EECS C106B / 206B Robotic Manipulation and Interaction **Spring 2022**

Lecture 16: Geometry and Algebra of Multiple views

Scribes: Doug Pan, Sunisha Fernandez

16.1 Overview

In this section, we will be discussing about the geometry of point and line features, their corresponding multiple view matrix, geometric interpretation of the ranks and reconstruction algorithm. We will also be discussing the Universal rank constraints and list out various examples for the same.

16.2 Geometry of point features

16.2.1 Point Features

Let us consider a point p in 3-D with its homogeneous coordinates as

$$X = [X, Y, Z, W]^T \in \mathbb{R}^4, \quad where \quad W = 1$$

The corresponding homogeneous coordinates of its 2-D image will be

$$x = [x, y, z]^T \in \mathbb{R}^3, \quad where \quad z = 1$$

The projection of the 3-D point X to an image plane will be given as

$$\lambda(t)x(t) = \Pi(t)X$$

where $\lambda(t) \in \mathbb{R}$ is called the depth scale component and $\Pi(t) = [R(t), T(t)] \in \mathbb{R}^{3\times 4}$ is called the multiple-view projection matrix.

16.2.2 Multiple view matrix for a point and its rank

Without loss of generality (WLOG), let us choose frame 1 as the reference such that

$$\lambda_i x_i = \lambda_1 R_i x_1 + T_i$$

Taking the cross product of x_i and the above equation implies

$$x_i \times (\lambda_i x_i = \lambda_1 R_i x_1 + T_i) \iff \begin{bmatrix} \widehat{x_i} R_i x_1 & \widehat{x_i} T_i \end{bmatrix} \begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix} = 0$$

We could generalize this for all frames or reference and thus create M_p

$$M_p = \begin{bmatrix} \widehat{x_2} R_2 x_1 & \widehat{x_2} T_2 \\ \widehat{x_3} R_3 x_1 & \widehat{x_3} T_3 \\ \vdots & \vdots \\ \widehat{x_m} R_m x_1 & \widehat{x_m} T_m \end{bmatrix} \in \mathbb{R}^{3(m-1)\times 2}$$

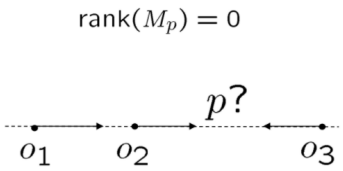


Figure 16.1: The three images and the three optical centres lie on a straight line

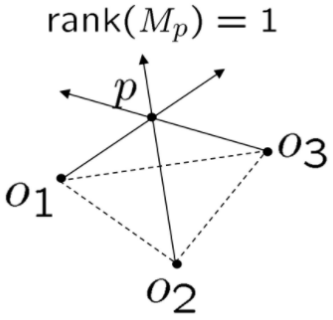


Figure 16.2: Three rays extended from the three images x_1, x_2, x_3 from their camera origin intersect at one point p in space

where M_p is called the multiple view matrix associated with a point feature p. Here, M_p involves both the image x_1 and the co-images $\widehat{x_2}, \widehat{x_3}$.....$\widehat{x_m}$.The M_p encodes the 3-D information.

The above matrix has a rank $M_p \leq 1$ where rank(M_p)=1 is generic (as all rows when multiplied with $\begin{bmatrix}\lambda & 1\end{bmatrix}^T$ will give 0 and thus it will be linearly dependent) and rank(M_p)=0 is degenerate.

Figure 16.1 is a case where Rank (Mp) = 0, there is no clear solution. One possibility is that the center axis of all three images are collinear and it is not possible to determine depth. Figure 16.2 is a case where Rank (Mp) = 1, it means that it is possible to determine the depth from the three images.

16.2.3 Geometric interpretation of the rank condition

Given three vectors $x_1, x_2, x_3 \in \mathbb{R}^3$ as shown in Figure 16.2, they should automatically satisfy both bilinear (epipolar) and trilinear constraints

$$x_i^T \widehat{T}_i R_i x_1 = 0 \tag{16.1}$$

$$\widehat{x_i}(R_i x_1 T_j^T - T_i x_1^T R_j^T)\widehat{x_j} = 0 \tag{16.2}$$

where equation 16.1 is Bilinear/epipolar constraint and equation 16.2 is trilinear constraint.

The situation demostrated in Figure 16.1 shows regardless of what 3-D feature point one chooses, epipolar constraints alone do not provide sufficient constraints to determine a unique preimage point from any given three images. Fortunately,the 3-D preimage is uniquely determined if the three images satisfy the trilinear constraint, except for one rare degenerate configuration in which the point p in question lies on the line between collinear optical centers.

Other ways to formulate the constraints of m corresponding images of n points x_i^j

$$\lambda_i^j x_i^j = \Pi_i X^j$$

$$\Pi_i = [R_i, T_i]$$

The above given equations are equivalent to

$$H_p = \begin{bmatrix} \Pi_1 & x_1 & 0 & \dots & 0 \\ \Pi_2 & 0 & x_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \Pi_1 & x_1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} X \\ -\lambda_1 \\ \vdots \\ -\lambda_m \end{bmatrix} = 0 \tag{16.3}$$

$$H_p \in \mathbb{R}^{3m \times (m+4)}, \quad \det\left(H_{(m+4)\times(m+4)}\right) = 0$$

and

$$F_p = \begin{bmatrix} \widehat{x_1}\Pi_1 \\ \widehat{x_2}\Pi_2 \\ \vdots \\ \widehat{x_m}\Pi_m \end{bmatrix} \tag{16.4}$$

$$X = 0, F_p \in \mathbb{R}^{3m \times 4}, \quad \det(F_{4\times 4}) = 0$$

Equation 16.3 was presented by Heyden et al. and Equation 16.4 was presented Faugeras et al.

16.2.4 Reconstruction Algorithm for point features

Given m images of n points

$$(x_1^j, \dots, x_i^j, \dots, x_m^j)_{j=1,\dots,n}^{i=1,\dots,m}$$

The equation relating points provided we have rotation and translations of cameras is given by

$$\lambda^j \begin{bmatrix} \widehat{x_2^j} R_2 x_1^j \\ \widehat{x_3^j} R_3 x_1^j \\ \vdots \\ \widehat{x_m^j} R_m x_1^j \end{bmatrix} + \begin{bmatrix} \widehat{x}_2^j T_2 \\ \widehat{x}_3^j T_3 \\ \vdots \\ \widehat{x}_m^j T_m \end{bmatrix} = 0, \in \mathbb{R}^{3(m-1)\times 1}$$

The equation relating rotation and translations provided we have the points is given by

$$P_i \begin{bmatrix} R_i^s \\ T_i^s \end{bmatrix} = \begin{bmatrix} \lambda^1 x_1^1 \otimes \widehat{x_i^1} & \widehat{x_i^1} \\ \lambda^2 x_1^2 \otimes \widehat{x_i^2} & \widehat{x_i^2} \\ \vdots & \vdots \\ \lambda^n x_1^n \otimes \widehat{x_i^n} & \widehat{x_i^n} \end{bmatrix} \begin{bmatrix} R_i^s \\ T_i^s \end{bmatrix} = 0, \in \mathbb{R}^{3n\times 1}$$

where $\otimes$ is the Kronecker product.

In general P_i will have rank 11, if $n \geq 6$

Given m images of $n > 6$ points, for the j-th point,

$$\begin{bmatrix} \widehat{x_2^j} R_2 x_1^j & \widehat{x_2^j} T_2 \\ \widehat{x_3^j} R_3 x_1^j & \widehat{x_3^j} T_3 \\ \widehat{x_m^j} R_m x_1^j & \widehat{x_m^j} T_m \end{bmatrix} \begin{bmatrix} \lambda^j \\ 1 \end{bmatrix} = 0$$

We can extract λ^{j^s} using singular value decomposition (SVD).

Similarly, for the i-th image,

$$\begin{bmatrix} \lambda^1 x_1^1 \otimes \widehat{x_i^1} & \widehat{x_i^1} \\ \lambda^2 x_1^2 \otimes \widehat{x_i^2} & \widehat{x_i^2} \\ \vdots & \vdots \\ \lambda^n x_1^n \otimes \widehat{x_i^n} & \widehat{x_i^n} \end{bmatrix} \begin{bmatrix} R_i^s \\ T_i^s \end{bmatrix} = 0$$

We can extract (R_i^s, T_i^s) using singular value decomposition (SVD).

Thus, the algorithm followed is:

1. Initialization
   - Set k = 0
   - Compute (R_2, T_2) using the 8-point algorithm
   - Compute $\lambda^j = \lambda_k^j$ and normalize so that $\alpha_k^1 = 1$
2. Compute $(\tilde{R}_i, \tilde{T}_i)$ as the null space of $P_{i=2,\ldots,m}$
3. Compute new $\lambda^j = \lambda_{k+1}^j$ as the null space of $M^j, j = 1, \ldots, n$. Normalize so that $\lambda_{k+1}^1 = 1$
4. If $\|\lambda_k - \lambda_{k+1}\| \leq \epsilon$ stop, else $k = k + 1$ and goto step 2.

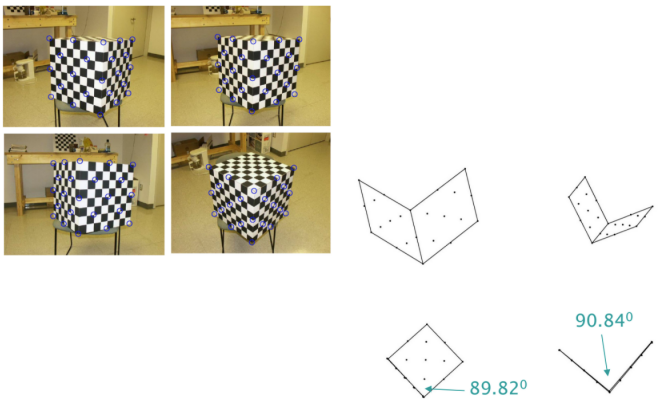


Figure 16.3: Four images of the cube and its features and the reconstructed 3-D features viewed from novel vantage points

16.3 Geometry of line features

16.3.1 Line Features

Let us consider the homogeneous representation of a line L in 3-D as

$$X = X_o + \mu V, \quad X_o, V \in \mathbb{R}^4, \mu \in \mathbb{R}$$

The corresponding homogeneous representation of its 2-D image will be

$$l = [a, b, c]^T \in \mathbb{R}^3, \quad ax + by + c = 0$$

The projection of the 3-D line to an image plane will be given as

$$l(t)^T x(t) = l(t)^T \Pi(t) X = 0$$

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

where $\lambda(t) \in \mathbb{R}$ is called the depth scale component and $\Pi(t) = [R(t), T(t)] \in \mathbb{R}^{3\times 4}$ is called the multiple-view projection matrix.

16.3.2 Multiple view matrix for a line and its rank

The Multiple view matrix associated with a line feature L (M_l) is given by:

$$M_l = \begin{bmatrix} l_2^T R_2 \widehat{l_1} & l_2^T T_2 \\ l_3^T R_3 \widehat{l_1} & l_3^T T_3 \\ \vdots & \vdots \\ l_m^T R_m \widehat{l_1} & l_m^T T_m \end{bmatrix} \in \mathbb{R}^{(m-1)\times 4}$$

16.3.3 Geometric interpretation of the rank condition

Given three vectors $l_1, l_2, l_3 \in \mathbb{R}^3$ that are images of the same line L in space, they satisfy the trilinear constraint (Equation 16.5).

$$l_j^T T_j l_i^T R_i \widehat{l_1} - l_i^T T_i l_j^T R_j \widehat{l_1} = 0 \tag{16.5}$$

Figure 16.4 shows that the rank of M_l is 1 and Figure 16.5 shows that if the rank of M_l is 0.

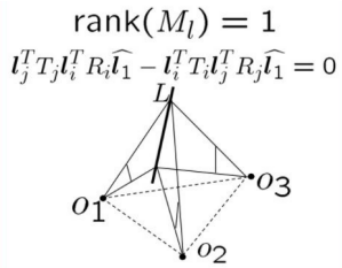


Figure 16.4: Three planes extended from the three images from their camera origin(o_1,o_2,o_3) intersect in one line L in space

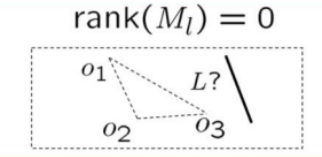


Figure 16.5: The three images and the three optical centres (o_1,o_2,o_3) lie on a plane P

Let us consider a condition where the multi-view matrix is

$$M_l = \begin{bmatrix} l_2^T R_2 \widehat{l_1} & l_2^T T_2 \\ l_3^T R_3 \widehat{l_1} & l_3^T T_3 \\ l_4^T R_4 \widehat{l_1} & l_4^T T_4 \\ l_5^T R_5 \widehat{l_1} & l_5^T T_5 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

Thus the rank of M_l

$$rank(M_l) = 3, 2, 1$$

The figure corresponding for each rank is displayed below in Figure 16.6

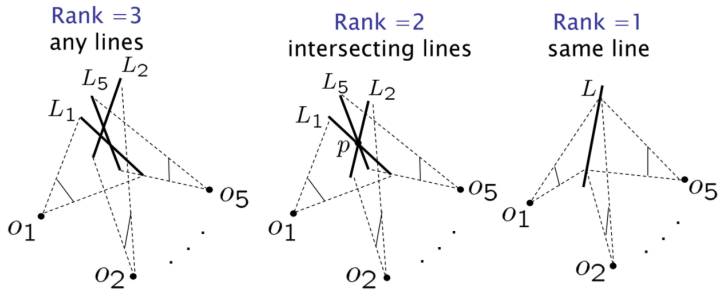


Figure 16.6: Line features for higher ranks

16.3.4 Reconstruction algorithm for line features

1. Initialization

   - Set k = 0
   - Compute (R_2, T_2) and (R_3, T_3) using linear algorithm
   - Compute $\lambda^j = \lambda_k^j$ and normalize so that $\alpha_k^1 = 1$

2. Compute $(\tilde{R}_i, \tilde{T}_i)$ as the null space of $P_{i=2,\ldots,m}$

3. Compute new $\lambda^j = \lambda_{k+1}^j$ as the null space of $M^j, j = 1, \ldots, n$. Normalize so that $\lambda_{k+1}^1 = 1$

4. If $\|\lambda_k - \lambda_{k+1}\| \leq \epsilon$ stop, else $k = k + 1$ and goto step 2.

16.3.5 Summary of point and line features

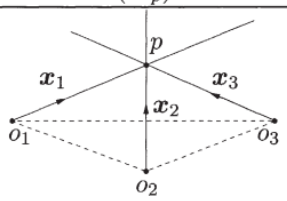
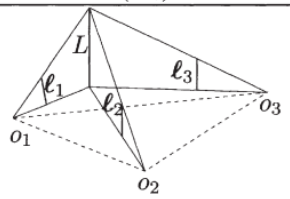
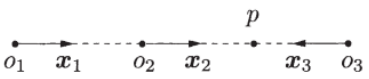
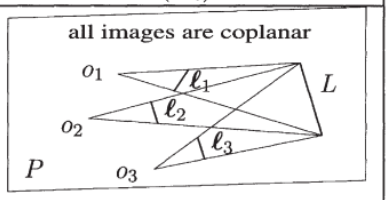
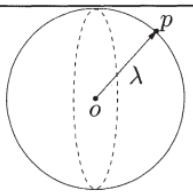
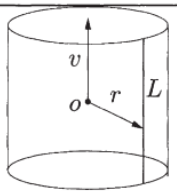
Point feature	Line feature
$M_p = \begin{bmatrix} \widehat{x_2}R_2x_1 & \widehat{x_2}T_2 \\ \widehat{x_3}R_3x_1 & \widehat{x_3}T_3 \\ \vdots & \vdots \\ \widehat{x_m}R_mx_1 & \widehat{x_m}T_m \end{bmatrix}$	$M_l = \begin{bmatrix} \ell_2^T R_2 \widehat{\ell_1} & \ell_2^T T_2 \\ \ell_3^T R_3 \widehat{\ell_1} & \ell_3^T T_3 \\ \vdots & \vdots \\ \ell_m^T R_m \widehat{\ell_1} & \ell_m^T T_m \end{bmatrix}$
$\text{rank}(M_p) = 1$	$\text{rank}(M_l) = 1$
	
$\text{rank}(M_p) = 0$	$\text{rank}(M_l) = 0$
all images are colinear 	all images are coplanar 
Constraints from $\text{rank}(M_p) \leq 1$	Constraints from $\text{rank}(M_l) \leq 1$
$x_i^T \widehat{T}_i R_i x_1 = 0$	$\ell_i^T R_i \widehat{\ell_1} R_j^T \ell_j = 0$
$\widehat{x_i}(T_i x_1^T R_j^T - R_i x_1 T_j^T)\widehat{x_j} = 0$	$\ell_j^T (T_j \ell_i^T R_i - R_j \ell_i^T T_i)\widehat{\ell_1} = 0$
3-D information encoded in M_p	3-D information encoded in M_l
	

Figure 16.7: Comparison between multiple images of a point and a line

16.4 Universal Rank Constraints

Points and lines are treated as separate entities in many cases. However, in reality, there could be a relation between a point and a line. Therefore, joint incidence relations are not exploited. In such cases, we can form a universal rank condition:

$$M = \begin{bmatrix} D_2^{\perp} R_2 D_1 & D_2^T T_2 \\ D_3^{\perp} R_3 D_1 & D_3^T T_3 \\ \vdots & \vdots \\ D_m^{\perp} R_m D_1 & D_m^T T_m \end{bmatrix}$$

where,

$$D_i = x_i \quad or \quad \widehat{l_i}$$
$$D_i^{\perp} = \widehat{x_i} \quad or \quad \widehat{l_i}^T$$

If $D_1 = \widehat{l_1}$ and $D_i^{\perp} = \widehat{x_i}$ for some $i \geq 2$, then for Multi-nonlinear constraints

$$1 \leq rank(M) \leq 2$$

otherwise for Multi-linear constraints,

$$0 \leq rank(M) \leq 1$$

16.4.1 Universal Rank Constraint: Points and Lines

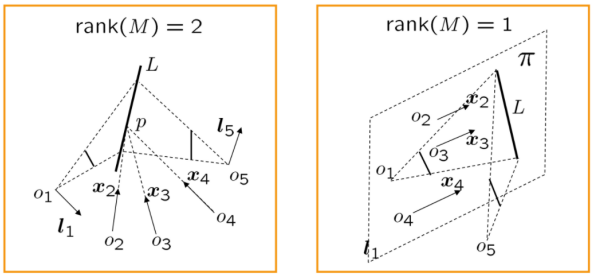


Figure 16.8: The image on the left is a generic configuration for the case rank $(M_l p) = 2$. The image on the right is a degenerate configuration for the case rank(M) 1: a line-point-point-point-line scenario.

$$M = \begin{bmatrix} \widehat{x_2}R_2\widehat{l_1} & \widehat{x_2}T_2 \\ \widehat{x_3}R_3\widehat{l_1} & \widehat{x_3}T_3 \\ \widehat{x_4}R_4\widehat{l_1} & \widehat{x_4}T_4 \\ \widehat{l_5}^T R_5\widehat{l_1} & \widehat{l_5}^T T_5 \end{bmatrix} \in \mathbb{R}^{10\times 4}; 1 \leq rank(M) \leq 2$$

According to Figure 16.8, If rank(M) = 2, it can be shown that the point p must lie on the preimage plane of l_1, and it is also the preimage of all the image points $x_2, x_3, ..., x_m$. The line L, however, is determined only up to this plane, and the point p may or may not have to be on this line.

16.4.2 Universal Rank Constraint: Family of intersecting lines

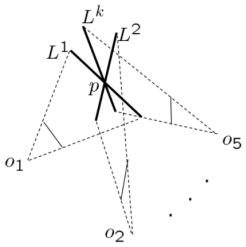


Figure 16.9: Rank 2 always takes the case where you take in the reference frame an image of the line. This means that if you use the universal rank condition, $D_1 = \widehat{l_1}$

$$M = \begin{bmatrix} \widehat{l_2}^T R_2 \widehat{l_1} & \widehat{l_2}^T T_2 \\ \widehat{l_3}^T R_3 \widehat{l_1} & \widehat{l_3}^T T_3 \\ \widehat{l_4}^T R_4 \widehat{l_1} & \widehat{l_4}^T T_4 \\ \widehat{l_5}^T R_5 \widehat{l_1} & \widehat{l_5}^T T_5 \end{bmatrix} \in \mathbb{R}^{4\times 4}; 1 \leq rank(\tilde{M}_l) \leq 2$$

16.4.3 Universal Rank Constraint: Multiple images of a cube

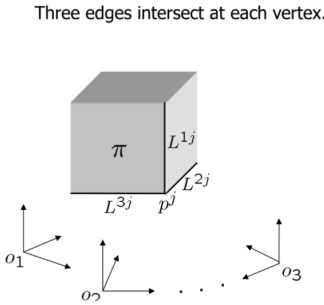


Figure 16.10: Suppose you start with looking at the vertex of the cube, then you can express the matrix with any images of the lines L^{1-3}. You have many components. Rank of the matrix is an algebraic convenient way to write all of it. Solve for λ, R, T

$$M^j = \begin{bmatrix} \widehat{x_2}_j R_2 x_1^j & \widehat{x_2^j} T_2 \\ l_2^{1jT} R_2 \widehat{x_1^j} & l_2^{1jT} T_2 \\ l_2^{2jT} R_2 \widehat{x_1^j} & l_2^{2jT} T_2 \\ l_2^{3jT} R_2 \widehat{x_1^j} & l_2^{2jT} T_2 \\ \vdots & \vdots \\ \widehat{x_m}_j R_m x_1^j & \widehat{x_m^j} T_m \\ l_m^{1jT} R_m \widehat{x_1^j} & l_m^{1jT} T_m \\ l_m^{2jT} R_m \widehat{x_1^j} & l_m^{2jT} T_m \\ l_m^{3jT} R_m \widehat{x_1^j} & l_m^{2jT} T_m \end{bmatrix} 0 \leq rank(M^j) \leq 1$$

16.4.4 Multiple View Matrix for Coplanar Point features

Let us consider a homogeneous representation of a 3D plane π :

$$aX + bY + cZ + d = 0$$

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. Thus, the multiple view matrix M will be:

$$M = \begin{bmatrix} D_2^{\perp} R_2 D_1 & D_2^{\perp} T_2 \\ D_3^{\perp} R_3 D_1 & D_3^{\perp} T_3 \\ \vdots & \vdots \\ D_m^{\perp} R_m D_1 & D_m^{\perp} T_m \\ \pi^1 D_1 & \pi^2 \end{bmatrix}$$

We need to specify all the images from the points and lines which lie on the same plane π and add the row $[\pi^1 D_1, \pi^2]$ to the multiple view matrix. Everything else remains the same as the universal rank condition.

So, the M_p and M_l matrix are as follows:

$$M_p = \begin{bmatrix} \widehat{x_2} R_2 x_1 & \widehat{x_2} T_2 \\ \widehat{x_3} R_3 x_1 & \widehat{x_3} T_3 \\ \vdots & \vdots \\ \widehat{x_m} R_m x_1 & \widehat{x_m} T_m \\ \pi^1 x_1 & \pi^2 \end{bmatrix} \in \mathbb{R}^{3(m-2)\times 2}; 0 \leq rank(M_p) \leq 1$$

$$M_l = \begin{bmatrix} l_2^T R_2 \widehat{l_1} & l_2^T T_2 \\ l_3^T R_3 \widehat{l_1} & l_3^T T_3 \\ \vdots & \vdots \\ l_m^T R_m \widehat{l_1} & l_m^T T_m \\ \pi^1 \widehat{l_1} & \pi^2 \end{bmatrix} \in \mathbb{R}^{(m)\times 4}; 0 \leq rank(M_l) \leq 1$$

Simultaneously, this results a homography in addition to the previous constraints

$$\widehat{x_i}(R_i \pi^2 - T_i \pi^1) x_1 = 0$$

$$l_i^T (R_i \pi^2 - T_i \pi^1) \widehat{l_1} = 0$$

16.4.5 Multiple View Matrix for Coplanar Point features

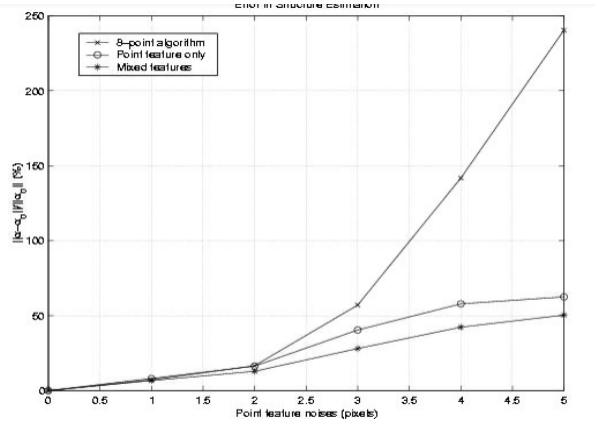
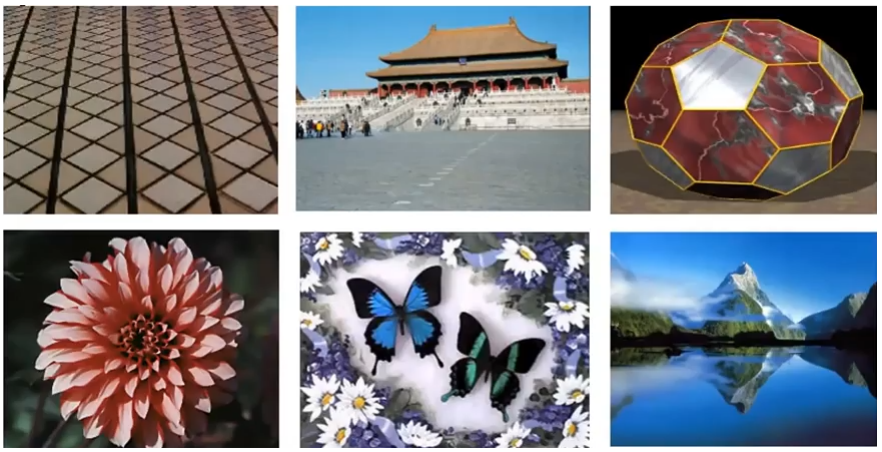


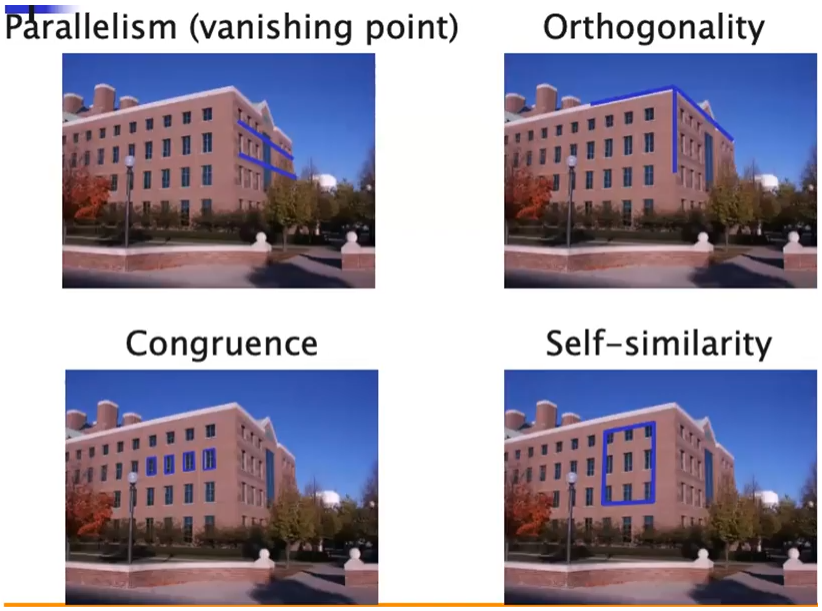
Figure 16.11: You can verify performance of this approach by comparing the 8 point algorithm, point features-only and using mixed features. Horizontal is the level of noise at measurement. Vertical axis is accuracy where it recovery rotation and translation. With 8point, it can only tolerate certain kinds of noise. You use points, it gets better. You use more and more information, you get more and more better recovery. 8 point algorithm start to fall apart. You use 8point to initialize and use multi point to get the solution better.

16.5 Scene Knowledge and Symmetry

The world is much more beautiful and complex than points, lines, planes. When looking at images of the world, you don't even need a second image, or even a second image to know where exactly you are. Why do objects like these give away their pose or geometry?



You can guess based on some geometric properties listed below. They are all different types of symmetry - rotational, translational, or reflective symmetry.



Our brains are tuned to symmetry, and is very powerful. There has been a deep history of work in this area spanning mathematics, cognitive science, computer vision, detection and recognition and reconstruction. To organize our thoughts, there are three main questions.

1. Why does an image of a symmetric object give away its structure?
2. Why does an image of a symmetric object give away its pose?
3. What else can we get from an image of a symmetric object?

Here is the key. Mathematics is about reduction. If there's ONE thing you take away, it is this. You get equivalent views from rotational geometry.

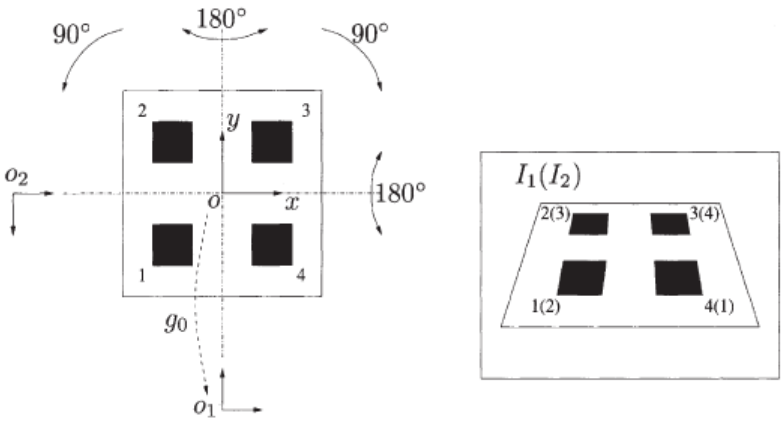


Figure 16.12: Left: a checkerboard exhibits different types of symmetries, including rotations about o by 90 and reflections along the x- and y-axes. g_0 is the relative pose between the board and our vantage point. Right: an image I_1 taken at location o_1. Notice that the image would be identical if it was taken at o_2 instead (I_2).

Here, something comes for free. Remember, before you do 8-point algorithm, you need to detect and find corresponding points. Look here, you get correspondence for free. It's the same image. For a square, you get eight views. It's the same image, but you shuffle the order of the corners.

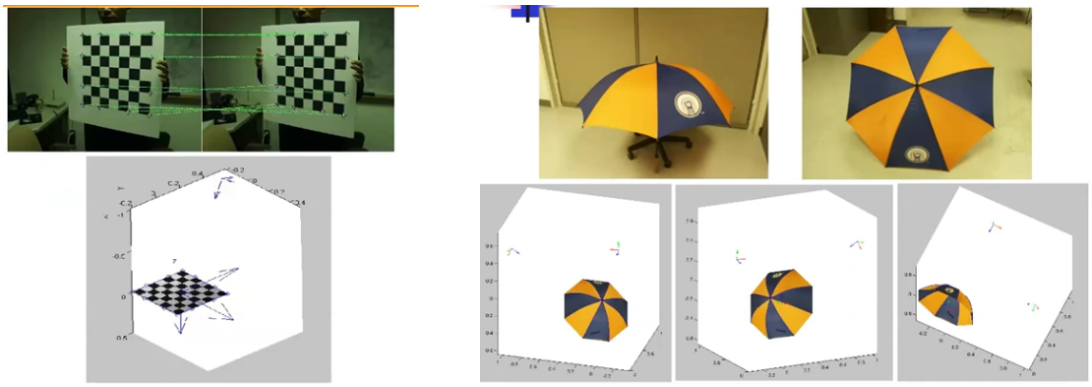
Now, you can recover the camera pose. That's why you know exactly the camera pose. All the geometry from multiview applies here. We call them "virtual" images, but they are as good as the real one.

You can apply this to reflective and translational symmetry also.

There's something you also get other than correspondence. You also get the baseline for free with these virtual images. You don't have to move.

There is so much information in the 3D world. All the virtual images produced by the same image are related by the multiview matrix.

But to get this for free, you can only get symmetry if you have gotten the recognition correct. The most powerful part of geometry right now is the recognition part. It's a recognition problem. Once that's done, geometry is simple. Correspondence comes for free, big baseline comes for free. Even the R, T in the multiview matrix, all of them are constrained. There is only ONE unique solution.



(a) Symmetry-based reconstruction (reflection

(b) Symmetric Curves and Surfaces

Figure 16.13: If you knew something was symmetric, you can reconstruct it off one image with pixel-wise accuracy which you cannot get from SLAM. Symmetry gives you accuracy good enough for AR. For the umbrella, if you know the curve, you can reconstruct it and the camera pose perfectly.

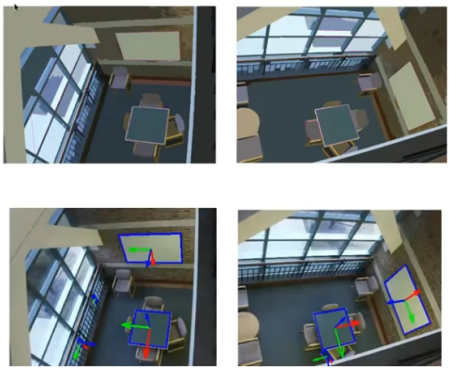
In order to associate features, points, lines, planes, you need to solve the recognition problem. You need to recognize frame to frame, or which points form the same object.

The problem is about how you can extract, detect, match symmetric objects across images.

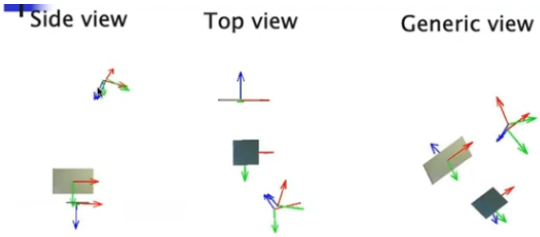
The naive approach 20 years ago without learning:

1. Color based segmentation
2. Polygon fitting
3. Symmetry verification
4. single view recovery - if we know something is symmetric, we can recover the view.

5. find the set of camera pose consistent with symmetric object. There are multiple symmetric object, find the camera motion that gives you the pose for all symmetric objects.



(a) Top shows initial pair of images. Bottom shows images annotated with symmetric objects. Views taken very carefully by ONLY rotating the camera. There is no translation and no baseline.



Length ratio	Reconstruction	Ground truth
Whiteboard	1.506	1.51
Table	1.003	1.00

(b) Reconstruction

Figure 16.14: Reconstruction results of the objects. It recovered the objects to the same scale compared to the ground truth.

These are results from 20 years ago. Shows how much things are missing from modern robots.

Not only can you recover pose, you can calibrate from symmetry.

What you saw was a dinosaur approach to verify that it works.

To summarize,

1. Multiple (perspective) images = multiple-view rank condition - They are all variations on multi-view rank matrices (rank = 0, 1, 2, 3) and all of the constraints.
2. single image + symmetry = "multi-view" rank condition - It is the same as above except that the multi view is special. Images are virtual and interpreted. Correspondence, baseline, structure come for free.
3. multiple images + symmetry = rank condition + scale correction - Again. Same as above. Now you get scale.
4. matching + symmetry = rank condition + scale correction + clique identification. Careful about matching symmetry.

16.6 Extra stuff

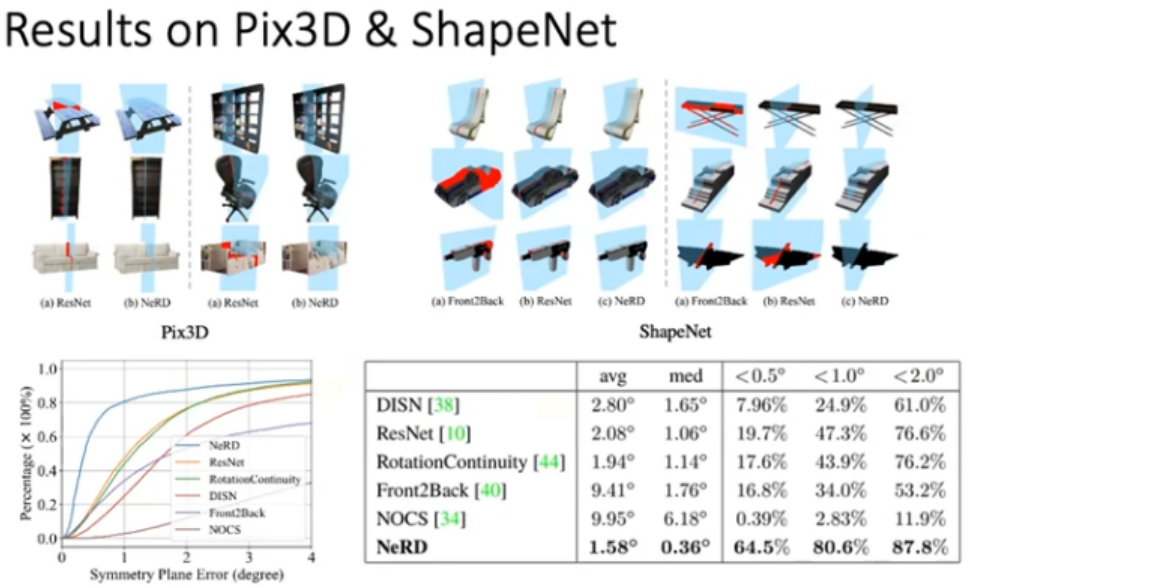
This is a sneak peek of 294. Only recently have we started to investigate.

There are limitations of traditional 3D reconstruction. Textureless objects, transparency, repetitive patterns, large baseline, moving objects all make life harder.

There are deep learning approaches to get pose, point clouds etc, but they haven't solved 3D problems. NN based 3D vision not even as good as nearest neighbors.

The hard part of the modern problem is recognition. Where is the correspondence? Where are the structures? Where are the primitives?

Learning and geometry are complimentary.



	avg	med	<0.5°	<1.0°	<2.0°
DISN [38]	2.80°	1.65°	7.96%	24.9%	61.0%
ResNet [10]	2.08°	1.06°	19.7%	47.3%	76.6%
RotationContinuity [44]	1.94°	1.14°	17.6%	43.9%	76.2%
Front2Back [40]	9.41°	1.76°	16.8%	34.0%	53.2%
NOCS [34]	9.95°	6.18°	0.39%	2.83%	11.9%
NeRD	**1.58°**	**0.36°**	**64.5%**	**80.6%**	**87.8%**

Yichao Zhou, Shichen Liu, Yi Ma (2020). Learning to Detect 3D Reflection Symmetry for Single-View Reconstruction. Technical Report. arXiv:2006.10042 [cs.CV].

Figure 16.15: Here are the results of combining geometry and learning. Improvement is dramatic. That's it. Lessons for learning people and geometry people. If you don't put them together, maybe you'll be a little better than nearest neighbors.

Autonomous driving, service robot in manmade environment, none of this structure is being used for localization, mapping, navigation.

Tango, Magic Leap. None of them are using these advances.