3^{1jT} R_3 \mathbf{x}_1^j & \mathbf{l}_3^{1jT} T_3 \\ \mathbf{l}_3^{2jT} R_3 \mathbf{x}_1^j & \mathbf{l}_3^{2jT} T_3 \\ \mathbf{l}_3^{3jT} R_3 \mathbf{x}_1^j & \mathbf{l}_3^{3jT} T_3 \end{bmatrix} \in \mathbb{R}^{12 \times 2} \tag{16}$$

where $\mathbf{x}_i^j \in \mathbb{R}^3$ means the image of the j^{th} corner in the i^{th} view and $\mathbf{l}_i^{kj} \in \mathbb{R}^3$ means the image of the k^{th} edge associated to the j^{th} corner in the i^{th} view. Theorem 1 says $(M) = 1$. One can 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 $\mathbf{x}_1^j, \mathbf{x}_2^j, \mathbf{x}_3^j$ of the j^{th} corner p^j itself, the extra rows associated to the line features $\mathbf{l}_i^{kj}, i, k = 1, 2, 3$ also help to determine the depth scale λ_1^j.

We can already see one advantage of the rank condition: It can simultaneously handle multiple incidence conditions associated to the same feature.[8] In principle, using Corollary 1, one can further take into account that the four vertices and edges on each face are coplanar. Since such incidence conditions among 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 incidence relationships among features of various types.

In order to estimate α^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 $\alpha^j = [\lambda_1^j, 1]^T$, α^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 α^j's, the equations

$$M^j \alpha^j = 0, j = 1, \dots, 8 \tag{17}$$

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 $\vec{R}_i = [r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}]^T \in \mathbb{R}^9$ and $\vec{T}_i = T_i \in \mathbb{R}^3$, $i = 2, 3$. It is then equivalent to solve the following equations for $i = 2, 3$:

$$P_i \begin{bmatrix} \vec{R}_i \\ \vec{T}_i \end{bmatrix} = \begin{bmatrix} \lambda_1^1 [\mathbf{x}_i^1]_\times * \mathbf{x}_1^{1T} & [\mathbf{x}_i^1]_\times \\ \lambda_1^1 \mathbf{l}_i^{11T} * \mathbf{x}_1^{1T} & \mathbf{l}_i^{11T} \\ \lambda_1^1 \mathbf{l}_i^{21T} * \mathbf{x}_1^{1T} & \mathbf{l}_i^{21T} \\ \lambda_1^1 \mathbf{l}_i^{31T} * \mathbf{x}_1^{1T} & \mathbf{l}_i^{31T} \\ \vdots & \vdots \\ \lambda_1^8 [\mathbf{x}_i^8]_\times * \mathbf{x}_1^{8T} & [\mathbf{x}_i^8]_\times \\ \lambda_1^8 \mathbf{l}_i^{18T} * \mathbf{x}_1^{8T} & \mathbf{l}_i^{18T} \\ \lambda_1^8 \mathbf{l}_i^{28T} * \mathbf{x}_1^{8T} & \mathbf{l}_i^{28T} \\ \lambda_1^8 \mathbf{l}_i^{38T} * \mathbf{x}_1^{8T} & \mathbf{l}_i^{38T} \end{bmatrix} \begin{bmatrix} \vec{R}_i \\ \vec{T}_i \end{bmatrix} = 0 \in \mathbb{R}^{48}, \tag{18}$$

where $A * B$ 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 $(\vec{R}_i, \vec{T}_i)$.

Let $\tilde{T}_i \in \mathbb{R}^3$ and $\tilde{R}_i \in \mathbb{R}^{3\times3}$ be the (unique) solution of (18) in matrix form. Such a solution can be obtained numerically as the eigenvector of P_i associated to the smallest singular value. Let $\tilde{R}_i = U_i S_i V_i^T$ be the SVD of $\tilde{R}_i$. Then the solution of (18) in $\mathbb{R}^3 \times SO(3)$ is given by:

$$T_i = \frac{\mathrm{sign}(\det(U_i V_i^T))}{\sqrt[3]{\det(S_i)}} \tilde{T}_i \in \mathbb{R}^3, \tag{19}$$

$$R_i = \mathrm{sign}(\det(U_i V_i^T))\, U_i V_i^T \in SO(3). \tag{20}$$

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 $\mathbf{x}_1^j, \dots, \mathbf{x}_m^j$ of $n(=8)$ points p^j, $j = 1, \dots, n$ (as the corners of a cube), and the images $\mathbf{l}_i^{kj}, k = 1, 2, 3$ of the three edges intersecting at p^j, estimate the motions (R_i, T_i), $i = 2, \dots, m$ 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 $\alpha_s^j = [\lambda_1^j / \lambda_1^1, 1]^T$ where λ_1^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, $i = 2, \dots, m$.
3. Compute (R_i, T_i) from (19) and (20) for $i = 2, \dots, m$.
4. Compute the new $\alpha_{s+1}^j = \alpha^j$ from (17). Normalize so that $\lambda_{1,s+1}^1 = 1$.
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 the converged $(R_i, T_i), i = 2, \dots, m$ and the structure of the points (with respect to the first camera frame) is the converged depth scalar $\lambda_1^j, j = 1, \dots, n$.

We have a few comments on the proposed algorithm:
1. The reason to set $\lambda_{1,s+1}^1 = 1$ 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.

[8] 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.

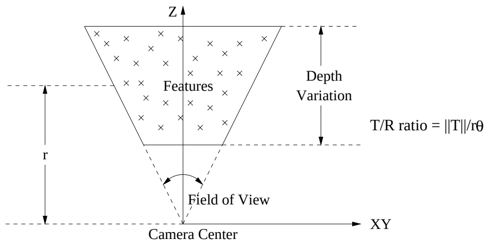

Fig. 4. Simulation setup

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 $D_1^{\perp}$ is chosen to be $[\mathbf{l}_1]_{\times}^T$.
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

We carried out extensive simulations to determine the performance of the proposed algorithms as the noise in the measurements and the number of features and views vary. The simulation parameters are as follows: the camera's field of view is 90^o, image size is 500×500, everything is measured in units of focal length of the camera, and features typically are suited with a depth variation is 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 Figure 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. [9] Error measure for rotation is $\arccos\left(\frac{tr(R\tilde{R}^T)-1}{2}\right)$ in degrees where $\tilde{R}$ is an estimate of the true R. Error measure for translation is the angle between T and $\tilde{T}$ in degrees where $\tilde{T}$ is an estimate of the true T. Error measure for the scene structure is the percentage of $\|\alpha - \tilde{\alpha}\|/\|\alpha\|$ where $\tilde{\alpha}$ is an estimate of the true α.

A. Simulations on a structured scene

In this simulation, we apply the algorithm to a scene which consists of (four) cubes only. Cubes are good objects to test the algorithm since the relationships between their corners and edges are easily defined and they represent a fundamental structure of many objects in real-life. The length of the four cube edges are 30, 40, 60 and 80 units of focal length, respectively. The cubes are arranged such that the depth of their corners ranges from 75 to 350 units of focal length. The three motions (relative to the first view) are an XX-motion with -10 degrees rotation and 20 units translation, a YY-motion with 10 degrees rotation and 20 units translation and another YY-motion with -10 degrees rotation and 20 units translation, as shown in Figure 5.

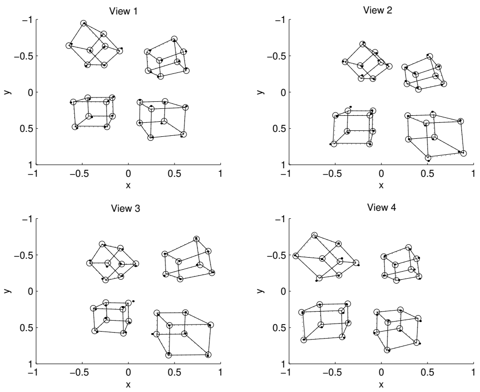

Fig. 5. Four views of four 3-D cubes in (normalized) image coordinates. The circle and the dotted lines are the original images, the dots and the solid lines are the noisy observations under 5 pixels noise on point features and 0.5 degrees noise on line features.

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 to 1 degree. 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 Figure 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 (17) without all the rows associated to the line features; and the "Mixed features" algorithm uses essentially the same M as in (17). 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. The scene comprises of 24 randomly generated points in the truncated pyramid as shown in Figure 4. They are then connected by 40 randomly chosen lines. The two consecutive XX-motion and YY-motion 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

[9]Since line features can be measured more reliably than point features, lower noise level is added to them in simulations.

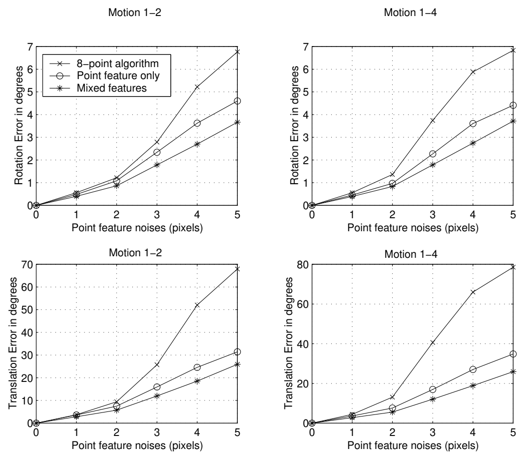

Fig. 6. Motion estimates error versus level of noises. "Motion x-y" means the estimate for the motion between image frames x and y. Since the results are very much similar, we only plotted "Motion 1-2" and "Motion 1-4".

θ 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. This results in improved estimates for translation as shown by Figure 7. And the structure estimates are similarly improved (data not shown) as expected. See [11] for details.

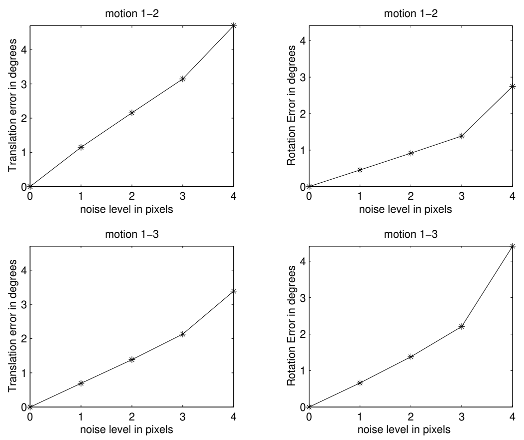

Fig. 7. Motion estimates error versus level of noises for random scenes. "motion x-y" means the estimate for the motion between image frames x and y.

VII. Discussions and conclusions

This paper has proposed a unified paradigm which synthesizes 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 a so-called 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 or quadruple-wise analysis. Compared to conventional multiple view analysis based on trifocal tensors, the multiple view matrix based approach clearly separates meaningful geometric degeneracies from degeneracies which may be artificially introduced by the use of algebraic equations or tensors. In particular, as shown in this paper, any configuration which causes a further drop of rank in the multiple view matrix *exactly* corresponds to certain global geometric degeneracy. Combined with previous results on point, line and planar features [11], results in this paper give rise to a coherent but simple geometric theory that is genuine for multiple images.

The proposed approach aims to provide 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 (hence naive) way of using the rank condition. There are many other ways to improve them: 1. One can use better error measures in the 2-D image to recover the motion and structure optimally subject to the rank condition; 2. Slight change of the algorithm may handle occlusions; 3. Better numerical methods should 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

[1] S. Avidan and A. Shashua. Novel view synthesis by cascading trilinear tensors. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 4(4), 1998.

[2] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *Geometry of Multiple Images.* The MIT Press, 2001.

[3] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

[4] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between N images. In *Proceedings of Fifth International Conference on Computer Vision*, pages 951–6, Cambridge, MA, USA, 1995. IEEE Comput. Soc. Press.

[5] R. Hartley. Lines and points in three views - a unified approach. In *Proceedings of 1994 Image Understanding Workshop*, pages 1006–1016, Monterey, CA USA, 1994. OMNIPRESS.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge, 2000.

[7] A. Heyden and K. Åström. Algebraic properties of multilinear constraints. *Mathematical Methods in Applied Sciences*, 20(13):1135–62, 1997.

[8] E. Kruppa. Zur ermittlung eines objecktes aus zwei perspektiven mit innerer orientierung. *Sitz.-Ber.Akad.Wiss., Math.Naturw., Kl.Abt.IIa, 122:1939-1948*, 1913.

[9] Y. Liu, T. Huang, and O. Faugeras. Determination of camera location from 2-D to 3-D line and point correspondences. *IEEE Transactions on PAMI*, pages 28–37, 1990.

[10] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[11] Y. Ma, K. Huang, R. Vidal, J. Košecká, and S. Sastry. Rank conditions of multiple view matrix in multiple view geometry. *techincal report*, June 18, 2001.

[12] A. Shashua. Trilinearity in visual recognition by alignment. In *the Proceedings of ECCV, Volume I*, pages 479–484. Springer-Verlag, 1994.

[13] M. Spetsakis and Y. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4(3):171–184, 1990.

[14] B. Triggs. Matching constraints and the joint image. In *Proceedings of Fifth International Conference on Computer Vision*, pages 338–43, Cambridge, MA, USA, 1995. IEEE Comput. Soc. Press.