Nearest approaches to multiple lines in *n*-dimensional space

Lejia Han and John C. Bancroft

ABSTRACT

An efficient matrix approach is described to obtain the nearest approach to non-intersecting lines in 3-dimensional or any n-dimensional spaces. By the nearest approach, we mean the nearest point(s) or vector(s) to all lines and also each on-line point or vector with respective to that nearest-to-all point(s) or vector(s) as well. The point of nearest approach on each line is also evaluated giving the nearest vector to all lines. The entire solution set can be provided by the matrix approach simultaneously, ensuring efficiency and accuracy at the same time.

INTRODUCTION

Finding the nearest location to pairs of non-intersecting spatial lines representing wave propagation directions in our seismic application (Han *et al*, 2009) had been required. Regardless of specific issues in analytic geometry, a pure data analysis method is developed and tested. Further investigation reveals that this method can be extended to multiple lines in 3-dimensional (3D) space and even in n-dimensional space.

In the following, we will introduce and discuss a matrix technique to find the nearest approach to two lines in 3D space, the nearest approach to multiple lines in 3D space, and the nearest approach to multiple lines in n-dimensional space, including our seismic-related experience and some extensive discuss to our understanding by far.

METHODOLOGY

In each of three approaches below, understanding the matrix-expanding pattern and establishing the variant matrix representation of the linear system respective to the number of given lines and the dimension of space is a key step.

The nearest approach to two lines in 3D space

This is the primary approach applied in our seismic applications (Han, 2010). It leads to additional two approaches as introduced in the next two sections.

Consider two lines in 3D space: l_1 and l_2 defined by the known points $P_1 = (x_{01}, y_{01}, z_{01})$ and $P_2 = (x_{02}, y_{02}, z_{02})$, and the direction cosines $U_1 = (u_{x1}, u_{y1}, u_{z1})$ and $U_2 = (u_{x2}, u_{y2}, u_{z2})$ for l_1 and l_2 respectively. Methods for finding the nearest approach to l_1 and l_2 are plenty in analytic geometry.

The solution by analytic geometry

There are many analytic geometry methods that can be used to find the nearest approach to two spatial lines. For example, the axis rotation method (Pirzadeh and Toussaint, 1998) developed in the Computational Geometry Lab at McGill for the minimum distance between two non-intersecting geometric objects such as polygons in 2D or polyhedral in 3D composed of linear edges and faces. Such sophisticated algorithms of multi-tasks can certainly be adopted to accommodate the two-line case.

In Figure 1, the closest points on the two lines are shown in green and cyan. A red line connects the two points and is the shortest path between the two lines. The black star is the midpoint to the desired nearest point. The axis of Figure 1a is rotated in 1b such that the viewing direction is along one of the lines. In this case, it is the bottom line in Figure 1, and is coincident with the green star in 1b. It is straight forward to construct the line normal to the upper line to the "point", and the midpoint is the desired closest point.

An algebraic method is shown below that used calculus to find the minimum distance between the two lines l_1 and l_2. as defined previously. This method was obtained from Dr. Bancroft (Bancroft, 2010) the pseudo code is:

```
P1  = [x01 y01 z01]
P2  = [x02 y02 z02]
U1  = [ux1 uy1 uz1]
U2  = [ux2 uy2 uz2]
P21 = P2 - P1
M   = cross correlation (U2,U1)
m2  = dot product (M,M)
R   = cross correlation (P21,M/m2)
t1  = dot product(R,U2)
t2  = dot product(R,U1)
Q1  = P1 + t1*U1
Q2  = P2 + t2*U2
%the nearest point location in 3D space
MID_gem = (Q1 + Q2)/2
```

(a)

(b)

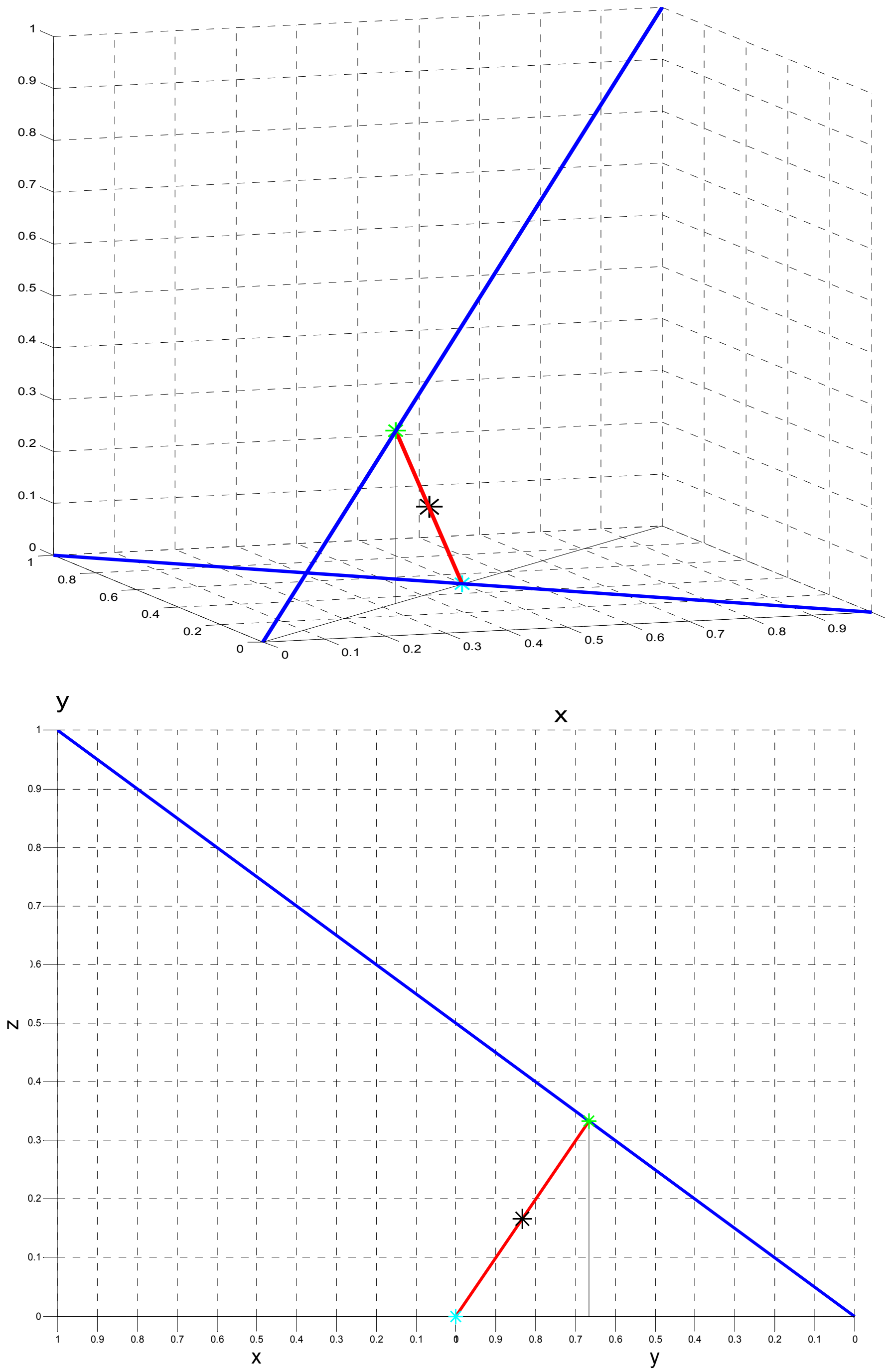

FIG.1 Two lines in blue show the nearest on-line points (blue and green), and the closest points on the line (cyan) and the shortest line (red) are shown in a) as a perspective view and b) a vertical plane viewed along the lower line.

The solution by our matrix approach

Consider l_1 and l_2 as defined previously. Then the two lines can be represented by the following equations:

$$\frac{x-x_{01}}{u_{x1}} = \frac{y-y_{01}}{u_{y1}} = \frac{z-z_{01}}{u_{z1}} = a_1 \tag{A-1}$$

$$\frac{\hat{x}-x_{02}}{u_{x2}} = \frac{\hat{y}-y_{02}}{u_{y2}} = \frac{\hat{z}-z_{02}}{u_{z2}} = a_2 \quad \text{(A-2)}$$

where a_1 and a_2 are parameter variables representing the Euclidean length along the two lines respectively. Based on that, we can construct an expanded form of the two-line linear system in the following way:

$$\begin{cases} x + 0.y + 0.z - u_{x1}.a_1 - 0.a_2 = x_{01} \\ 0.x + y + 0.z - u_{y1}.a_1 - 0.a_2 = y_{01} \\ 0.x + 0.y + z - u_{z1}.a_1 - 0.a_2 = z_{01} \\ \hat{x} + 0.\hat{y} + 0.\hat{z} - 0.a_1 - u_{x2}.a_2 = x_{02} \\ 0.\hat{x} + \hat{y} + 0.\hat{z} - 0.a_1 - u_{y2}.a_2 = y_{02} \\ 0.\hat{x} + 0.\hat{y} + \hat{z} - 0.a_1 - u_{z2}.a_2 = z_{02} \end{cases} \quad \text{(A-3)}$$

A previous method used calculus to find the minimum distance between two lines and we propose to use the same technique by assume in the above equations that (x, y, z) and $(\hat{x}, \hat{y}, \hat{z})$ are the same point and that a least squares solution will find that point. With this assumption we create the ***m*** vector as

$$\boldsymbol{m} = \begin{bmatrix} x \\ y \\ z \\ a_1 \\ a_2 \end{bmatrix}, \quad \text{(A-4)}$$

and anticipate that a1 and a2 will define the nearest point on the two lines. We do not prove this result but have verified the method by comparing the results with the algebraic solution.

We continue the method by defining the ***G*** matrix as

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & -u_{x1} & 0 \\ 0 & 1 & 0 & -u_{y1} & 0 \\ 0 & 0 & 1 & -u_{z1} & 0 \\ 1 & 0 & 0 & 0 & -u_{x2} \\ 0 & 1 & 0 & 0 & -u_{y2} \\ 0 & 0 & 1 & 0 & -u_{z2} \end{bmatrix}, \quad \text{(A-5)}$$

and **d** as

$$\boldsymbol{d} = \begin{bmatrix} x_{01} \\ y_{01} \\ z_{01} \\ x_{02} \\ y_{02} \\ z_{02} \end{bmatrix}. \quad \text{(A-6)}$$

Interesting problems arise when two lines are parallel. We intuitively know that the nearest point becomes a line. The rank of the matrix G is then reduced from 5 to 4. In this case we can use a singular value decomposition (SVD) method.

1. Nearest point by least-squares if G has full column rank

In this situation, we are able to write the least-squares solution to (A-4) as

$$\boldsymbol{m}_{\mathbf{L2}} = (\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{d} \tag{A-8}$$

The nearest spatial location to l_1 and l_2 is determined by the first three elements of (A-8) as

$$\mathrm{MID}_{\mathbf{L2\_}\boldsymbol{all}} = \boldsymbol{m}_{\mathbf{L2}}[1:3], \tag{A-9}$$

and the respective nearest on-line points by the remaining two elements as

$$\mathrm{MID}_{\mathbf{L2\_}\boldsymbol{l1}} = \mathrm{P1} + \mathrm{U1} * \boldsymbol{m}_{\mathbf{L2}}(4), \tag{A-10}$$

$$\mathrm{MID}_{\mathbf{L2\_}\boldsymbol{l2}} = \mathrm{P2} + \mathrm{U2} * \boldsymbol{m}_{\mathbf{L2}}(5). \tag{A-11}$$

2. Nearest point by SVD if G is rank-deficient

The least-squares technique is not applicable to (A-4) in this situation, as (***GTG***) is not invertible, i.e. $(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G})^{-1}$ does not exist. Instead, by using the SVD technique, ***G*** is first decomposed into the following components:

$$\boldsymbol{G} = \boldsymbol{U}_{6x6}\boldsymbol{S}_{6x5}\boldsymbol{V}_{5x5}^{\boldsymbol{T}} \tag{A-12}$$

where ***U*** and ***V*** represent the data space and the model space respectively, and ***S*** represents the diagonal matrix containing the singular values. Assume the rank of ***G*** is *p*, and then the compact form of (A-12) will become

$$\boldsymbol{G} = \boldsymbol{U}_p\boldsymbol{S}_p\boldsymbol{V}_p^{\boldsymbol{T}}. \tag{A-13}$$

It should be noticed that the only rank-deficient case of ***G*** in (A-4) happens at the time when two lines are parallel, which leads to rank (***G***) = 4. We will present and encapsulate details in the extended approach. We are now able to build up the SVD solution to (A-4) as

$$\boldsymbol{m}_{\mathrm{svd}} = \boldsymbol{V}_p\boldsymbol{S}_p^{-1}\boldsymbol{U}_p^{\boldsymbol{T}}\boldsymbol{d} \tag{A-14}$$

Then, the nearest spatial point to l_1 and l_2 can be determined by the first three elements of (A-14) as

$$\mathrm{MID}_{\mathbf{svd\_all}} = \boldsymbol{m}_{\mathrm{svd}}[1:3] \tag{A-15}$$

and the respective nearest points along each line are determined by the remaining two elements as

$$\mathrm{MID}_{\mathbf{svd\_}\boldsymbol{l1}} = \mathrm{P1} + \mathrm{U1} * \mathrm{MID}_{\mathbf{svd}}(4), \tag{A-16}$$

$$\mathrm{MID}_{\mathbf{svd\_}\boldsymbol{l2}} = \mathrm{P2} + \mathrm{U2} * \mathrm{MID}_{\mathbf{svd}}(5). \tag{A-17}$$

To our understanding, the evidence might be reasoned theoretically in the following way, though it might not be a rigorous proof.

1. Unlike seismic data, the known information about the two lines is defined exactly, with no deviation due to, for example, random noise. Therefore, the least-squares solution that minimizes the Euclidean length of ||**d-Gm**|| should be the exact nearest point to the two lines. Therefore, there is no need for a statistical assessment due to noise, as is usually done with seismic data.

2. The SVD technique provides solutions to more general cases than the least-squares technique. It encompasses the case of **G** having full column rank, and in that case the SVD solution is exactly the least-squares solution (Aster et al, 2005).

3. Any pair of non-intersecting lines will lead **G** being full column rank, hence the solution of **Gm=d** is unique for any pair of non-parallel lines in 3D space.

Thus, for any pair of non-intersecting and non-parallel lines in 3D space, the nearest point to both lines is unique, and solutions from analytic geometry, least-squares, and SVD are equivalent. If the two lines are parallel, we know that the nearest point to both is not unique and is any point along some "middle" line nearest to both. Figure 3 shows such a point (the green circle) from the SVD solution. We will encapsulate the complete set of solution points into the general solution addressed in the extended approach later.

Notice that unlike the non-parallel pair, there is no least-squares solution (the blue circle in Figure 2 for the nearest approach to the parallel lines in 4. There is only a single SVD-solution (the green circle in Figure 4) available to meet the request of nearest approach at this parallel situation.

Application to microseismic data

We illustrate the technique with a microseismic problem that has a total of 44 receivers in 3 wells. Each receiver records in 3 axes, from which the direction to the source is estimated. Lines are now defined at the receivers in a direction towards the estimated source location. From the 44 lines we can define 946 pairs of lines to get the estimated nearest points. The lines and their corresponding nearest points are shown in Figure 4. In this data, we used a singular value decomposition (SVD) technique.

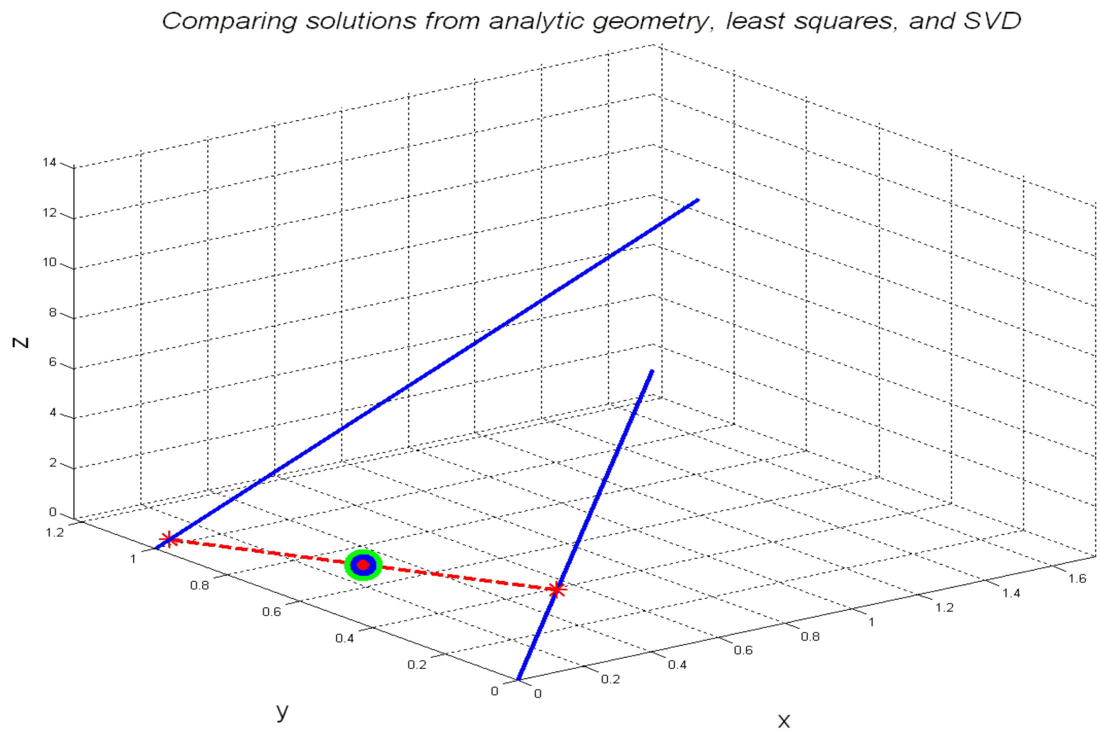

FIG.2 The nearest-point solutions by analytic geometry (red dot), least-squares (blue circle), and SVD (green circle) are identical for a pair of non-parallel lines (blue), as well as the two associated nearest points (red stars) along each line.

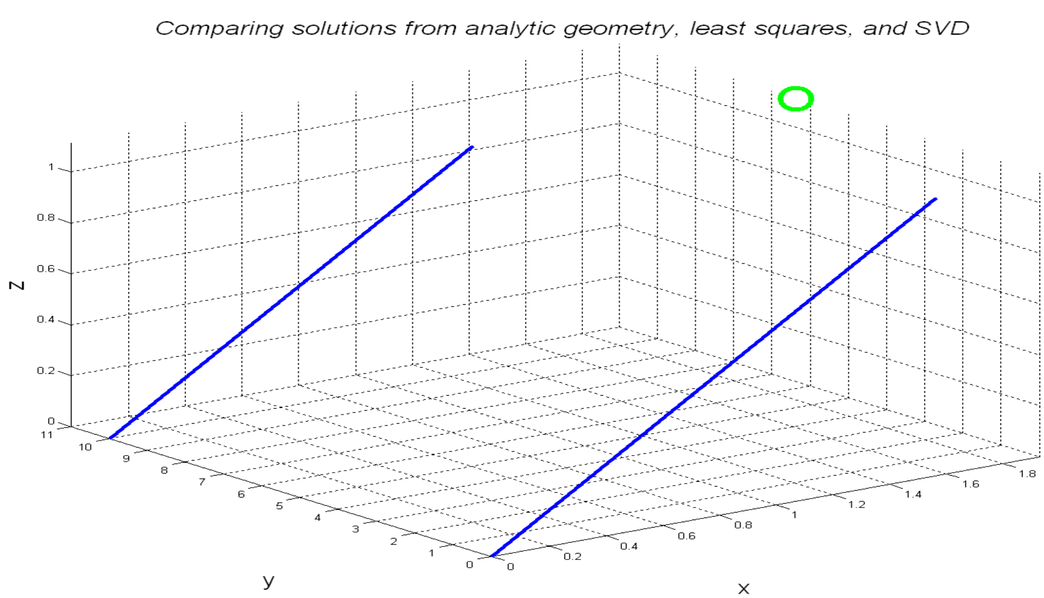

FIG.3 The nearest point to two parallel lines (blue) only results from SVD, not from least-squares and our analytic geometry method. Rank (G) =4 in this situation.

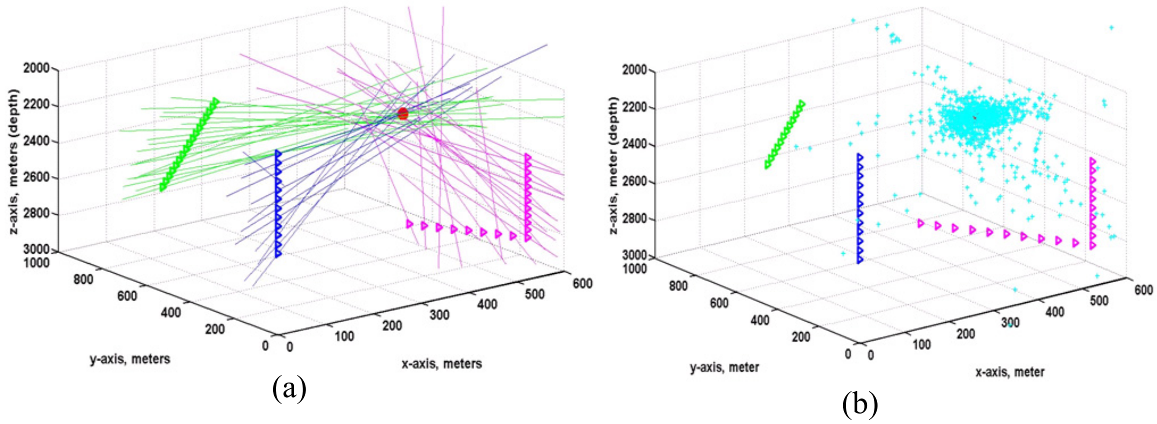

FIG.4 Pairs of lines for all receivers (green, blue, and pink) in (a) produce the nearest points (cyan) in (b).

The nearest approach solution provided by the SVD technique is the one that minimizes the Euclidean lengths of both $||\mathbf{d}\text{-}\mathbf{G}\mathbf{m}_{\mathbf{svd}}||$ and $||\mathbf{m}_{\mathbf{svd}}||$. A theoretical explanation of this issue can be found in the book "Parameter Estimation and Inversion Problems" (Aster et al, 2005). It is also stressed in this book that it is better in practice to use the SVD solution than to use the least-squares solution because of numerical accuracy issues.

The nearest approach to multiple lines in 3D space

Consider *m* lines given in 3D space, with each defined by a known point and direction cosines. We denote the m lines as *l1, l2, l3, …, lm*, the respective on-line points as p1=[x_{01}, y_{01}, z_{01}], p2=[x_{02}, y_{02}, z_{02}], p3=[x_{03}, y_{03}, z_{03}], …., pm=[x_{0m}, y_{0m}, z_{0m}], and the respective direction cosines as U1=[$u_{x1}, u_{y1}, uz1$], U2=[u_{x2}, u_{y2}, u_{z2}], U3=[u_{x3}, u_{y3}, u_{z3}], …, Um=[u_{xm}, u_{ym}, u_{zm}]. Then the *m* lines can be represented by the following equations respectively as

$$\frac{x-x_{01}}{u_{x1}} = \frac{y-y_{01}}{u_{y1}} = \frac{z-z_{01}}{u_{z1}} = a_1,$$

$$\frac{x-x_{02}}{u_{x2}} = \frac{y-y_{02}}{u_{y2}} = \frac{z-z_{02}}{u_{z2}} = a_2,$$

$$\frac{x-x_{03}}{u_{x3}} = \frac{y-y_{03}}{u_{y3}} = \frac{z-z_{03}}{u_{z3}} = a_3,$$

$$\vdots \qquad \text{(B-1)}$$

$$\frac{x-x_{0m}}{u_{xm}} = \frac{y-y_{0m}}{u_{ym}} = \frac{z-z_{0m}}{u_{zm}} = a_m$$

where a_1, a_2, a_3, …, a_m are parameter variables representing the Euclidean lengths along *l1, l2, l3, …, lm* respectively. Thus, the *m*-line linear system can be expanded in the following way:

$$\begin{cases} x + 0.y + 0.z - u_{x1}.a_1 - 0.a_2 - 0.a3 \ldots - 0.a_m = x_{01} \\ 0.x + y + 0.z - u_{y1}.a_1 - 0.a_2 - 0.a3 \ldots - 0.a_m = y_{01} \\ 0.x + 0.y + z - u_{z1}.a_1 - 0.a_2 - 0.a3 \ldots - 0.a_m = z_{01} \\ x + 0.y + 0.z - 0.a_1 - u_{x2}.a_2 - 0.a3 \ldots - o.a_m = x_{02} \\ 0.x + y + 0.z - 0.a_1 - u_{x2}.a_2 - 0.a3 \ldots - o.a_m = y_{02} \\ 0.x + 0.y + z - 0.a_1 - u_{x2}.a_2 - 0.a3 \ldots - o.a_m = z_{02} \\ x + 0.y + 0.z - 0.a_1 - 0.a_2 - u_{x3}.a3 \ldots - 0.a_m = x_{03} \\ 0.x + y + 0.z - 0.a_1 - 0.a_2 - u_{x3}.a3 \ldots - 0.a_m = y_{03} \\ 0.x + 0.y + z - 0.a_1 - 0.a_2 - u_{x3}.a3 \ldots - 0.a_m = z_{03} \\ \vdots \\ x + 0.y + 0.z - 0.a_1 - 0.a_2 - 0.a3 \ldots - u_{xm}.a_m = x_{0m} \\ 0.x + y + 0.z - 0.a_1 - 0.a_2 - 0.a3 \ldots - u_{ym}.a_m = y_{0m} \\ 0.x + 0.y + z - 0.a_1 - 0.a_2 - 0.a3 \ldots - u_{zm}.a_m = z_{0m} \end{cases} \tag{B-2}$$

Accordingly, the matrix representation for the above expanded m-line linear system will be

$$\boldsymbol{Gm} = \boldsymbol{d} \tag{B-3}$$

where G denotes a (m*3) by (m+3) matrix in the following form:

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & -u_{x1} & 0 & 0 & & 0 \\ 0 & 1 & 0 & -u_{y1} & 0 & 0 & & 0 \\ 0 & 0 & 1 & -u_{z1} & 0 & 0 & & 0 \\ 1 & 0 & 0 & 0 & -u_{x2} & 0 & & 0 \\ 0 & 1 & 0 & 0 & -u_{y2} & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & -u_{z2} & 0 & & 0 \\ 1 & 0 & 0 & 0 & 0 & -u_{x3} & & 0 \\ 0 & 1 & 0 & 0 & 0 & -u_{y3} & & 0 \\ 0 & 0 & 1 & 0 & 0 & -u_{z3} & & 0 \\ & & & \vdots & & & \ddots & \vdots \\ 1 & 0 & 0 & -u_{x1}.0 & -u_{x2}.0 & -u_{x3}.0 & & -u_{xm} \\ 0 & 1 & 0 & -u_{y1}.0 & -u_{y2}.0 & -u_{x3}.0 & \cdots & -u_{xm} \\ 0 & 0 & 1 & -u_{z1}.0 & -u_{z2}.0 & -u_{x3}.0 & & -u_{xm} \end{bmatrix}, \tag{B-4}$$

and m and d denote two column vectors of the following forms

$$\boldsymbol{m} = \begin{bmatrix} x \\ y \\ z \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_m \end{bmatrix}, \tag{B-5}$$

$$\boldsymbol{d} = \begin{bmatrix} x_{01} \\ y_{01} \\ z_{01} \\ x_{02} \\ y_{02} \\ z_{02} \\ x_{03} \\ y_{03} \\ z_{03} \\ \vdots \\ x_{0m} \\ y_{0m} \\ z_{0m} \end{bmatrix}. \quad \text{(B-6)}$$

Solving for **m**, we obtain the best fit (x, y, z) for the nearest point along with the points on each line given by the a_i's. The point then defines the best estimated locations shown in Figure 4b.

General solution

For the general solution, we ignore the least-squares technique as it is not able to provide solutions once G is rank-deficient. By using SVD, we derive the general solutions of the nearest approach for all cases of multiple lines in 3D space, including cases with a unique solution and cases with an infinite number of solutions as well, in the following way:

1. Decomposing G in (B-3) by the SVD technique yields

$$\boldsymbol{G} = \boldsymbol{U_{pxp}} * \boldsymbol{S_{pxq}} * \boldsymbol{V_{qxq}^T} \quad \text{(B-7)}$$

where p=m*3 and q=m+3 are used to indicate the p by q matrix of G, U is a p by p orthogonal matrix with columns that are unit vectors spanning the data space (Rm), V is a q by q orthogonal matrix with columns that are basis vectors spanning the model space (Rn), S is an m by n diagonal matrix with nonnegative diagonal elements (singular values).

2. Expanding the above SVD representation of G in terms of the columns of U and V gives

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{U}_{.,1}, & \boldsymbol{U}_{.,2}, & \cdots & \boldsymbol{U}_{.,k}, & \dots & \boldsymbol{U}_{.,p} \end{bmatrix} \begin{bmatrix} \boldsymbol{S_p} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_{.,1}, & \boldsymbol{V}_{.,2}, & \cdots & \boldsymbol{V}_{.,k}, & \dots & \boldsymbol{V}_{.,q} \end{bmatrix}^{\boldsymbol{T}} \quad \text{(B-8)}$$

3. Simplify G into the compact forms

$$\boldsymbol{G} = [\boldsymbol{U_k} \quad \boldsymbol{U_0}] \begin{bmatrix} \boldsymbol{S_p} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} [\boldsymbol{V_k} \quad \boldsymbol{V_0}]^{\boldsymbol{T}}$$

$$\boldsymbol{G} = \boldsymbol{U_k} * \boldsymbol{S_k} * \boldsymbol{V_k^T} \quad \text{(B-9)}$$

4. Obtain the SVD solution to (B-3)

$$\boldsymbol{m_{svd}} = \boldsymbol{V_k} * \boldsymbol{S_k^{-1}} * \boldsymbol{U_k^T} * \boldsymbol{d} \quad \text{(B-10)}$$

5. Obtain the SVD solution point nearest to all m lines given in 3D space

$$\mathrm{MID}_{\boldsymbol{toAll}} = \boldsymbol{m_{svd}}[1{:}3] \qquad \text{(B-11)}$$

6. Obtain all respective on-line nearest points

$$\begin{aligned} \mathrm{MID}_{\boldsymbol{tol1}} &= \mathrm{P1} + \mathrm{U1} * m_{svd}[4], \\ \mathrm{MID}_{\boldsymbol{tol2}} &= \mathrm{P2} + \mathrm{U2} * m_{svd}[5], \\ \mathrm{MID}_{\boldsymbol{tol3}} &= \mathrm{P3} + \mathrm{U3} * m_{svd}[6], \\ &\vdots \\ \mathrm{MID}_{\boldsymbol{tolm}} &= \mathrm{Pm} + \mathrm{Um} * m_{svd}[3+m]. \end{aligned} \qquad \text{(B-12)}$$

7. If k=q, then the solution of nearest approach to all m lines is unique and is the collection of the finite number of spatial points just derived above:

$$\{\mathrm{MID}_{\boldsymbol{toAll}}, \mathrm{MID}_{\boldsymbol{tol1}}, \mathrm{MID}_{\boldsymbol{tol2}}, \mathrm{MID}_{\boldsymbol{tol3}}, \ldots, \mathrm{MID}_{\boldsymbol{tolm}}\}. \qquad \text{(B-13)}$$

8. If k<q, then the solution set of nearest approach to all m lines is not unique and is the collection of the finite number of spatial points as determined below:

$$\boldsymbol{m}_{toAllPara} = \boldsymbol{m_{svd}} + \boldsymbol{V}_0 * \boldsymbol{H} \qquad \text{(B-14)}$$

where $\boldsymbol{V_0}$ is a p by (*q-k*) matrix spanning the null space of $\boldsymbol{G^T}$, and $\boldsymbol{H}$ is a column vector of *p-k* elements scaling $\boldsymbol{V_0}$. With $\boldsymbol{G}$ from the expanded matrix model of the *m*-line linear system in 3D space, k is (q-1) definitely if k<q, and accordingly $\boldsymbol{V_0}$ becomes a one column vector. Then the above equation can be reduced to the following compact form:

$$\boldsymbol{m}_{toAllPara} = \boldsymbol{m_{svd}} + h * \boldsymbol{v}_0 \qquad \text{(B-15)}$$

where h is a single scaling parameter and $\boldsymbol{v}_0$is a p-element column vector.

We are now able to define the infinite number of spatial points qualifying as the nearest point to all *m* lines as

$$\mathrm{MID}_{toAllPara} = \boldsymbol{m}_{toAllPara}[1{:}3], \qquad \text{(B-16)}$$

and the respective on-line nearest points as

$$\begin{aligned} \mathrm{MID}_{tol1} &= \mathrm{P1} + \mathrm{U1} * \boldsymbol{m}_{toAllPara}[4], \\ \mathrm{MID}_{tol2} &= \mathrm{P2} + \mathrm{U2} * \boldsymbol{m}_{toAllPara}[5], \\ \mathrm{MID}_{tol3} &= \mathrm{P3} + \mathrm{U3} * \boldsymbol{m}_{toAllPara}[6], \\ &\vdots \\ \mathrm{MID}_{tolm} &= \mathrm{P}m + \mathrm{U}m * \boldsymbol{m}_{toAllPara}[3+m]. \end{aligned} \qquad \text{(B-17)}$$

Thus we are able to collect the complete solution sets at this situation as

$$\{\mathrm{MID}_{toAllPara}, \mathrm{MID}_{tol1}, \mathrm{MID}_{tol2}, \mathrm{MID}_{tol2}, \mathrm{MID}_{tol3}, \ldots, \mathrm{MID}_{tolm}\}. \qquad \text{(B-18)}$$

We will illustrate the four cases by choosing four lines. Figure 5 shows the four different cases.

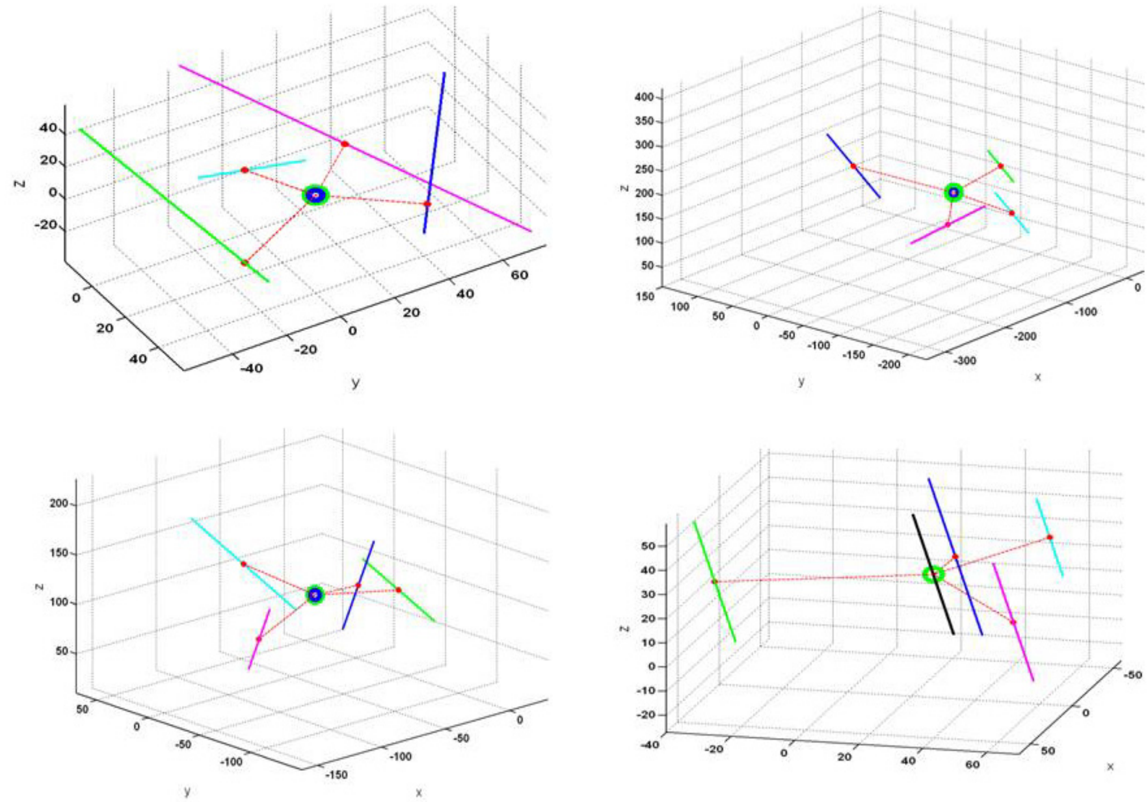

FIG.5 Case illustration with four non-intersecting lines (a) no parallel at all, (b) two mutual-parallel pairs (cyan and green, pink and blue), (c) 3 lines parallel (blue, cyan, and green), and (d) all parallel. Part (a), (b), and (c) show three cases with a single unique solution, while part (d) shows the case with an infinite number of equivalently nearest points along the solution line (black).

It can be observed in Figure 6 that if four lines are all parallel, then the solution sets of the nearest approach to all four lines is infinite, as indicated by the black line in part (d), instead of the single unique point as shown in part (a), (b), and (c).

To be specific, if there is any non-parallel pair within any number of m lines given in 3D space, then the solution of the nearest approach to the multiple lines, including the nearest point to all lines as $\mathrm{MID}_{toAllPara}$ the respective nearest points along each line as MID_{l1}, MID_{l2}, MID_{l2}, MID_{l3}, …, MID_{lm}, is not unique, as defined in the above equations accordingly.

Notice that at such an all-parallel case, the matrix approach cannot be applied with the least-squares technique; although it is solvable by SVD, we are limited to a single set of solution. In contrast to the single SVD-solution point, Figure 6 illustrates one of the infinite non-SVD solution points along the black line, i.e., all points equivalently nearest to the four parallel lines (color-coded in cyan, green, pink, and blue). However the SVD solution is shown by the black point and is closest to the origin. Choosing the origin closest to the active solution area may enable the use of more points when using the SVD method.

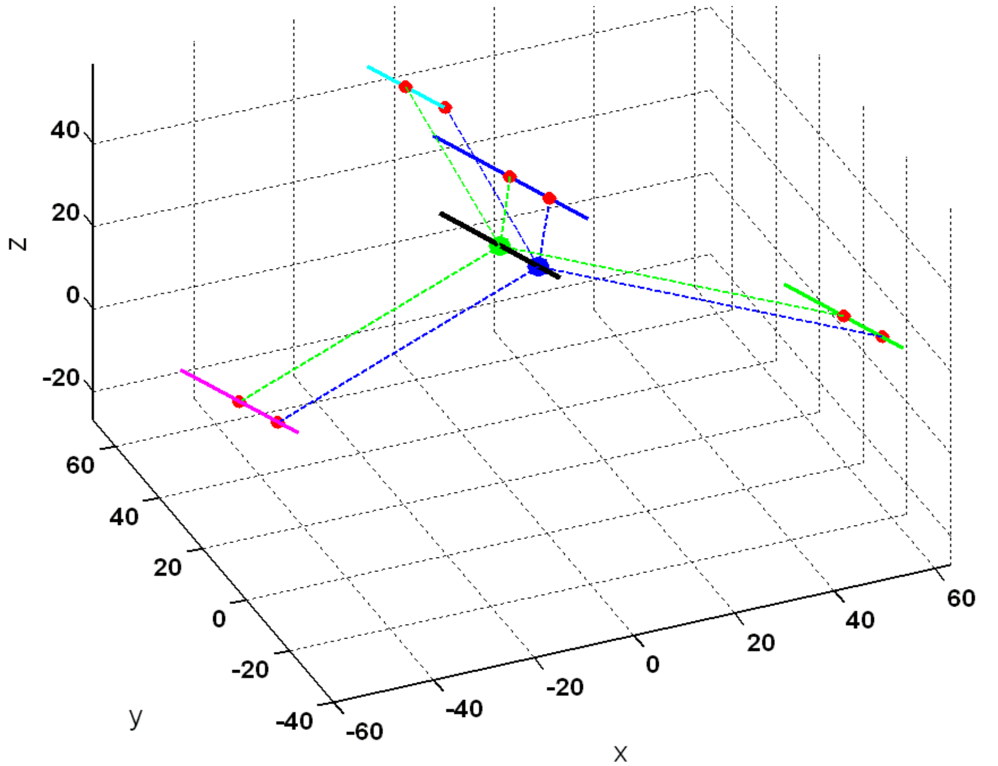

FIG.6 Illustration of the single unique SVD solution point (green dot) and one of the other infinite solution points (blue dot), as well as their respective on-line nearest points, i.e., four red dots connected with green dashed lines and four red dots connected with blue dashed lines.

The nearest approach to multiple lines in n-dimensional space

Consider m lines given in an n-dimensional space. Assume we know a orthogonal basis $\boldsymbol{w}=[w_1, w_2, w_3, ..., w_n]$ about this space and a vector on each line, denoted as p1=$[x_{11}, x_{12}, x_{13}, ..., x_{1n}]$, p2=$[x_{21}, x_{22}, x_{23}, ..., x_{2n}]$, p3=$[x_{31}, x_{32}, x_{33}, ..., x_{3n}]$, …., pm=$[x_{m1}, x_{m2}, x_{m3}, ..., x_{mn}]$ on l_1, l_2, l_3, …., l_m respectively. Also assume that we know the direction cosines of each line to the set of orthogonal basis, denoted as u1=$[u_{11}, u_{12}, u_{13}, ..., u_{1n}]$, u2=$[u_{21}, u_{22}, u_{23}, ..., u_{2n}]$, u3=$[u_{31}, u_{32}, u_{33}, ..., u_{3n}]$, …um=$[u_{m1}, u_{m2}, u_{m3}, ..., u_{mn}]$ for l_1, l_2, l_3, …., l_m respectively.

Notice that it seems common to have the orthogonal projections of the m lines on w. We then expect that the sets of direction cosines could be derived by taking the dot products of the projected line portions with w. However, we ignore the mathematical details and assume we are able to obtain the direction cosines for each line in n-dimensional space, as well. Based on this, we can follow the previous linear system expanding pattern and obtain our solutions for the nearest approach to multiples lines in n-dimensional space.

The *m* lines defined above can be represented respectively by the following linear equations:

$$\frac{x_1 - x_{11}}{u_{11}} = \frac{x_2 - x_{12}}{u_{12}} = \frac{x_3 - x_{13}}{u_{13}} = \cdots = \frac{x_n - x_{1n}}{u_{1n}} = a_1,$$

$$\frac{x_1 - x_{21}}{u_{21}} = \frac{x_2 - y_{22}}{u_{22}} = \frac{x_3 - z_{23}}{u_{23}} = \cdots = \frac{x_n - x_{2n}}{u_{2n}} = a_2,$$

$$\frac{x_1 - x_{31}}{u_{31}} = \frac{x_2 - x_{32}}{u_{32}} = \frac{x_3 - x_{33}}{u_{33}} = \cdots = \frac{x_n - x_{3n}}{u_{3n}} = a_3,$$

$$\vdots \tag{C-1}$$

$$\frac{x_1 - x_{m1}}{u_{m1}} = \frac{x_2 - x_{m2}}{u_{m2}} = \frac{x_3 - x_{m3}}{u_{m3}} = \cdots = \frac{x_n - x_{mn}}{u_{mn}} = a_m$$

$$\begin{cases} x_1 + 0.x_2 + 0.x_3 + \cdots + 0.x_n - u_{11}.a_1 - 0.a_2 - 0.a3 \ldots - 0.a_m = x_{11} \\ 0.x_1 + x_2 + 0.x_3 + \cdots + 0.x_n - u_{12}.a_1 - 0.a_2 - 0.a3 \ldots - 0.a_m = x_{12} \\ 0.x_1 + 0.x_2 + x_3 + \cdots + 0.x_n - u_{13}.a_1 - 0.a_2 - 0.a3 \ldots - 0.a_m = x_{13} \\ \vdots \\ 0.x_1 + 0.x_2 + 0.x_3 + \cdots + 1.x_n - u_{1n}.a_1 - 0.a_2 - 0.a3 \ldots - a_m = x_{1n} \\ x_1 + 0.x_2 + 0.x_3 + \cdots + 0.x_n - 0.a_1 - u_{21}.a_2 - 0.a3 \ldots - 0.a_m = x_{21} \\ 0.x_1 + x_2 + 0.x_3 + \cdots + 0.x_n - 0.a_1 - u_{21}.a_2 - 0.a3 \ldots - 0.a_m = x_{22} \\ 0.x_1 + 0.x_2 + x_3 + \cdots + 0.x_n - 0.a_1 - u_{23}.a_2 - 0.a3 \ldots - 0.a_m = x_{23} \\ \vdots \\ 0.x_1 + 0.x_2 + 0.x_3 + \cdots + 1.x_n - 0.a_1 - u_{2n}.a_2 - 0.a3 \ldots - a_m = x_{2n} \\ x_1 + 0.x_2 + 0.x_3 + \cdots + 0.x_n - 0.a_1 - 0.a_2 - u_{31}.a3 \ldots - 0.a_m = x_{31} \\ 0.x_1 + x_2 + 0.x_3 + \cdots + 0.x_n - 0.a_1 - 0.a_2 - u_{32}.a3 \ldots - 0.a_m = x_{32} \\ 0.x_1 + 0.x_2 + x_3 + \cdots + 0.x_n - 0.a_1 - 0.a_2 - u_{33}.a3 \ldots - 0.a_m = x_{33} \\ \vdots \\ 0.x_1 + 0.x_2 + 0.x_3 + \cdots + 1.x_n - 0.a_1 - u_{3n}.a_2 - 0.a3 \ldots - a_m = x_{3n} \\ \vdots \\ x_1 + 0.x_2 + 0.x_3 + \cdots + 0.x_n - 0.a_1 - 0.a_2 - 0.a3 \ldots - u_{m1}.a_m = x_{m1} \\ 0.x_1 + x_2 + 0.x_3 + \cdots + 0.x_n - 0.a_1 - 0.a_2 - 0.a3 \ldots - u_{m2}.a_m = x_{m2} \\ 0.x_1 + 0.x_2 + x_3 + \cdots + 0.x_n - 0.a_1 - 0.a_2 - 0.a3 \ldots - u_{m3}.a_m = x_{m3} \\ \vdots \\ 0.x_1 + 0.x_2 + 0.x_3 + \cdots + x_n - 0.a_1 - 0.a_2 - 0.a3 \ldots - u_{mn}.a_m = x_{mn} \end{cases} \tag{C-2}$$

Then, a point of nearest approach to all m lines could be obtained generally in the following matrix representation of their linear system as

$$\boldsymbol{Gm} = \boldsymbol{d} \tag{C-3}$$

where G is a (m*n) by (m+n) matrix with the following form

$$
\boldsymbol{G} = \begin{bmatrix}
1 & 0 & 0 & 0 & \dots & 0 & -u_{11} & 0 & & 0 \\
0 & 1 & 0 & 0 & \dots & 0 & -u_{12} & 0 & & 0 \\
0 & 0 & 1 & 0 & \dots & 0 & -u_{13} & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & 0 & \dots & 0 & 1 & -u_{1m} & 0 & & 0 \\
 & & & & \vdots & & & & \ddots & \vdots \\
1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & & -u_{m1} \\
0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & & -u_{m2} \\
0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & -u_{m3} \\
 & & & & \vdots & & & & & & & \vdots \\
0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & & & -u_{mn}
\end{bmatrix}, \qquad \text{(C-4)}
$$

and m and d denote two column vectors with the following forms respectively

$$
\boldsymbol{m} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \\ a_1 \\ a_2 \\ a_3 \\ \dots \\ a_m \end{bmatrix} \qquad \text{(C-5)}
$$

$$
\boldsymbol{d} = \begin{bmatrix} x_{011} \\ \dots \\ x_{01n} \\ x_{021} \\ \dots \\ x_{02n} \\ \dots \\ x_{0n1} \\ \dots \\ x_{0nn} \end{bmatrix}. \qquad \text{(C-6)}
$$

Following the approach to the general solution in 3D space, we derive the general solution of nearest approach to multiple lines in n-dimensional space as follows:

1. Decompose G in (C-3) by the SVD technique

$$
\boldsymbol{G} = \boldsymbol{U_{pxp}} * \boldsymbol{S_{pxq}} * \boldsymbol{V_{qxq}^T} \qquad \text{(C-7)}
$$

where $p=m*n$ and $q=m+n$ are used to indicate the p by q matrix of $\boldsymbol{G}$, $\boldsymbol{U}$ is a p by p orthogonal matrix with columns that are unit vectors spanning the data space ($\boldsymbol{R^m}$), $\boldsymbol{V}$ is a q by q orthogonal matrix with columns that are basis vectors spanning the model space ($\boldsymbol{R^n}$), $\boldsymbol{S}$ is an m by n diagonal matrix with nonnegative diagonal elements (singular values).

2. Expanding the SVD representation of $\boldsymbol{G}$ in terms of the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ gives

$$
\boldsymbol{G} = [\boldsymbol{U}_{.,1}, \quad \boldsymbol{U}_{.,2}, \quad \cdots \quad \boldsymbol{U}_{.,k}, \quad \dots \quad \boldsymbol{U}_{.,p}] \begin{bmatrix} \boldsymbol{S_p} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} [\boldsymbol{V}_{.,1}, \quad \boldsymbol{V}_{.,2}, \quad \cdots \quad \boldsymbol{V}_{.,k}, \quad \dots \quad \boldsymbol{V}_{.,q}]^{\boldsymbol{T}} \qquad \text{(C-8)}
$$

3. Simplify G into the compact forms

$$G = [U_k \quad U_0] \begin{bmatrix} S_p & 0 \\ 0 & 0 \end{bmatrix} [V_k \quad V_0]^T$$

$$G = U_k * S_k * V_k^T \tag{C-9}$$

4. Obtain the SVD solution to (C-3)

$$m_{svd} = V_k * S_k^{-1} * U_k^T * d \tag{C-10}$$

5. Obtain the SVD solution point nearest to all lines

$$\text{MID}_{toAll} = m_{svd}[1:n] \tag{C-11}$$

6. Obtain all respective on-line nearest points

$$\begin{aligned} \text{MID}_{tol1} &= \text{P1} + \text{U1} * m_{svd}[n+1], \\ \text{MID}_{tol2} &= \text{P2} + \text{U2} * m_{svd}[n+2], \\ \text{MID}_{tol3} &= \text{P3} + \text{U3} * m_{svd}[n+3], \\ &\vdots \\ \text{MID}_{tolm} &= \text{Pm} + \text{Um} * m_{svd}[n+m]. \end{aligned} \tag{C-12}$$

7. If k=q, then the solution of the nearest approach to all m lines is unique and is the collection of the finite number of spatial points just derived above:

$$\{\text{MID}_{toAll}, \text{MID}_{tol1}, \text{MID}_{tol2}, \text{MID}_{tol3}, \ldots, \text{MID}_{tolm}\}. \tag{C-13}$$

8. If k<q, then the solution set of the nearest approach to all m lines is not unique and is the collection of the finite number of spatial points as determined below:

$$m_{toAllPara} = m_{svd} + V_0 * H \tag{C-14}$$

where V0 is a p by (q-k) matrix spanning the null space of GT, and H is a column vector of p-k elements scaling V0. With G from the expanded matrix model of the m-line linear system in n-dimensional space, k is (q-1) definitely if k<q, and accordingly V0 becomes a one column vector. Thus the above equation can be reduced to the following compact form:

$$m_{toAllPara} = m_{svd} + h * v_0 \tag{C-15}$$

where h is a single scaling parameter and v_0 is a p-element column vector.
Then we are able to define the infinite number of spatial points qualifying as the nearest point to all m lines as

$$\text{MID}_{toAllPara} = m_{toAllPara}[1:n], \tag{C-16}$$

and the respective on-line nearest points as

$$\begin{aligned} \text{MID}_{tol1} &= \text{P1} + \text{U1} * m_{toAllPara}[n+1], \\ \text{MID}_{tol2} &= \text{P2} + \text{U2} * m_{toAllPara}[n+2], \\ \text{MID}_{tol3} &= \text{P3} + \text{U3} * m_{toAllPara}[n+3], \\ &\vdots \\ \text{MID}_{tolm} &= \text{P}m + \text{U}m * m_{toAllPara}[n+m]. \end{aligned} \tag{C-17}$$

Thus we are able to collect the complete solution sets of this situation as

$$\{\mathrm{MID}_{toAllPara}, \mathrm{MID}_{tol1}, \mathrm{MID}_{tol2}, \mathrm{MID}_{tol3}, \mathrm{MID}_{tol4}, \ldots, \mathrm{MID}_{tolm}\}. \quad \text{(C-18)}$$

To be specific, if there is any non-parallel pair within any number of m lines given in n-dimensional space, the solution of nearest approach to the m lines, including the nearest point to all lines as $\mathrm{MID}_{toAllPara}$ and the respective nearest points along each line as MID_{tol1} MID_{tol2}, MID_{tol3}, MID_{tol4}, …, MID_{tolm}, is not unique, as defined in the above equations.

CONCLUSION

The matrix approaches above might be an efficient alternative to some analytic geometry methods for the nearest vector(s) to multiple lines at various geometry relations (intersecting, non-intersecting, or parallel) in 3D space or their extensions in n-dimensional space. The respective set(s) of nearest on-line vectors can be obtained simultaneously and efficiently as well.

ACKNOWLEDGEMENT

I was encouraged to further investigate this purely mathematical issue encountered in my thesis research by Dr. John Bancroft, with his primary testing codes, and comparing results from analytic geometry, least-squares, and SVD.

The authors gratefully acknowledge the support of the CREWES sponsors.

REFERENCES

Aster, R.C., Borchers, B. and Thurber, C.H. (2005). Parameter Estimation and Inverse problems.

Pirzadeh, H. and Toussaint, G.T. (1998). Computational geometry with the rotating clipers. McGill University.

Bard, M. and Himel, D. (2001). The minimum distance between two lines in n-space.

Sunday, D. (2010). Distances between lines and segments with their closest point of approach. http://softsurfer.com/Archive/algorithm_0106/algorithm_0106.htm#Closest%20Point%20of%20Approach.