Introduction to Multiview Rank Conditions and their Applications: A Review. *

Jana Košecká ◇ Yi Ma †
◇Department of Computer Science, George Mason University
†Electrical & Computer Engineering Department, University of Illinois at Urbana-Champaign
e-mail:kosecka@cs.gmu.edu, yima@uiuc.edu

ABSTRACT

Understanding the representations of 3D scenes as encoded in multiple views taken by a camera from different vantage points is central to many tasks in image and video analysis. These tasks range from recovering the camera motion, 3D structure of the scene and detection and characterization of multiple motions in video. We will demonstrate that the natural representations of a 3D scene in 2D images is in terms of the incidence relations among different geometric primitives, which can be concisely characterized by rank conditions of multiview matrices. The proposed rank conditions capture all existing independent multilinear constraints and enable truly global geometric analysis of the multiple views comprised of different geometric features. In addition to the analysis, we present natural factorization based linear algorithms for structure and motion recovery, image transfer and matching across multiple views applicable in both calibrated and uncalibrated setting. We will demonstrate the approach experimentally on a problem of multiframe structure and motion recovery using point and line features and their incidence relations.

1 INTRODUCTION

Analysis, alignment and characterization of the content of multiple images of a scene captured by a camera from different vantage points is central to many tasks in video and image analysis. Most of the past research on video coding, compression and multimedia applications involving video originated in image processing community and focused predominantly on 2D image processing techniques to encode the information in the image stream. On the other hand large amount of work in computer vision community has been devoted to the problems of recovery of 3D models of the environments from multiple views. The applications range from building 3D models from photographs (generally referred to as image rendering techniques) with applications to architectural preservation, computer graphics or special effects in movie industry, augmented reality systems, human computer interaction or object level modeling to retail purposes.

*The work is supported by NSF grant IIS-0118732.

It is inevitable that it is the 3D structure of the environment which gives rise to the video and photography content and hence should be exploited in the analysis. It is therefore central to understand and study how is the 3D structure encoded in multiple views of the scene and what is the relationship between the projections of the 3D world and camera displacements. Considering scenarios where the observed motion of the objects in the scene and/or camera is rigid, the relationships are to a large extent characterized by various geometric constraints between observable geometric primitives and rigid body motion.

Characterization of the existing geometric constraints has a long history both in computer vision and photogrammetry. The basic formulations of the intrinsic geometric constraints governing *perspective* projections of point features in two views originated in photogrammetry and were later revived the computer vision community in early eighties [1]. Natural extensions of relationships between two views is to consider multiple views and different feature primitives. In the computer vision literature, fundamental and structure independent relationships between image features and camera displacements were characterized by the so-called multilinear matching constraints [2, 3, 4, 5]. These geometric relationships were used extensively for feature matching, point-line transfer to a new view and motion and structure recovery from three views [6, 7]. This line of work culminated recently in publication of two monographs on this topic [8, 9].

In this paper we present new characterization of the existing multiview constraints in terms of rank conditions of appropriate multiple view matrices introduced in [10, 11]. We start first by introducing the rank conditions among multiple views of point and line features separately. We will demonstrate in an intuitive way that the rank conditions of these multiview matrices captures the relationships among all previously known multilinear constraints and generalizes previously studied trilinear constraints involving mixed point and line features to a multiview setting. As we will see the linear formulation of the problem will give rise to natural algorithms for geometric feature matching, feature transfer across

multiple views and motion and structure recovery. In order to demonstrate the wide applicability of the framework in the limited space, we will focus more on the geometric intuition behind the formulation and algorithmic aspects, while omitting the detailed proofs of the statements. These can be found in [10, 11]. In the last section we present the applicability of the multiple view matrix of mixed features for a consistent motion and structure recovery which properly incorporate all the incidence conditions in a scene and outline conceptual algorithms for image matching and feature transfer to a novel view.

2 MULTIPLE VIEWS OF POINTS AND LINES

First we will introduce basic notation and concepts used through out the paper. An image $\boldsymbol{x}(t) = [x(t), y(t), 1]^T \in \mathbb{R}^3$ of a point $p \in \mathbb{E}^3$, with coordinates $\boldsymbol{X} = [X, Y, Z, 1]^T \in \mathbb{R}^4$ relative to a fixed world coordinate frame, taken by a moving camera satisfies the following relationship

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

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 $\boldsymbol{x}, \boldsymbol{X}$ and g are in *homogeneous representation*. A straight line $L \subset \mathbb{E}^3$, defined by $L = \{\boldsymbol{X} \mid \boldsymbol{X} = \boldsymbol{X}_0 + \alpha v\}$, 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 $\boldsymbol{l}(t) = [a(t), b(t), c(t)]^T \in \mathbb{R}^3$ of L taken by the moving camera then satisfies the following equation

$$\boldsymbol{l}(t)^T\boldsymbol{x}(t) = \boldsymbol{l}(t)^T A(t)Pg(t)\boldsymbol{X} = 0 \tag{2}$$

for the image $\boldsymbol{x}(t)$ of any point on the line L.[1] In a practice we usually have available images of $\boldsymbol{x}(t)$ or $\boldsymbol{l}(t)$ at some time instances: $t_1, t_2, \ldots, t_m$, which we denote

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

Observing set of points and/or lines in multiple views gives rise to the following system of equations

$$\lambda_i\boldsymbol{x}_i = \Pi_i\boldsymbol{X} = [R_i, T_i]\boldsymbol{X}, \tag{3}$$

$$\boldsymbol{l}_i^T\boldsymbol{x}_i = \boldsymbol{l}_i^T\Pi_i\boldsymbol{X}_0 = \boldsymbol{l}_i^T\Pi_i v = 0 \tag{4}$$

for $i = 1, \ldots, m$. The suggestive notation $\Pi_i = [R_i, T_i]$ here, does not necessarily correspond to 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. Observe that the unknowns, λ, $\boldsymbol{X}$ 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. By algebraically eliminating some of the unknowns from the above equations, the remaining relationships would be between $\boldsymbol{x}, \boldsymbol{l}$ and Π only, i.e. between the images and the camera configuration. These relationships are referred to as intrinsic and provide a starting point for recovering the remaining information from images. The elimination step is rather straightforward in the two view case, where the unknown scales can be eliminated by multiplying both sides of the equation by the vector $T_2 \times \boldsymbol{x}_2 = \widehat{T_2}\boldsymbol{x}_2$[2]

$$\lambda_2\boldsymbol{x}_2 = R_2\lambda_1\boldsymbol{x}_1 + T_2 \ \Rightarrow\ \boldsymbol{x}_2^T\widehat{T_2}R_2\boldsymbol{x}_1 = 0 \tag{5}$$

Note that this yields an implicit constraint, so-called *epipolar constraint*, on the camera displacement (R_2, T_2) between two views, which can be consequently used for displacement recovery [1]. Geometric interpretation of the two view constraint is in Figure 1.

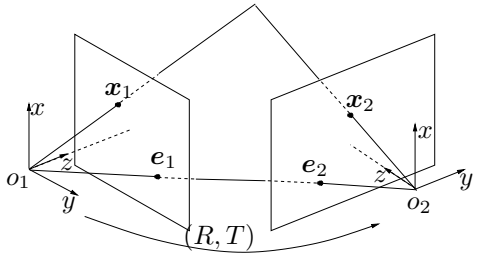

Figure 1: Two projections $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^3$ of a 3-D point p from two vantage points. The relative Euclidean transformation between the two vantage points is given by $(R, T) \in SE(3)$. The intersection of the line (o_1, o_2) with each image plane is the so-called *epipole*, that is $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ respectively.

However in the multiview case, there are many different, but algebraically equivalent, ways of eliminating the unknowns and hence characterizing these constraints. We here present a unified and concise way of characterizing the existing constraints in terms incidence relations among different geometric primitives and the rank conditions of their associated multiview matrices.

3 RANK CONDITIONS AND INCIDENCE RELATIONS

Our previous work [12] has shown that, multiple images of points, lines or planes are universally governed by certain rank conditions. Such conditions not only concisely capture geometric constraints among multiple images, but also are the key to reconstruction of the camera motion and scene structure. This line of work has allowed us to realize a very important principle about multiple view geometry

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

[2] As usual, for a vector $u \in \mathbb{R}^3$, we use $\widehat{u}$ to denote the skew symmetric matrix such that $\widehat{u}v = u \times v$ for all $v \in \mathbb{R}^3$.

To a large extent, multiple view geometry is about studying how can the incidence relations (among points, lines, and planes etc.) in 3-D space be expressed and exploited computationally in multiple 2-D image measurements.

As we will demonstrate the *rank conditions* of the appropriate multiview matrices turn out to be the correct tool for this purpose. Since the incidence relationships are invariant to change of viewpoint or camera calibration, and they can be effectively verified in images or be given as modeling conditions in practice, such knowledge can be and should be exploited if a consistent reconstruction is sought.

In this section, we briefly review a few classic rank conditions in multiple view geometry and their corresponding geometric intuition. We will not provide any proof for these results since they can be found in our previous paper [12]. Instead, we try to make the results compact here so that the reader can grasp the essence of this new formulation in the shortest time.

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:

$$\Pi_1 = [I, 0], \quad \Pi_2 = [R_2, T_2], \quad \ldots, \quad \Pi_m = [R_m, T_m],$$

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.

3.1 Multiple images of a point

For the multiple images $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ of a point p, as shown in Figure 2, it is necessary and sufficient for the

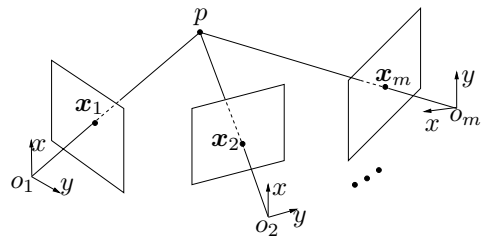

Figure 2: Lines extended from the multiple images $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ intersect at one point p in 3-D.

following so-called *multiple view matrix* M_p

$$M_p \doteq \begin{bmatrix} \widehat{\boldsymbol{x}_2} R_2 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2} T_2 \\ \widehat{\boldsymbol{x}_3} R_3 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_3} T_3 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m} R_m \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 2} \quad (7)$$

to satisfy the rank condition

$$\text{rank}(M_p) = 1. \quad (8)$$

The rank value drops to 0 if and only if all the camera centers $o_1, o_2, \ldots, o_m$ and the point p lie on the same straight line (the so-called rectilinear motion). Notice that for M_p to be rank-deficient, it is necessary for any pair of vectors $\widehat{\boldsymbol{x}}_i T_i, \widehat{\boldsymbol{x}}_i R_i \boldsymbol{x}_1$ to be linearly dependent. This gives us the well-known bilinear epipolar constraints $\boldsymbol{x}_i^T \widehat{T}_i R_i \boldsymbol{x}_1 = 0$ between the i^{th} and 1^{st} view. Hence, the constraint $\text{rank}(M_p) = 1$ consistently generalizes the epipolar constraint for 2 views to arbitrary m views. The constraints among more then two views come from additional linear dependencies between 'rows' of M_p. In order to characterize them we exploit the following linear algebraic fact: Given *non-zero* vectors $a_1, \ldots, a_n, b_1, \ldots, b_n \in \mathbb{R}^3$, the following matrix is rank deficient:

$$\begin{bmatrix} a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{bmatrix} \in \mathbb{R}^{3n\times 2} \quad (9)$$

if and only if $a_i b_j^T - b_i a_j^T = 0$ for all $i, j = 1, \ldots, n$. Applying the fact directly to the matrix M_p we obtain we obtain the well known trilinear constraint

$$\widehat{\boldsymbol{x}}_i (T_i \boldsymbol{x}_1^T R_j^T - R_i \boldsymbol{x}_1 T_j^T) \widehat{\boldsymbol{x}_j} = 0. \quad (10)$$

Hence, the rank condition on matrix M_p captures all trilinear relationships between the i^{th}, j^{th} and 1^{st} views. Notice that the multiple view matrix M_p being rank-deficient is equivalent to all its 2×2 minors having zero determinant. Since the 2×2 minors of M_p involve three images only, we may safely conclude that there are in fact *no additional* independent relationships among four views.

3.2 Multiple images of a line

For multiple images $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_m$ of a line L, as shown in Figure 3, it is necessary and sufficient for the

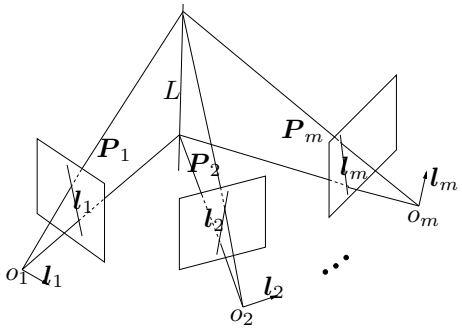

Figure 3: Planes extended from the images $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_m$ intersect at one line L in 3-D.

following multiple view matrix M_l

$$M_l \doteq \begin{bmatrix} \boldsymbol{l}_2^T R_2 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_3^T T_3 \\ \vdots & \vdots \\ \boldsymbol{l}_m^T R_m \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{(m-1)\times 4} \quad (11)$$

to satisfy the rank condition

$$\text{rank}(M_l) = 1. \tag{12}$$

The rank value drops to 0 if and only if all the planes $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_m$ coincide, or equivalently, all the camera centers and the line are coplanar. The rank condition directly implies all trilinear constraints among m images of the line. To see this more explicitly, notice that for $\text{rank}(M_l) = 1$, it is necessary for any pair of row vectors of M_l to be linearly dependent. This gives us the well-known trilinear constraints

$$\boldsymbol{l}_j^T T_j \boldsymbol{l}_i^T R_i \widehat{\boldsymbol{l}_1} - \boldsymbol{l}_i^T T_i \boldsymbol{l}_j^T R_j \widehat{\boldsymbol{l}_1} = 0 \tag{13}$$

among the 1^{st}, i^{th} and j^{th} images. Hence the constraint $\text{rank}(M_l) = 1$ is a generalization of the trilinear constraint (for 3 views) to arbitrary m views. Note that when $m = 3$ it is equivalent to the trilinear constraint for lines, except for some rare degenerate cases, where for example $\boldsymbol{l}_i^T T_i = 0$ for some i.

3.3 Multiple images of intersecting lines

For a set of lines $L^1, L^2, \ldots, L^m$ which intersect at point p and their images in multiple views, as shown in Figure 4, it is necessary and sufficient for the following

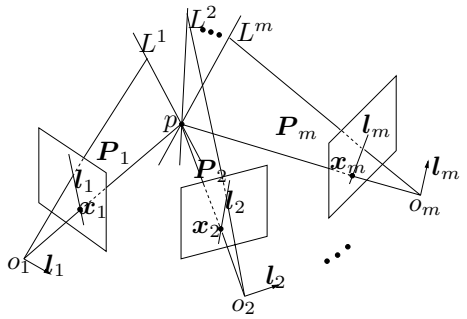

Figure 4: Planes extended from the image lines $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_m$ intersect at one point p in 3-D.

multiple view matrix M_l

$$M_l \doteq \begin{bmatrix} \boldsymbol{l}_2^T R_2 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_3^T T_3 \\ \vdots & \vdots \\ \boldsymbol{l}_m^T R_m \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{(m-1)\times 4} \tag{14}$$

to satisfy the rank condition

$$\text{rank}(M_l) = 2. \tag{15}$$

It is interesting to notice that the rank of M_l will drop back to 1 if the family of lines $L^1, L^2, \ldots, L^m$ happen to be the same line L in 3-D; and the rank will further drop to 0 if the family of planes $\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_m$ happen to collapse into one. Hence a change in the rank value indeed corresponds to a qualitative change in the types of 3-D incidence relations among the involved multiple images.

The interplay between points and lines also gives rise to some interesting rank conditions on mixed versions of the multiple view matrix. For instance, suppose that in the reference view you observe a point $\boldsymbol{x}_1$ and in the remaining views the lines $\boldsymbol{l}_2, \ldots, \boldsymbol{l}_m$ incident to that point, such that in each view $\boldsymbol{l}_i^T \boldsymbol{x}_i = 0$ for $i = 2, \ldots m$. The associated rank condition for this case can be derived from the rank condition on M_l and the following matrix

$$M_{pl} \doteq \begin{bmatrix} \boldsymbol{l}_2^T R_2 \boldsymbol{x}_1 & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3 \boldsymbol{x}_1 & \boldsymbol{l}_3^T T_3 \\ \vdots & \vdots \\ \boldsymbol{l}_m^T R_m \boldsymbol{x}_1 & \boldsymbol{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{(m-1)\times 2} \tag{16}$$

then satisfies the rank condition

$$\text{rank}(M_{pl}) = 1. \tag{17}$$

For the rank condition to be satisfied, it is again necessary as in the point case, that all 2×2 minors of M_{pl} be zero. This yields the following constraints among arbitrary ith, jth view with the first view as a reference

$$(\boldsymbol{l}_i^T R_i \boldsymbol{x}_1)(\boldsymbol{l}_j^T T_j) - (\boldsymbol{l}_j^T R_j \boldsymbol{x}_1)(\boldsymbol{l}_i^T T_i) = 0 \quad \in \mathbb{R}.$$

This gives the trilinear *point-line-line* constraint. Note that the incidence relationship is relaxed in this case since the lines in the subsequent views do not have to correspond to each other as long as they intersect at the same point. We will demonstrate the use of these types of constraints for structure and motion recovery in the following section. In principle, one can arbitrarily decide to choose a point feature or a line feature in each image and the resulting multiple view matrix always obeys certain rank condition. A general law for this is given in [12].

3.4 Multiple images of a planar scene

Another case commonly encountered in practical situations is when the set of points is restricted to lie on a plane in $\mathbb{R}^3$. The plane can be described as $\pi^T \boldsymbol{X} = 0$ for a vector $\pi = [\pi^1, \pi^2] \in \mathbb{R}^4$ with $\pi^1 \in \mathbb{R}^3, \pi^2 \in \mathbb{R}$. Then for multiple images of a point p or a line L on the plane, as shown in Figure 5, it is necessary and sufficient for the following multiple view matrices M_p and M_l

$$M_p = \begin{bmatrix} \widehat{\boldsymbol{x}_2} R_2 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2} T_2 \\ \widehat{\boldsymbol{x}_3} R_3 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_3} T_3 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m} R_m \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m} T_m \\ \pi^1 \boldsymbol{x}_1 & \pi^2 \end{bmatrix}, \; M_l = \begin{bmatrix} \boldsymbol{l}_2^T R_2 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_3^T T_3 \\ \vdots & \vdots \\ \boldsymbol{l}_m^T R_m \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_m^T T_m \\ \pi^1 \widehat{\boldsymbol{l}_1} & \pi^2 \end{bmatrix}$$

to satisfy the rank condition

$$\text{rank}(M_p) = 1, \quad \text{rank}(M_l) = 1. \tag{18}$$

That is, with the extra rows $\begin{bmatrix} \pi^1 \boldsymbol{x}_1 & \pi^2 \end{bmatrix}$ or $\begin{bmatrix} \pi^1 \widehat{\boldsymbol{l}_1} & \pi^2 \end{bmatrix}$ appended to regular M_p and M_l, respectively, the rank

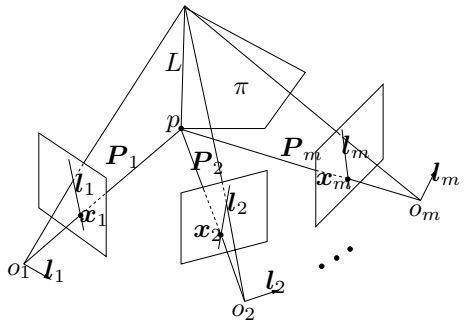

Figure 5: Planes extended from the image lines $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_m$ and the plane π intersect at one line L; Lines extended from the images $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ and the plane π intersect at one point p.

conditions on $M_p(M_l)$ remain the same. Again, the rank value drops to 0 if and only if all the planes $\pi, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_m$ collapse into one.

The rank conditions on the augmented M_p and M_l imply some extra equations (by considering minors of the sub-matrix consisting of the i^{th} group of three rows of M_p or M_l and its last three rows)

$$\begin{cases} \widehat{\boldsymbol{x}}_i \left(R_i - \frac{1}{\pi^2} T_i \pi^1\right) \boldsymbol{x}_1 &= 0, \\ \boldsymbol{l}_i^T \left(R_i - \frac{1}{\pi^2} T_i \pi^1\right) \widehat{\boldsymbol{l}_1} &= 0, \end{cases} \tag{19}$$

for $i = 2, \ldots, m$. These are nothing but the *homography* between the i^{th} and the 1^{st} images of the plane π. By now, we can summarize a few characteristics about this matrix rank approach:

1. The multiple view matrix rank captures geometric (and algebraic) constraints among multiple images in a global fashion, and with it, we no longer need to break an image set or sequence into pair-wise or triple-wise ones;

2. Different rank values of a multiple view matrix directly correspond to qualitatively different types of 3-D incidence relations among multiple images, and hence, a drop of rank value always implies occurrence of degenerate configuration.

3. The universality of the rank conditions suggests the possibility of utilizing all incidence relations among all images of all features in a unified fashion for 3-D reconstruction, and as we will see in the next section, they directly imply a linear factorization algorithm for multiple view reconstruction.

4 MOTION AND STRUCTURE RECOVERY

The unified formulation of the rank condition enable us to solve the problem of motion and structure recovery from multiple views using both point and line features. 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 conceptually how this works consider a sequence of 8 widely separated views of a desk scene as shown in Figure 7.

For some of the corner points p, we can specify the incidence relationship of certain line segments. For example the front corner of the box in the foreground has three lines incident to it and the one on the back box only two lines. We depict them in the image in terms of line segments. Suppose that each point j has k incident edge $L^{1j} \ldots L^{kj}$, for $j = 1, \ldots, n$. From m images of the scene the multiple view matrix M^j for each point p has the following form

$$M^j = \begin{bmatrix} \widehat{\boldsymbol{x}_2^j} R_2 \boldsymbol{x}_1^j & \widehat{\boldsymbol{x}_2^j} T_2 \\ \boldsymbol{l}_2^{1jT} R_2 \boldsymbol{x}_1^j & \boldsymbol{l}_2^{1jT} T_2 \\ \vdots & \vdots \\ \boldsymbol{l}_2^{kjT} R_2 \boldsymbol{x}_1^j & \boldsymbol{l}_2^{kjT} T_2 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m^j} R_m \boldsymbol{x}_1^j & \widehat{\boldsymbol{x}_m^j} T_m \\ \boldsymbol{l}_m^{1jT} R_m \boldsymbol{x}_1^j & \boldsymbol{l}_m^{1jT} T_m \\ \vdots & \vdots \\ \boldsymbol{l}_m^{kjT} R_m \boldsymbol{x}_1^j & \boldsymbol{l}_m^{kjT} T_m \end{bmatrix} \in \mathbb{R}^{(m-1)(k+3) \times 2} \tag{20}$$

where $\boldsymbol{x}_i^j$ means the image of the j^{th} corner in the i^{th} view and $\boldsymbol{l}_i^{kj}$ means the image of the k^{th} edge associated to the j^{th} corner in the i^{th} view. Note that one can easily 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 $\boldsymbol{x}_1^j, \ldots \boldsymbol{x}_m^j$ of the j^{th} corner p itself, the extra rows associated to the line features $\boldsymbol{l}_i^{kj}, i = 2, \ldots m; k = 1, \ldots, k$ also help to determine the depth scale λ_1^j. We can already see one advantage of

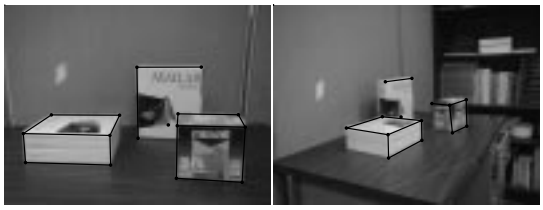

Figure 6: 1^{st} and 7^{th} frame of the test sequence.

the rank condition: It can simultaneously handle multiple incidence conditions associated to the same feature;[3] In principle, using the coplanar rank conditions (18), one can further take into account that the some of the 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

[3] 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.

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 α^j for each point we need to know the matrix M^j, i.e. we need to know the motions (R_i, T_i) for $i = 2, \ldots, m$. From the geometric meaning of $\alpha^j = [\lambda_1^j, 1]^T$ we can initialize α^j's if we know only the motion (R_2, T_2) between the first two views. The two view displacement can be estimated using the standard 8 point algorithm [1]. Knowing α^j's note that each row now becomes linear in R_i, T_i. For example for $i = 2$ we have $\lambda_1^j \widehat{\boldsymbol{x}_2^j} R_2 \boldsymbol{x}_1^j + \widehat{\boldsymbol{x}_2^j} T_2 = 0$. Since all the equations

$$M^j \alpha^j = 0, \quad j = 1, 2, \ldots, n \tag{21}$$

become linear in (R_i, T_i), we can select the appropriate ones 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 $\vec{T}_i = T_i \in \mathbb{R}^3$, $i = 2, \ldots, m$. It is then equivalent to solve the following equations for $i = 2, \ldots, m$:

$$P_i \begin{bmatrix} \vec{R}_i \\ \vec{T}_i \end{bmatrix} = \begin{bmatrix} \lambda_1^1 \widehat{\boldsymbol{x}_i^1} * \boldsymbol{x}_1^{1T} & \widehat{\boldsymbol{x}_i^1} \\ \lambda_1^1 \boldsymbol{l}_i^{11T} * \boldsymbol{x}_1^{1T} & \boldsymbol{l}_i^{11T} \\ \vdots & \\ \lambda_1^1 \boldsymbol{l}_i^{k1T} * \boldsymbol{x}_1^{1T} & \boldsymbol{l}_i^{k1T} \\ \vdots & \vdots \\ \lambda_1^n \widehat{\boldsymbol{x}_i^n} * \boldsymbol{x}_1^{nT} & \widehat{\boldsymbol{x}_i^n} \\ \lambda_1^n \boldsymbol{l}_i^{1nT} * \boldsymbol{x}_1^{nT} & \boldsymbol{l}_i^{1nT} \\ \vdots & \\ \lambda_1^n \boldsymbol{l}_i^{knT} * \boldsymbol{x}_1^{nT} & \boldsymbol{l}_i^{knT} \end{bmatrix} \begin{bmatrix} \vec{R}_i \\ \vec{T}_i \end{bmatrix} = 0 \in \mathbb{R}^{n(3+k)}, \tag{22}$$

where $A * B$ is the *Kronecker product* of A and B. In general, if we have rich enough set of features such that the rank of the matrix P_i is at least 11, there is a unique solution to $(\vec{R}_i, \vec{T}_i)$.

Let $\tilde{T}_i \in \mathbb{R}^3$ and $\tilde{R}_i \in \mathbb{R}^{3\times 3}$ be the (unique) solution of (22) 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 (22) in $\mathbb{R}^3 \times SO(3)$ is given by:

$$T_i = \frac{\text{sign}(\det(U_i V_i^T))}{\sqrt[3]{\det(S_i)}} \tilde{T}_i \in \mathbb{R}^3, \tag{23}$$

$$R_i = \text{sign}(\det(U_i V_i^T))\, U_i V_i^T \in SO(3). \tag{24}$$

We then have the following linear algorithm:

Algorithm 1 (Multiple view factorization) Given $m(\geq 3)$ images $\boldsymbol{x}_1^j, \ldots, \boldsymbol{x}_m^j$ of $n(\geq 8)$ points p^j, $j = 1, \ldots, n$ (as the corners of a cube), and the images $\boldsymbol{l}_i^{kj}, k = 1, 2, 3$ of the three edges intersecting at p^j, 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 [1].
   (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, $i = 2, \ldots, m$.
3. Compute (R_i, T_i) from (23) and (24) for $i = 2, \ldots, m$.
4. Compute the new $\alpha_{s+1}^j = \alpha^j$ from (21). 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 $(R_i, T_i), i = 2, \ldots, m$ and the structure of the points (with respect to the first camera frame) is given by the converged depth scalar $\lambda_1^j, j = 1, \ldots, n$.

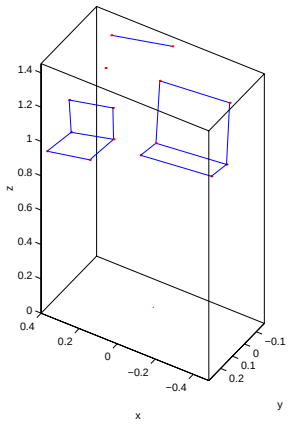

Figure 7: Recovered structure of the scene as seen from a novel viewpoint. Only the features visible in the first frame are visualized.

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.
2. Although the algorithm utilizes only one type of incidence relationships between points and lines captured by the multiple view matrix, it can be easily generalized to include additional incidence relationships or planar restrictions.
3. Although the algorithm is given for the calibrated case, the first two steps the factorization algorithm work also for the uncalibrated case: in the initialization (R_2, T_2) can simply be the canonical decomposition of the fundamental matrix; the remaining (R_i, T_i)'s computed in step 2 will differ from the true ones by the same projective transformation.

4.1 Correspondence

In the following section we will demonstrate how the rank condition can be used for feature matching. We outline the test for the point case only, however the same technique can be applied in an analogous way to other multiview matrices. Notice that in the point case $M_p \in \mathbb{R}^{3(m-1)\times 2}$ being rank-deficient is equivalent to the determinant of $M_p^T M_p \in \mathbb{R}^{2\times 2}$ being zero

$$\det(M_p^T M_p) = 0. \tag{25}$$

where $M_p^T M_p$ is a function of the projection matrix Π and images $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$. If Π is known and we would like to test if given m vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathbb{R}^3$ indeed satisfy all the constraints that m images of a single 3-D pre-image should, we only need to test if the above determinant is zero. A more numerically robust algorithm is outlined below.

Algorithm 2 (Multiple view matching test: point) *Suppose the projection matrix Π associated to m camera frames are given. Then for given vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathbb{R}^3$,*

1. *Compute the matrix $M_p \in \mathbb{R}^{3(m-1)\times 2}$ as in (7);*
2. *Compute second eigenvalue λ_2 of $M_p^T M_p$;*
3. *If $\lambda_2 \leq \epsilon$ for some pre-fixed threshold, the m image vectors match.*

Similar matching test can be developed for the line case. In general, in order to test correspondence using the above algorithm, we need to know the projection matrix Π, which represents the multiview camera configuration. Fortunately, in order to recover Π, we only need few corresponding points and other techniques allow us to do so. Once the initial motion estimate is obtained, the recovered Π can be used to establish more correspondences. Hence the algorithm can be viewed as a natural multiview extension of the commonly used RANSAC based matching algorithm for two view geometry.

4.2 Image transfer

When the two views are widely separated it is quite common that certain features become occluded in some views, while still remain visible in the others. Alternatively we would like to have some means how to predict the location of a feature without explicitly computing the scene structure and back-projecting the 3D entity into an image. This task is often referred as image transfer and can be also naturally accomplished by exploiting the rank condition. Without loss of generality, suppose we know $m-1$ images $\boldsymbol{x}_2, \ldots, \boldsymbol{x}_m \in \mathbb{R}^3$, of a point in space, and we want to determine its 1^{st} image $\boldsymbol{x}_1$, based on known transformations R_i, T_i, $i = 2, \ldots, m$. Observe the structure of the point multiview matrix M_p and let us define the following matrix

$$N_p' \doteq \begin{bmatrix} \widehat{\boldsymbol{x}_2} R_2 & \widehat{\boldsymbol{x}_2} T_2 \\ \widehat{\boldsymbol{x}_2} R_m & \widehat{\boldsymbol{x}_2} T_2 \\ & \vdots \\ \widehat{\boldsymbol{x}_m} R_m & \widehat{\boldsymbol{x}_m} T_2 \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 4}. \tag{26}$$

Note that due to the rank deficiency of matrix M_p associated with the point matrix we have

$$N_p' \begin{bmatrix} \boldsymbol{x}_1 \\ \lambda \end{bmatrix} = 0. \tag{27}$$

Hence the vector $[\boldsymbol{x}_1^T, \lambda]^T$ is in the null space of N_p', which can be robustly computed using singular value decomposition. The algorithm is summarized below:

Algorithm 3 (Image transfer: point) *Suppose the projection matrix Π associated to m camera frames are given. Then for given vectors $\boldsymbol{x}_2, \ldots, \boldsymbol{x}_m \in \mathbb{R}^3$,*

1. *Compute the matrix $N_p' \in \mathbb{R}^{3(m-1)\times 4}$ as in (26);*
2. *Perform SVD such that $N_p' = USV^T$ and denote v_4 to be the 4^{th} column of V;*
3. *Let $\boldsymbol{x}_1$ be the vector formed by the last 3 elements of v_4 and then normalize $\boldsymbol{x}_1$ to have that last coordinate 1.*

In a parallel way we can develop the relationship for image transfer of a line to a new view, based on the rank condition of the matrix associated with the line. Define matrix

$$N_l' \doteq \begin{bmatrix} \boldsymbol{l}_2^T R_2 & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3 & \boldsymbol{l}_3^T T_3 \\ & \vdots \\ \boldsymbol{l}_m^T R_m & \boldsymbol{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{(m-1)\times 4}. \tag{28}$$

It is easy to see that in general we have $\text{rank}(N_l') = 2$ since $N_l' \boldsymbol{X} = 0$ if and only if $\boldsymbol{X} \in \text{span}(\boldsymbol{X}_o, v)$ where $\boldsymbol{X}_o$ and v were defined in Section 2. Hence $\boldsymbol{l}_1^T \tilde{v} = 0$ where $\tilde{(\cdot)}$ is taking the vector formed by first three elements. So we can perform SVD on N_l' to obtain USV^T. Since the last two columns of V, denoted v_3 and v_4, are linearly independent and $N_l' v_3 = N_l' v_4 = 0$. Thus $\boldsymbol{l}_1 = k\tilde{v}_2 \times \tilde{v}_4$ for some $k \neq 0$. This yields following practical line transfer algorithm

Algorithm 4 (Image transfer: line) *Suppose the projection matrix Π associated to m camera frames are given. Then for given vectors $\boldsymbol{l}_2, \ldots, \boldsymbol{l}_m \in \mathbb{R}^3$,*

1. *Compute the matrix $N_l' \in \mathbb{R}^{(m-1)*4}$ as in (28);*
2. *Perform SVD such that $N_l' = USV^T$, such that $N_l' = USV^T$, where $U \in \mathbb{R}^{(m-1)\times(m-1)}, S \in \mathbb{R}^{(m-1)\times 4}$ and $V \in \mathbb{R}^{4\times 4}$;*

3. *Denote v_3 and v_4 to be the 3^{rd} and 4^{th} column of V respectively; set $\tilde{v}_3$ to be the vector formed by the first three elements of v_3 and $\tilde{v}_4$ be the vector formed by the first 3 elements of v_4;*

4. *Let $\boldsymbol{l}_1 = \tilde{v}_3 \times \tilde{v}_4$ and normalize it.*

Figure 8: In the figure (8^{th} frame) on the right we overlay two vertical lines and two points which were transfered from the 4^{th} frame (left) where they were clearly visible.

5 Discussions and conclusions

This paper reviewed a unified paradigm recently proposed by the authors which synthesizes results and experiences in the study of multiple views of point, lines and planes. It is shown that all geometric relationships among multiple images are captured through a single rank condition on certain multiple view matrix. To a large extent, this matrix rank approach simplifies and unifies multiple view geometry. In addition, we can now carry out meaningful global geometric analysis for many images without going through a pairwise, triple-wise, or quadruple-wise relationships. 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 describing the constraints. 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 geometric degeneracy.

The proposed approach will certainly have impact on both theoretical analysis and algorithm development. The linear algorithms given in this paper and others [10] only show a straight-forward way of using the rank condition. But our simulation and experimental results have already shown much better performance then the extant "projective factorization" algorithm [13], and in fact, the performance has in many cases emulated nonlinear algorithms based on "bundle adjustment". Recent work has also shown that it is possible to generalize this matrix rank approach to study trajectory triangulation, curve features, and even dynamical scenes [14]. The full potential of this approach is yet to be investigated.

References

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

[2] O. Faugeras and B. Mourrain, "On the geometry and algebra of the point and line correspondences between N images," in *International Conference on Computer Vision*, 1995, pp. 951–6.

[3] B. Triggs, "Matching constraints and the joint image," in *International Conference on Computer Vision*, June 1995.

[4] A. Heyden and K. Åström, "Algebraic properties of multilinear constraints," *Mathematical Methods in Applied Sciences*, vol. 20, no. 13, pp. 1135–62, 1997.

[5] M. E. Spetsakis and Y. Alloimonos, "A multi-frame approach to visual motion perception," *Intl. Journal of Computer Vision*, vol. 16, no. 3, pp. 245–255, 1991.

[6] R. I. Hartley, "Lines and points in three views - a unified approach," in *Image Understanding Workshop*, 1994, pp. 1006–1016.

[7] S. Avidan and A. Shashua, "Novel view synthesis in tensor space," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 1034 – 1040.

[8] R. Harley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

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

[10] Y. Ma, K. Huang, R. Vidal, J. Kosecka, and S. Sastry, "Rank conditions of the multiple view matrix," in *CSL Tech. Report, University of Illinois Urbana Champain*. UILU-ENG 01-2214, 2001.

[11] Y. Ma, J. Kosecka, and K. Huang, "Rank deficiency condition of the multiple view matrix for mixed point and line features," in *Asian Conference on Computer Vision*, 2002.

[12] Yi Ma, R. Vidal, K. Huang, J. Kosecka, and S. Sastry, "Rank conditions in multiview geometry and its applications," *(submitted to) International Journal of Computer Vision*, 2002.

[13] P. Sturm and B. Triggs, "A factorization based algorithm for multi-image projective structure and motion," in *European Conference on Computer Vision*, 1996, pp. 709–20.

[14] K. Huang, R. Fossum, and Y. Ma, "Generalized rank conditions in multiple view geometry with applications to dynamical scenes," in *European Conference on Computer Vision*, 2002, pp. 201–215.