EECS C106B / 206B Robotic Manipulation and Interaction **Spring 2021**

Lecture 16: Multiple-View Reconstruction from Scene Knowledge

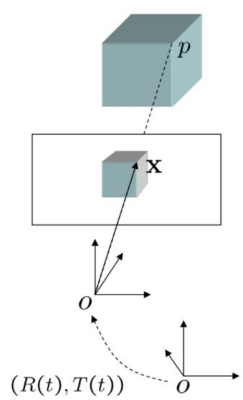
Scribes: Nareauphol Liu, Thiti Khomin

16.1 Review: Mutiple View Geometry

16.1.1 Multiple View Matrix for Point Features

Recall that:

1. We define the homogeneous coordinates of a 3-D point p to be
$$\mathbf{X} = [X, Y, Z, W]^T \in \mathbb{R}^4 \quad (\mathrm{W} = 1) \tag{16.1}$$
2. We define the homogeneous coordinates of its 2-D image to be
$$\mathbf{x} = [x, y, z]^T \in \mathbb{R}^3 \quad (\mathrm{z} = 1) \tag{16.2}$$



Looking at the picture above and Equations (16.1) and (16.2), we are able to define the Projection of a 3-D point to an image plane to be
$$\lambda(t)\boldsymbol{x}(t) = \Pi(t)\boldsymbol{X} \tag{16.3}$$

where $\lambda(t) \in \mathbb{R}, \Pi(t) = [R(t), T(t)] \in \mathbb{R}^{3\times 4}$ and $R(t) \to A(t)R(t), T(t) \to A(t)T(t)$

Recall that

16.1.2 Multiple View Matrix for Line Features

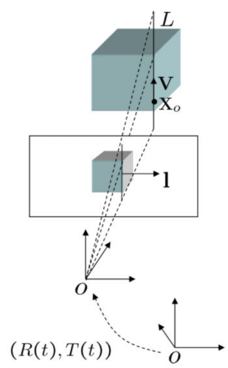
The projection of the line onto its 2-D image lies in the same plane spanned by the 3-D edge.

1. We define the homogeneous coordinates of a 3-D line L to be

$$\mathbf{X} = \mathbf{X_0} + \mu\mathbf{V}, \quad \text{where} \quad \mathbf{X_O}, \mathbf{V} \in \mathbb{R}^4, \mu \in \mathbb{R} \tag{16.4}$$

2. We define the homogeneous coordinates of its 2-D image to be

$$\mathbf{l} = [a, b, c]^T \in \mathbb{R}^3, \qquad \mathrm{ax + by + c = 0} \tag{16.5}$$



Looking at the picture above and Equations (16.4) and (16.5), we can compute the projection of a 3-D line to an image plane:

$$\mathbf{l}(t)^T \boldsymbol{x}(t) = \mathbf{l}(t)^T \Pi(t) \boldsymbol{X} = 0, \text{where} \Pi(t) = [R(t), T(t)] \in \mathbb{R}^{3\mathrm{x}4}, \tag{16.6}$$

$$\Pi(t) = [R(t), T(t)] \in \mathbb{R}^{3\mathrm{x}4}, \tag{16.7}$$

16.1.3 Ranks of Multi-view Matrix

Rank of a feature allows us to understand the geometric reconstruction of the object. The multi-view matrix of point, M_p, and line features, M_l, is given by:

$$\mathrm{M}_p = \begin{bmatrix} \widehat{\boldsymbol{x}_2} R_2 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2} T_2 \\ \widehat{\boldsymbol{x}_3} R_3 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_3} T_3 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m} R_m \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m} T_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 2}, \quad M_l = \begin{bmatrix} \boldsymbol{l}_2^T R_2 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3 \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_3^T T_3 \\ \vdots & \vdots \\ \boldsymbol{l}_m^T R_m \widehat{\boldsymbol{l}_1} & \boldsymbol{l}_m^T T_m \end{bmatrix} \in \mathbb{R}^{(m-1)\times 4}$$

If all line and point features 3D points intercept the same plane as the image's 2-D plane, the view matrix, M, has a rank of 1. However, if $rank(M) \leq 1$, there would be a loss of 3-D information of the image. The figure shown below visualizes this structure:

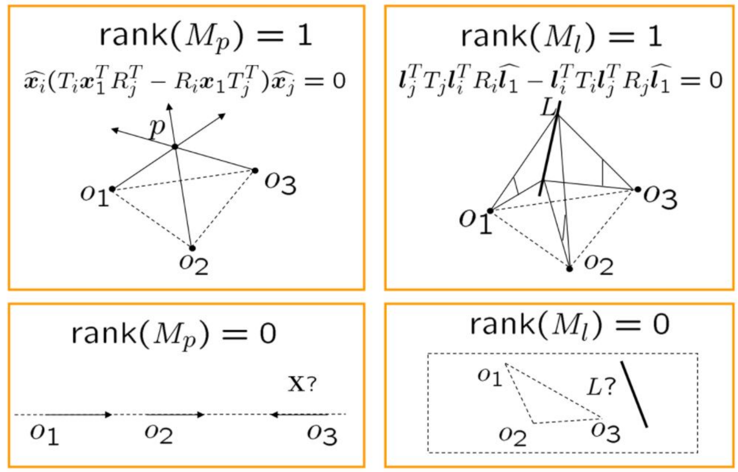


Figure 16.1: Rank and Image Reconstruction

In the case that rank(M) greater than 1, we would be generating different projections of the line as shown in the figure below

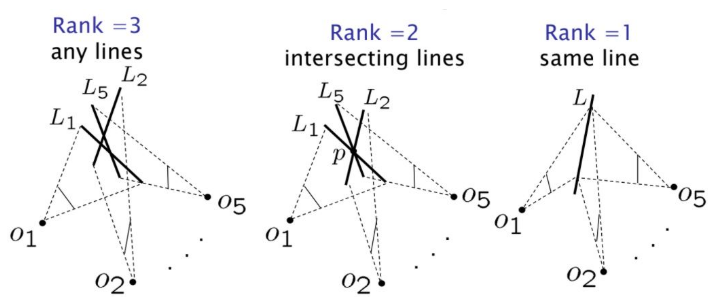


Figure 16.2: Possible Resulting images for different Ranks of a line reconstruction

As such, in order to reconstruct our feature, we want to find the rotation and translation of our views such that the rank is 1.

16.1.4 Universal Rank Constraint

In many cases, we have treated points and lines as separate entities. However, in reality there can be a joint relationship between a point and a line such as, say, a point on the corner of the window connects to an edge of a window. As such we can generate a universal rank condition:

$$M \doteq \begin{bmatrix} D_2^{\perp} R_2 D_1 & D_2^{\perp} T_2 \\ D_3^{\perp} R_3 D_1 & D_3^{\perp} T_3 \\ \vdots & \vdots \\ D_m^{\perp} R_m D_1 & D_m^{\perp} T_m \end{bmatrix}, \quad \text{where} \quad \begin{cases} D_i \doteq & \boldsymbol{x}_i \ \text{ or } \ \widehat{\boldsymbol{l}}_i, \\ D_i^{\perp} \doteq & \widehat{\boldsymbol{x}_i} \ \text{ or } \ \boldsymbol{l}_i^T. \end{cases}$$

we have that $D_1 = \hat{l_1}$ and $D_i^{\perp} = \hat{x}_i$ for some $i \geq 2$ then:

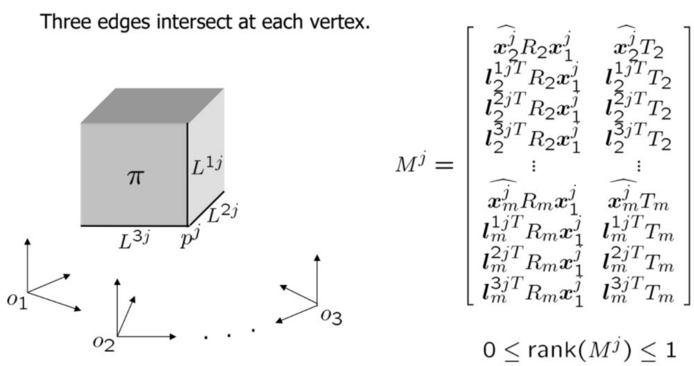
$$1 \leq rank(M) \leq 2 \tag{16.8}$$

Otherwise:

$$0 \leq rank(M) \leq 1 \tag{16.9}$$

16.1.5 Universal Rank Constraint (Example): Multiple Images of a cube

By utilizing this constraint, we can construct the multi view matrix of a cube as well as compute its rank. Below is the computation of the view matrix, M^j, of a cube where edges intersect at the vertex, p^j



What this is really saying is how those planes and those rays intersect in 3D. If it is rank 2, then each image of the line gives you a plane.

16.1.6 Coplanar Point Features

Extending the universal rank constraint, we can talk about the multiple view matrix for co-planar point features. The homogeneous representation of a 3D plane π is:

$$aX + bY + cZ + d = 0 \tag{16.10}$$

where $\pi X = 0$, $\pi = [\pi^1, \pi^2] : \pi^1 \in \mathbb{R}^3, \pi^2 \in \mathbb{R}^2$. The equation describes the surface normal of the plane π. Conceptually it is not different from the plane that we got from the image of the line.

Thus we can write our new M matrix to be: M $\doteq \begin{bmatrix} D_2^{\perp} R_2 D_1 & D_2^{\perp} T_2 \\ D_3^{\perp} R_3 D_1 & D_3^{\perp} T_3 \\ \vdots & \vdots \\ D_m^{\perp} R_m D_1 & D_m^{\perp} T_m \\ \pi^1 D_1 & \pi^2 \end{bmatrix}$

Recall that the M_l matrix for the line features is $\in \mathbb{R}^{(m-1)\text{x}4}$ and notice that we need 4 points in the M matrix for coplanar point features, which shows that conceptually they are both similar.

All you have to do to specify all the image from the points and lines which lie on the same plan π, is to add the row $[\pi^1 D_1 .pi^2]$ to the multiple view matrix. Everything else remains the same such as the universal rank condition. So we can rewrite out M_p and M_l matrix as:

$$M_p = \begin{bmatrix} \widehat{\boldsymbol{x}_2} R_2 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_2} T_2 \\ \widehat{\boldsymbol{x}_3} R_3 \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_3} T_3 \\ \vdots & \vdots \\ \widehat{\boldsymbol{x}_m} R_m \boldsymbol{x}_1 & \widehat{\boldsymbol{x}_m} T_m \\ \pi^1 \boldsymbol{x}_1 & \pi^2_\circ \end{bmatrix} \in \mathbb{R}^{(3m-2)\times 2} \qquad 0 \leq \text{rank}\,(M_p) \leq 1$$

$$\mathrm{M}_l = \begin{bmatrix} \boldsymbol{l}_2^T R_2 \widehat{\boldsymbol{l}}_1 & \boldsymbol{l}_2^T T_2 \\ \boldsymbol{l}_3^T R_3 \widehat{l}_1 & \boldsymbol{l}_3^T T_3 \\ \vdots & \vdots \\ \boldsymbol{l}_m^T R_m \widehat{\boldsymbol{l}}_1 & \boldsymbol{l}_m^T T_m \pi^1 \widehat{\boldsymbol{l}_1} \\ \pi^2 & \end{bmatrix} \in \mathbb{R}^{m\times 4} \qquad 0 \leq \text{rank}\,(M_l) \leq 1$$

Additionally, the new matrix gives us homography as the new matrix enforces homography:

1. Point feature: $\widehat{\boldsymbol{x}_i}\left(R_i\pi^2 - T_i\pi^1\right)\boldsymbol{x}_1 = 0$
2. Line feature: $\boldsymbol{l}_i^T\left(R_i\pi^2 - T_i\pi^1\right)\widehat{\boldsymbol{l}_1} = 0$

16.1.7 General Rank Constraint for Dynamic Scenes

In some higher dimensional spaces, if we see some low dimensional projections and take many of these views, how should they be related?

Take the image above as an example - If we have a dynamic point moving in 3D space with constant speed or accelerating, we can describe our trajectory as:
$X(t) = X_0 + tv_0 + \frac{t^2}{2}a$

$$\overline{\mathbf{X}} = \begin{bmatrix} X_0 \\ v_0 \\ a \\ 1 \end{bmatrix} \in \mathbb{R}^{10}, \bar{\Pi}(t) = \begin{bmatrix} I & tI & \frac{t^2}{2}I & 0 \end{bmatrix} \in \mathbb{R}^{3\times 10}, \lambda(t)\boldsymbol{x}(t) = \bar{\Pi}(t)\overline{\mathbf{X}}, \quad \mathbf{X} \in \mathbb{R}^{10}, \boldsymbol{x} \in \mathbb{R}^3$$

where X_0 is the position of the point, v_0 is the velocity of the point, and a is the acceleration of the point.

You can view the motion or the image of the moving point is a projection from some higher order dimension which have these parameters and we simply take an image of it

Thus we can generalize the constraints to a higher dimensional space to be:

$$\lambda(t)\boldsymbol{x}(t) = \Pi(t)\mathbf{X}(t)\mathbf{X}(t) = [b_1(t), b_2(t), \ldots, b_{n+1}(t)]\,\overline{\mathbf{X}}, \quad \lambda(t)\boldsymbol{x}(t) = \bar{\Pi}(t)\overline{\mathbf{X}}, \quad \overline{\mathbf{X}} \in \mathbb{R}^{n+1}, \boldsymbol{x} \in \mathbb{R}^3.$$

You must have the unified way of describing the image. Thus the projection from the n-dimensional space to a k-dimensional space could be describe as:

$$\begin{gathered} \lambda_i \boldsymbol{x}_i = \bar{\Pi}_i \mathbf{X}_i, \quad i = 1, \ldots, m \\ \bar{\Pi}_i = \begin{bmatrix} \bar{R}_i & \bar{T}_i \end{bmatrix} \\ \bar{R}_i \in \mathbb{R}^{(k+1)\times(k+1)}, \bar{T}_i \in \mathbb{R}^{(k+1)\times(n-k)} \end{gathered}$$

If the images through multiple views are truly the same image in the higher dimensional space, it will satisfy the general rank condition of multiple view matrix M describe to be:

$$M \doteq \begin{bmatrix} \left(D_2^{\perp}\right)^T \bar{R}_2 D_1 & \left(D_2^{\perp}\right)^T \bar{T}_2 \\ \left(D_3^{\perp}\right)^T \bar{R}_3 D_1 & \left(D_3^{1}\right)^T \bar{T}_3 \\ \vdots & \vdots \\ \left(D_m^{\perp}\right)^T \bar{R}_m D_1 & \left(D_m^{\perp}\right)^T \bar{T}_m \end{bmatrix}$$

where D_i's and $D_i^{\perp}$'s and are images and coimages of hyperplane respectively

16.1.8 Summary

1. Incidence relations $\Longleftrightarrow$ rank conditions

2. Rank conditions $\rightarrow$ multiple view factorization (think projections of the same image from higher dimension spaces)

3. Rank conditions imply all multi-focal constraints

4. Rank conditions for point, line planes, symmetric structures also follow the same philosophy (motivation for the next part)

16.2 Multiple-View Reconstruction from Scene Knowledge

Motivation: Pictures are rich and we are not exploiting most of the image. We want want to exploit certain structures and symmetry of the entire picture to give us more information we can use.

In order to reconstruct an image, we can utilize knowledge of an object such as parallelism, orthogonality, congruence, self-similarity, patterns, etc. Symmetry can give away a lot of information about an object and it can assist us.

By using symmetry, it is possible that robots can utilize this information in order to figure out its own position (self-orient themselves) which is very important in current field of robotics. As such, it can navigate by looking at camera images.

Example: Equivalent views from rotational symmetry

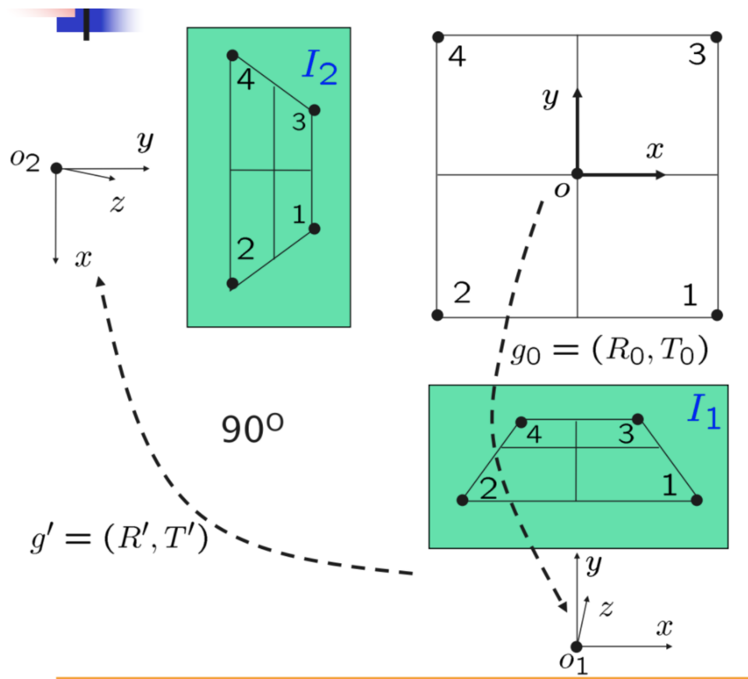


Figure 16.3: Views from rotational symmetry

From the picture above we can notice:

1. Same corner points are viewed differently depending on the symmetry
2. Interpret the same image taken from another view via g transformation and using the structural invariant seen in rotational symmetry
3. We assume the object appearance doesn't change after a transformation to another position. The object doesn't become itself after a rotational transformation
4. For some objects, we can get thousands of images of symmetry

This same philosophy and structure can be implied to other forms of symmetry in reflection symmetry, translational symmetry, etc. where the main idea is still the same as the rotational symmetry.

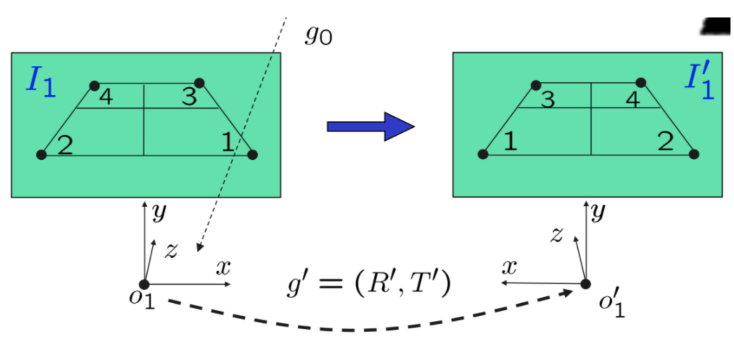


Figure 16.4: Views from reflection symmetry

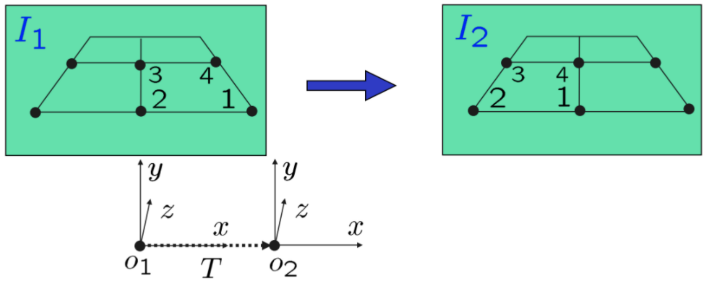


Figure 16.5: Views from translational symmetry

Definition: A set of 3-D features, S, is called a symmetric structure if there exists a nontrivial subgroup G of $E(3)$ that acts on it such that for every g in G, the map

$$g \in G : S \to S \tag{16.11}$$

is an (isometric) automorphism of S. We say the structure S has a group symmetry G.

$$\begin{aligned} \boldsymbol{X} &= [X, Y, Z, 1]^T \in \mathbb{R}^4, \quad \boldsymbol{x} = [x, y, z]^T \in \mathbb{R}^3 \\ g_0 &= \begin{bmatrix} R_0 & T_0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}, \quad \Pi_0 = [I, 0] \in \mathbb{R}^{3\times 4} \end{aligned}$$

We put our camera at some relative position g_0 and we take a picture Π_0. We can rotate the board or have some sort of symmetry transformation to get another picture. Every element allowed transformation in the symmetry group, we can take a different picture. This can be described mathematically as:

$$\boldsymbol{x} \sim \Pi_0 g_0 \boldsymbol{X} \to g(\boldsymbol{x}) \sim \Pi_0 g_0 g \boldsymbol{X}$$

We can interpret $g(\boldsymbol{x}) \sim \Pi_0 g_0 g \boldsymbol{X}$ as $g(\boldsymbol{x}) \sim \Pi_0 g_0 g \boldsymbol{X} = \Pi_0 g_0 g g_0^{-1} g_0 \boldsymbol{X}$. We can generalize this to:

$$\begin{aligned} g_1(\boldsymbol{x}) &\sim \Pi_0 g_0 g_1 g_0^{-1} (g_0 \boldsymbol{X}) \\ g_2(\boldsymbol{x}) &\sim \Pi_0 g_0 g_2 g_0^{-1} (g_0 \boldsymbol{X}) \\ &\vdots \\ g_m(\boldsymbol{x}) &\sim \Pi_0 g_0 g_m g_0^{-1} (g_0 \boldsymbol{X}) \end{aligned}$$

We realize that it is the same original image under going the different rigid body motions or the same image take from multiple perspectives. Thus symmetry has the interpretation of taking the same image from different positions which are different rigid body motions, which makes the recovery of the original image a lot easier.