CIS 5800

Machine Perception

Instructor: Lingjie Liu

Lec 12: March 5, 2025

Administrivia

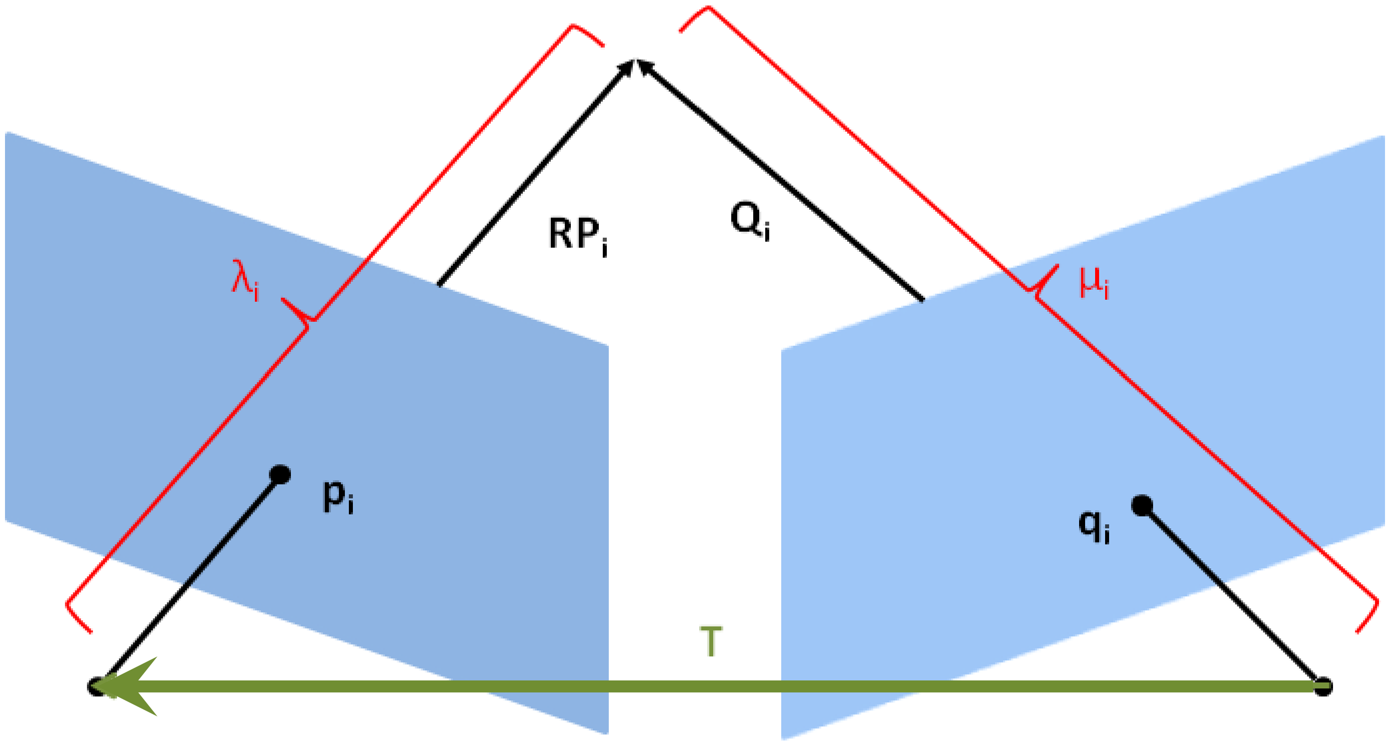
- Midterm exam coming up
  - Date reminder: Wednesday March 26 (class hours, we'll start at 12 noon exactly)
  - Syllabus: All material covered in class until Wed March 19.
  - Review questions pdf will be released on Canvas by this week, solutions will be out on Mon March 17, review lecture on Mon March 24. Try solving over the Spring Break!
  - If you are unable to attend the midterm exam in person on March 26, please complete the form by March 17: https://forms.gle/TCWYHAYn4Hd324BN9

  - Also, you need to contact the Weingarten Office for academic accommodations and send me the paperwork or approval from the Weingarten Office.

Administrivia

- Next week is Spring Break, no class.
- HW 2 deadline is tomorrow.

Recap: Two Calibrated Views of the Same 3D Scene

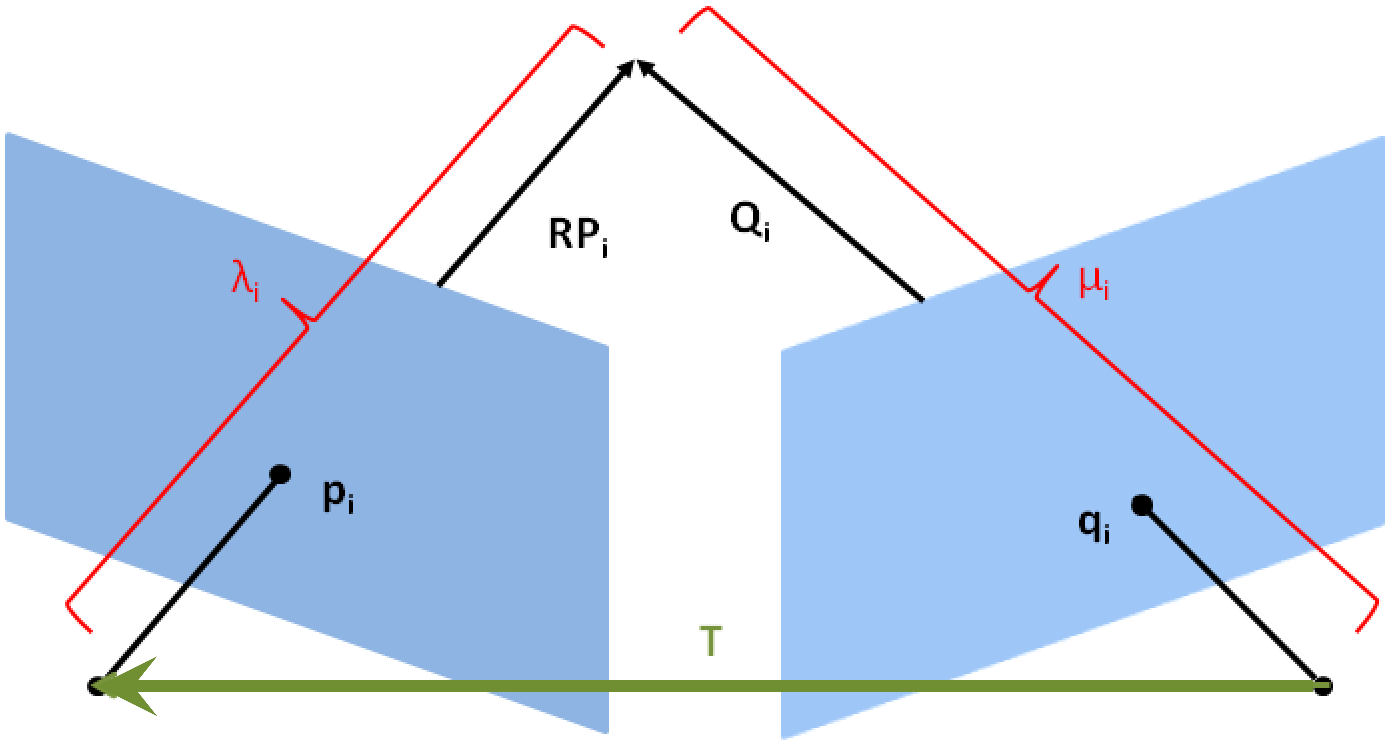


$$R(\lambda p) + T \ = \mu q$$

Given 2D correspondences (p,q)

Find motion R, T and depths λ, μ.

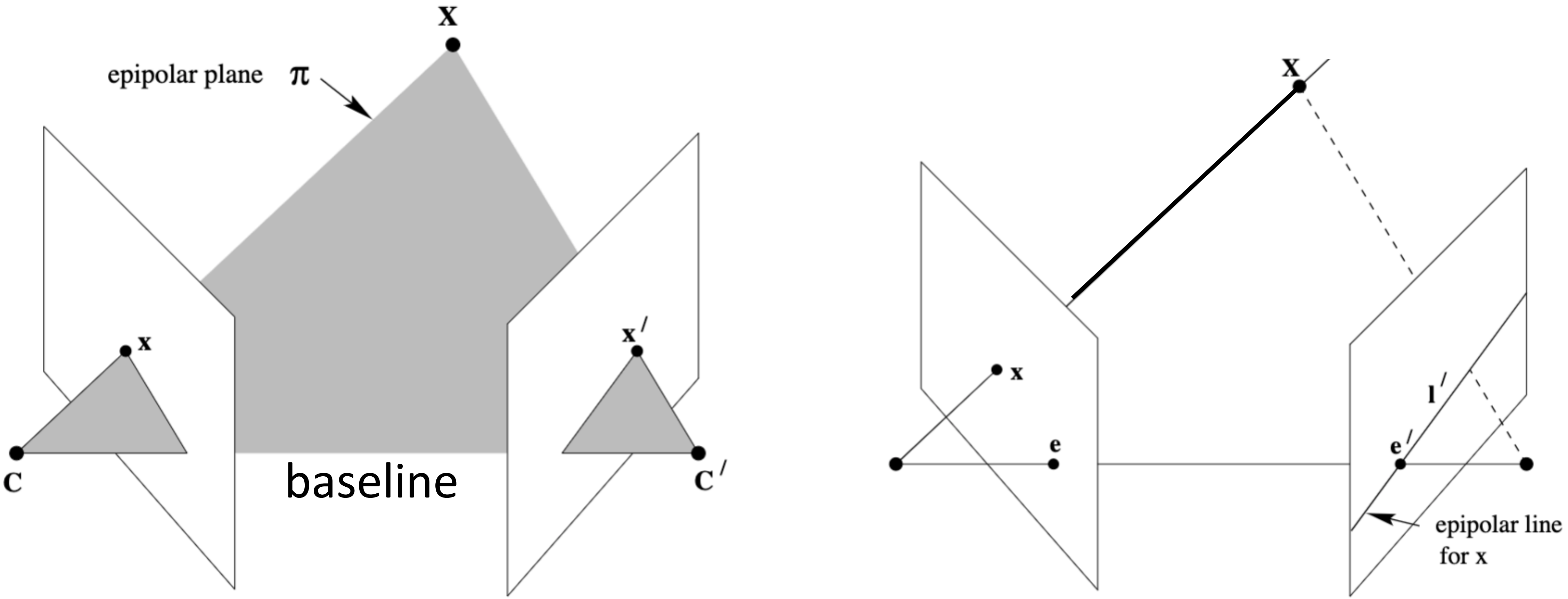
Recap: "Epipolar Constraints" Between Two Views of a Scene



We can eliminate the depths from $R(\lambda p) + T = \mu q$ and obtain the epipolar constraint:

$$\boldsymbol{q}_i^T (T \times R\boldsymbol{p}_i) = 0$$

Recap: Epipolar “Planes” and “Lines”



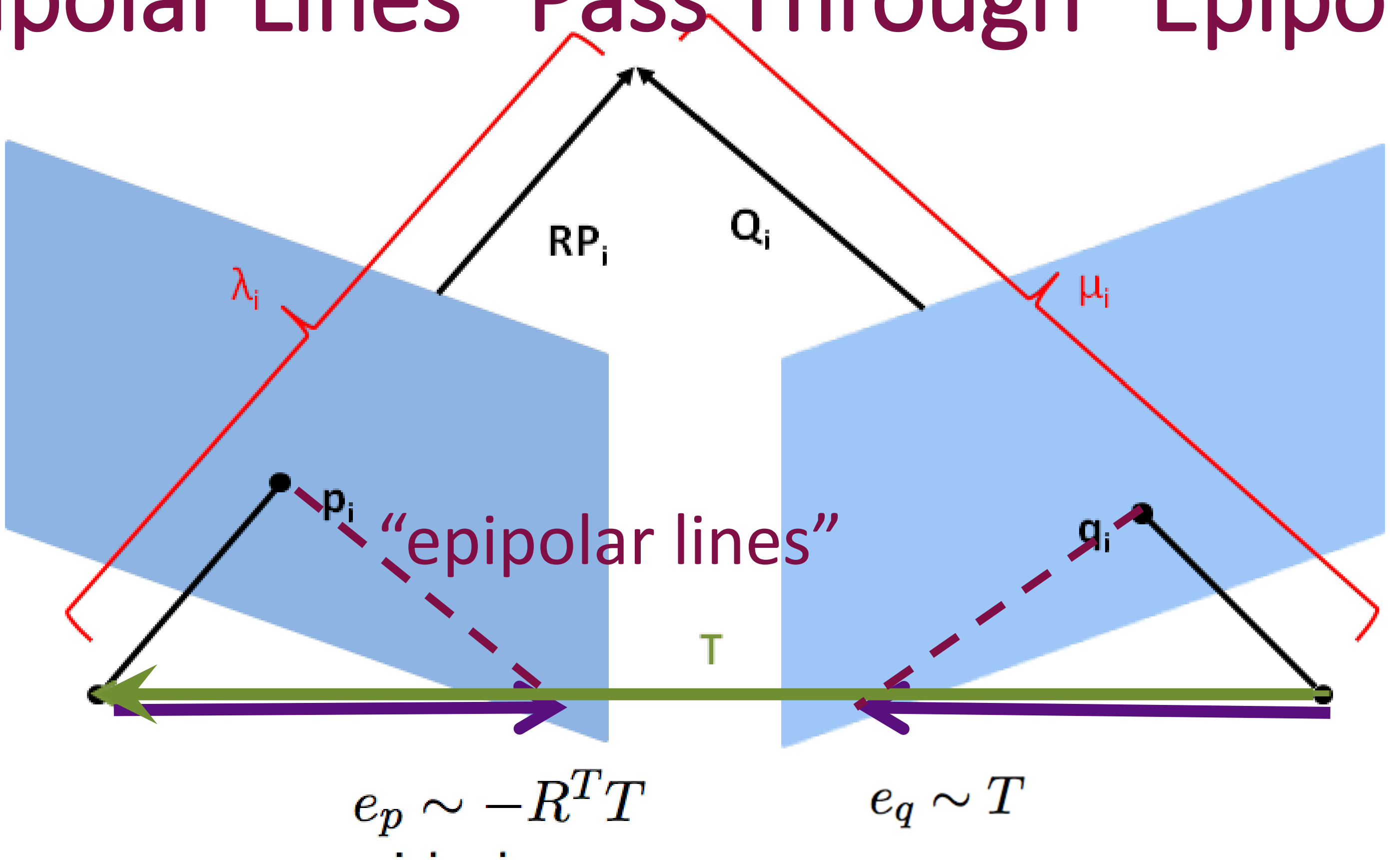
Epipolar planes are planes containing the baseline.

Any 3D point induces a corresponding epipolar plane.

Intersection of this plane with an image plane = epipolar line for that 3D point.

ZH Fig 9.1

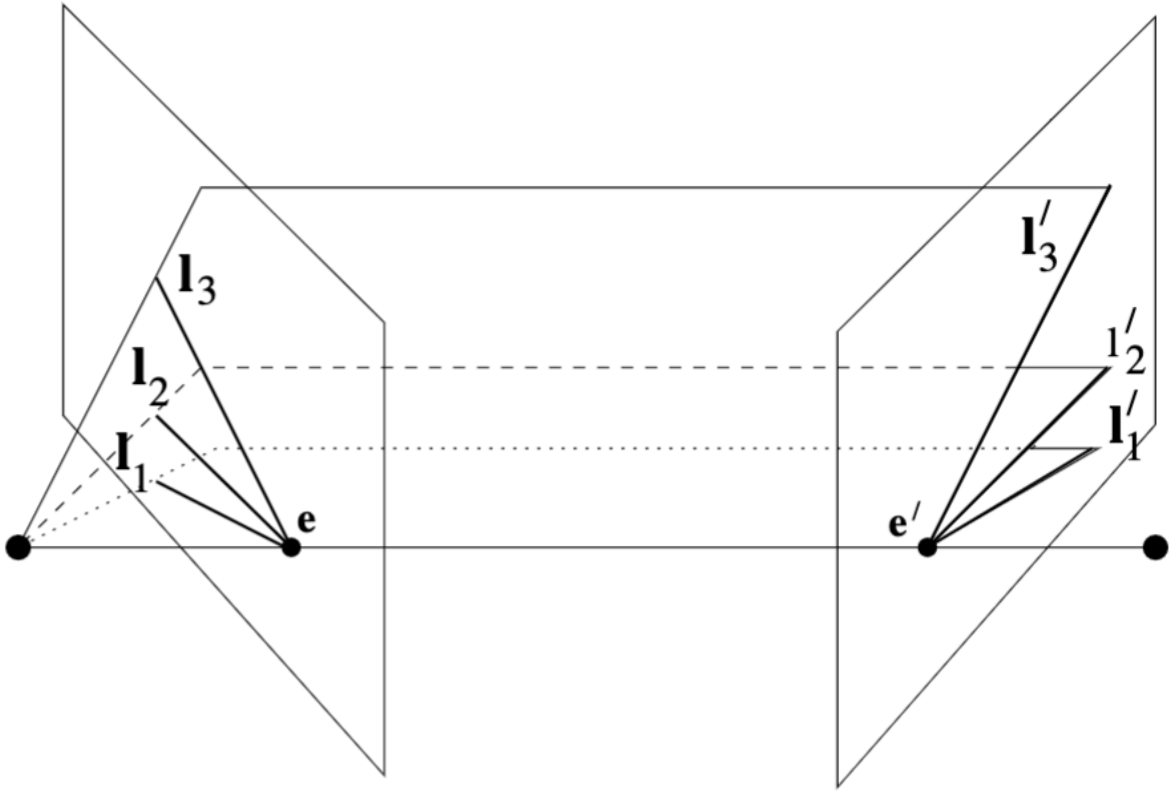
Recap: "Epipolar Lines" Pass Through "Epipoles"



$e_p \sim -R^T\ T$ and $e_q \sim T$ are the "epipoles" = images of the other camera center on each plane = intersections of baseline T with the two planes = VP of the translation direction in each plane.

All epipolar lines in each image plane pass through its epipole.

Recap: Epipolar “Pencils”



The epipolar lines in each image plane form a “pencil” whose tip is the epipole i.e. image of the other camera center / intersection of baseline with image plane. (Because all epipolar planes contain the baseline, after all)

Recap: Epipolar Constraints

$$\boldsymbol{q}_i^T(T \times R\boldsymbol{p}_i) = 0$$

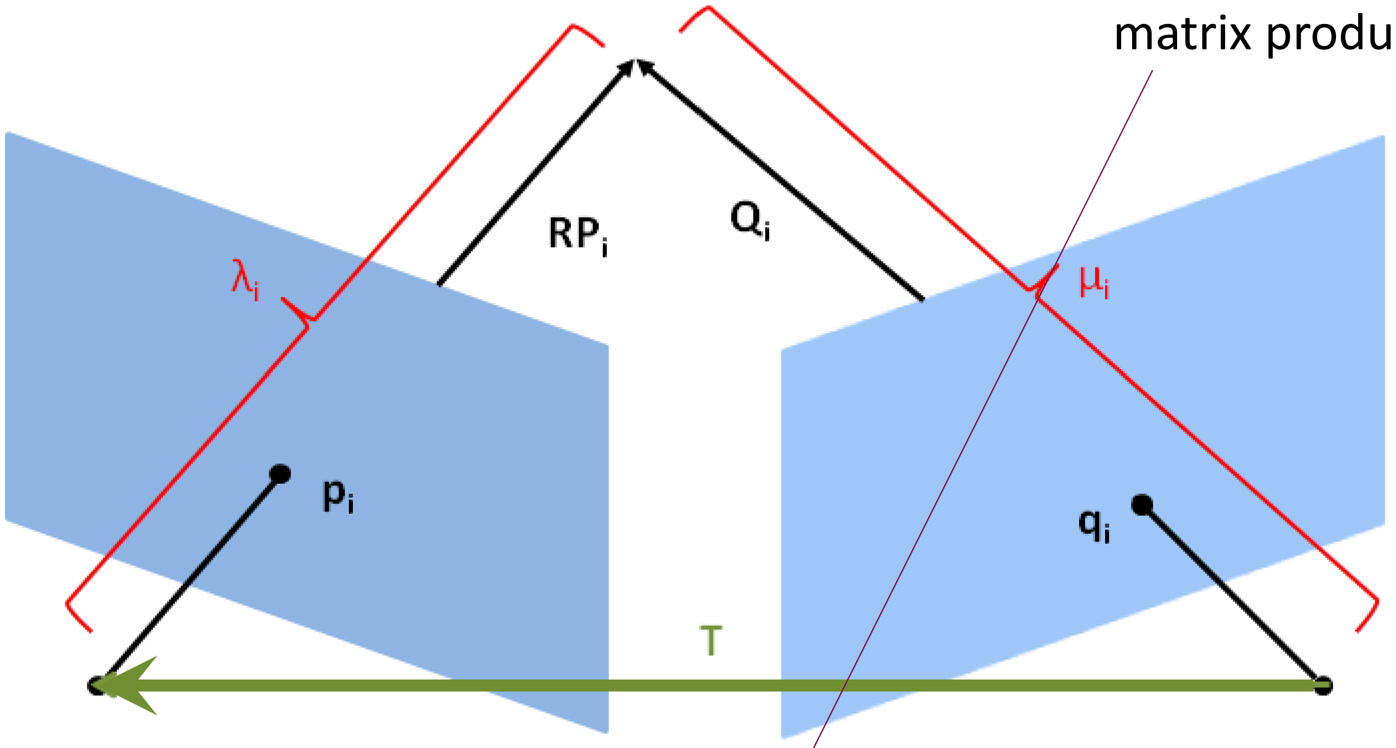
Gives us an equation in unknowns: R, T => route to an SfM solution?

Q: If this were linear in R, T, we could solve it with enough (p_i, q_i) pairs i.e. 2D->2D correspondences. Is it linear though?
A: No, but through a change of variables, we will soon make the equation linear in some new unknowns.

Also note: we are not only interested in R, T (the "motion"). We also want to solve for the depths λ_i, μ_i afterwards (the "structure").

Epipolar Constraints

Want to solve for R, T.
Can we make this look more like standard linear equations containing matrix products etc.?



We can eliminate the depths from $R(\lambda p) + T = \mu q$ and obtain the epipolar constraint:

$$q^T(T \times Rp) = 0$$

Review: Cross-products through skew-symmetric matrices

The vector cross product also can be expressed as the product of a skew-symmetric matrix and a vector:

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Sometimes written as $\hat{a}$. Remember, this is now a 3×3 matrix, while original a, b were 3x1 vectors.

Wikipedia on cross product

The Essential Matrix E

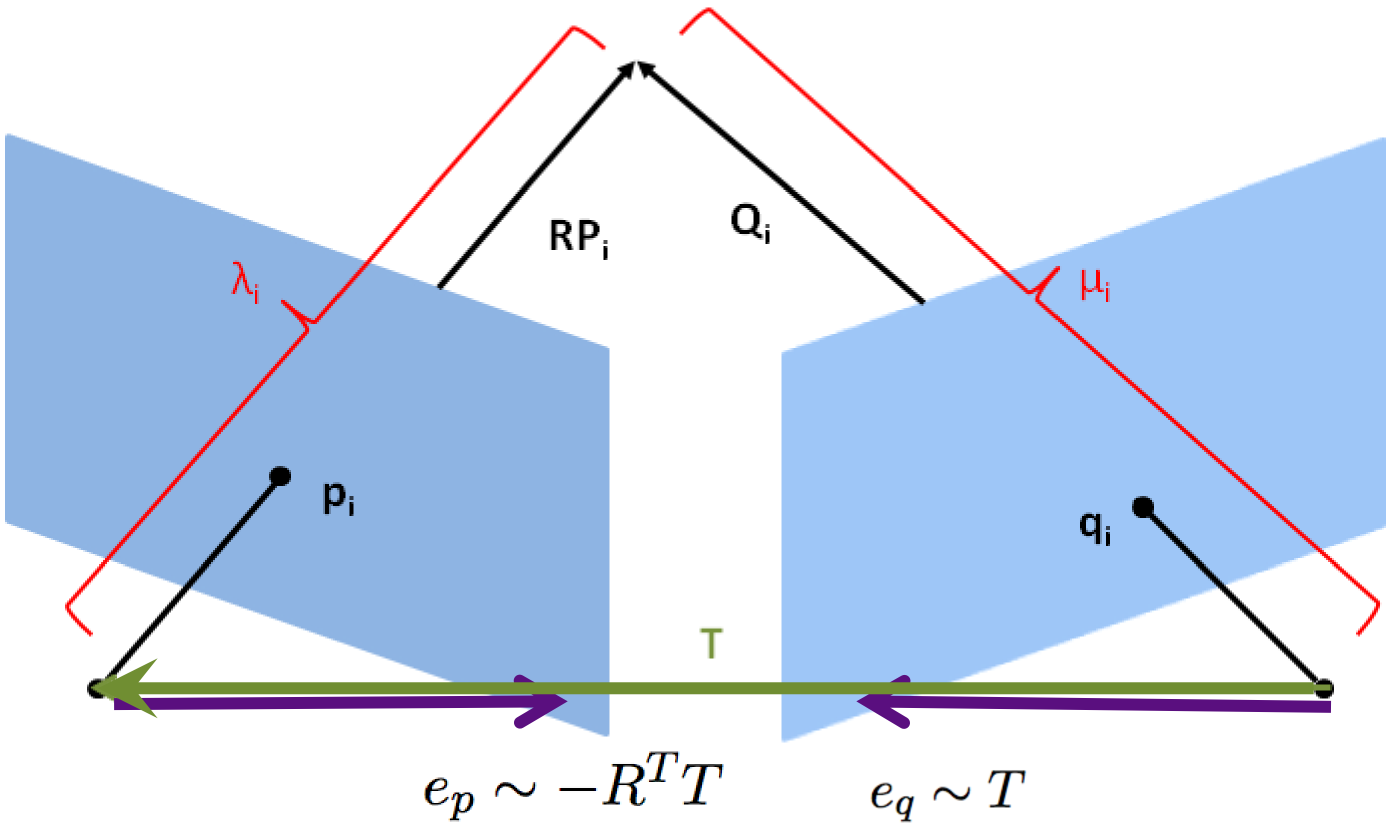
We had: $\boldsymbol{q}_i^T(T \times R\boldsymbol{p}_i) = 0$

$$\Rightarrow \boldsymbol{q}_i^T(\hat{T}R)\ \boldsymbol{p}_i = 0$$

Renaming $E = (\hat{T}R)$:

$$\boldsymbol{q}_i^T E\ \boldsymbol{p}_i = 0$$

"Essential matrix"



Now linear in the new unknowns $E_{3\times3}$! But will need to recover $T_{3\times1}$, $R_{3\times3}$ later.

Is the epipolar constraint really linear in E?

$q_i^T E_{3\times3} p_i = 0$ is a single equation that is linear in the elements of E
Can write this out explicitly as below.

If

$$E = \begin{pmatrix} e_1 & e_2 & e_3 \end{pmatrix}$$

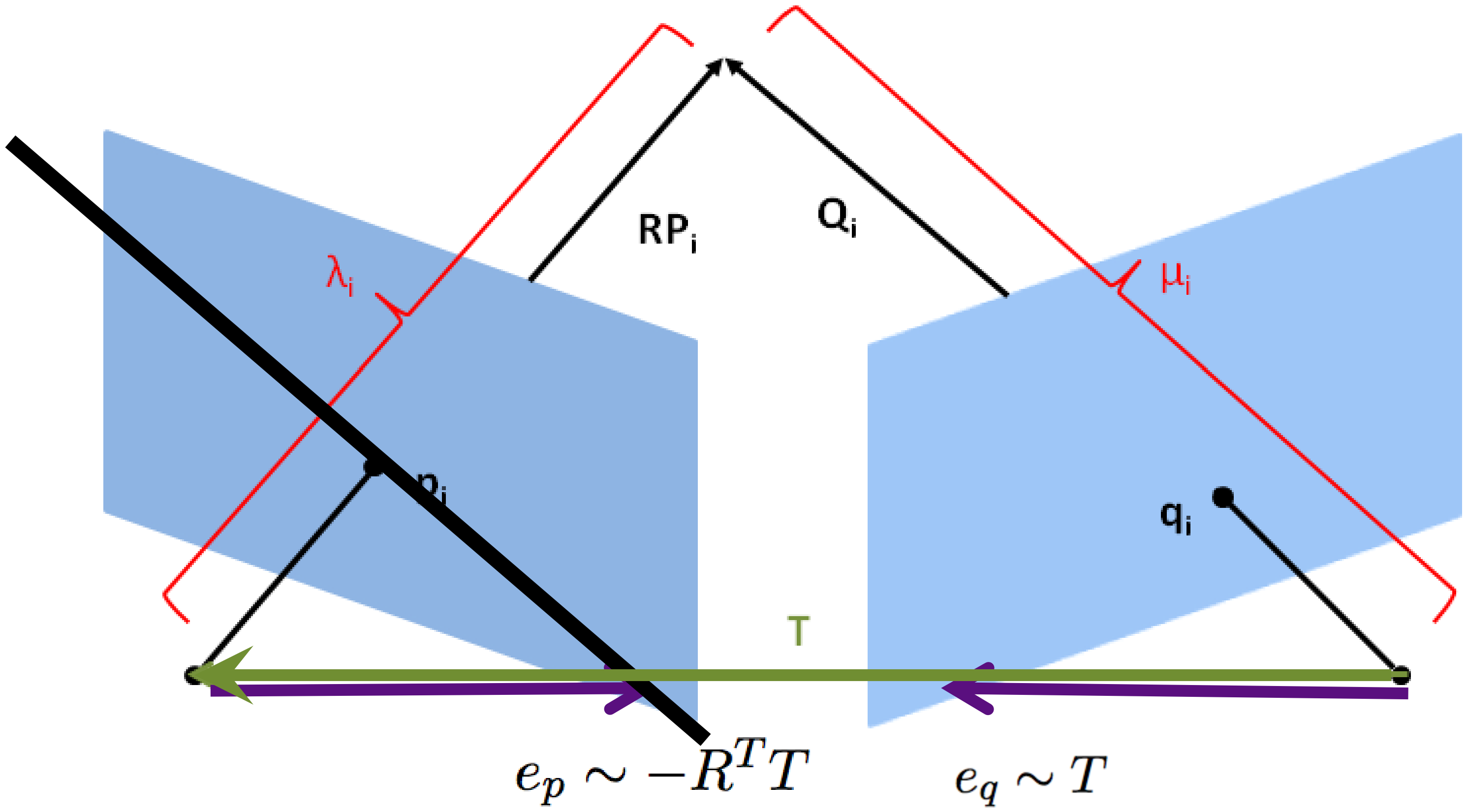
then epipolar constraint can be rewritten as

$$\begin{aligned} q^T \begin{pmatrix} e_1 & e_2 & e_3 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} &= q^T \left(p_x e_1 + p_y e_2 + p_z e_3\right) \\ &= \begin{pmatrix} p_x q^T & p_y q^T & p_z q^T \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = 0 \end{aligned}$$

This equation is linear

Longuet-Higgins 1981

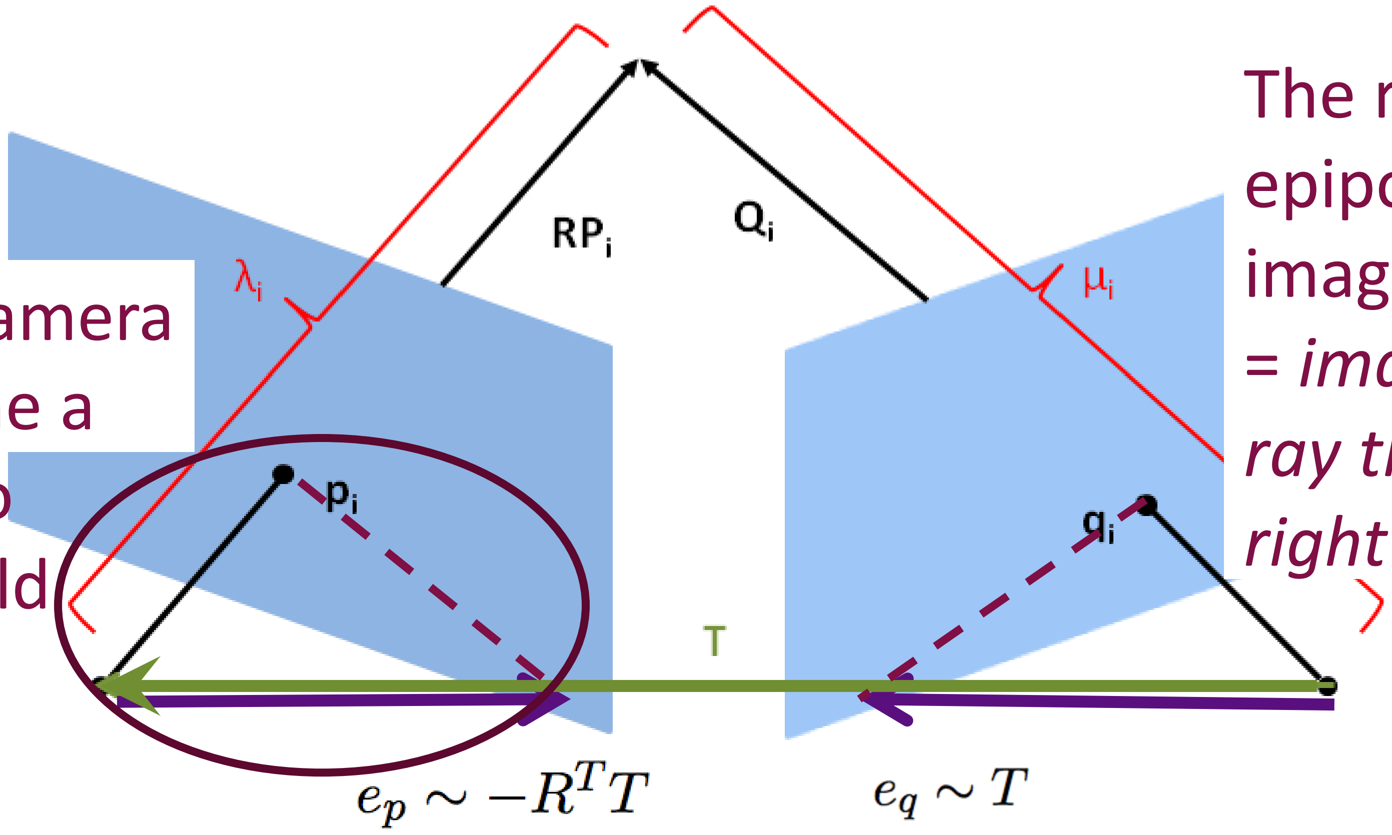
Epipolar Lines in Essential Matrix Notation



Equation $q^T E p = 0$ is a line equation in the p-plane with line coefficients $E^T q$. It is called the epipolar line in p-plane.

Epipolar Lines Constrain Point Correspondences!

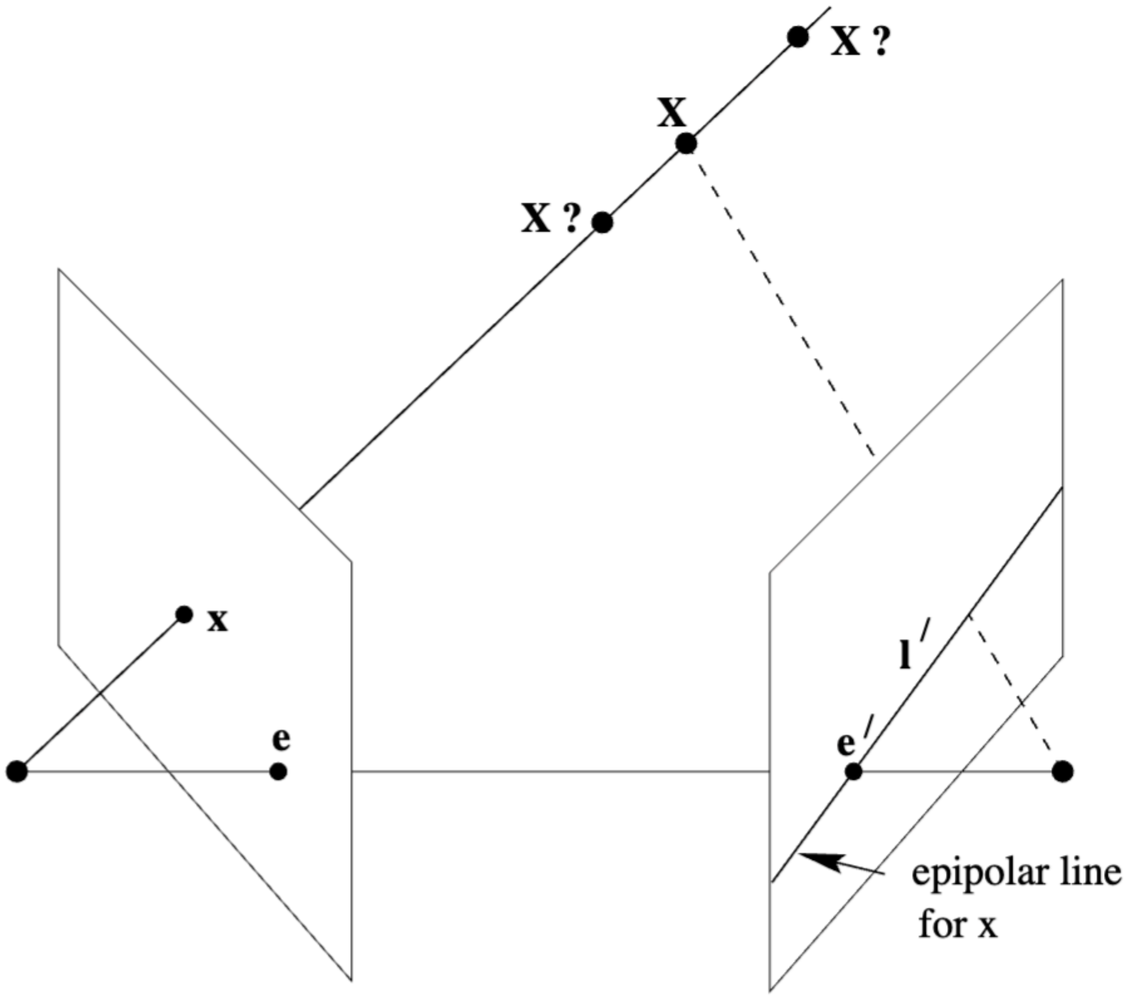
p_i and the two camera centers determine a plane (which also contains the world point P_i)



The relevant right epipolar line is the image of this plane = *image of the left ray through the right camera.*

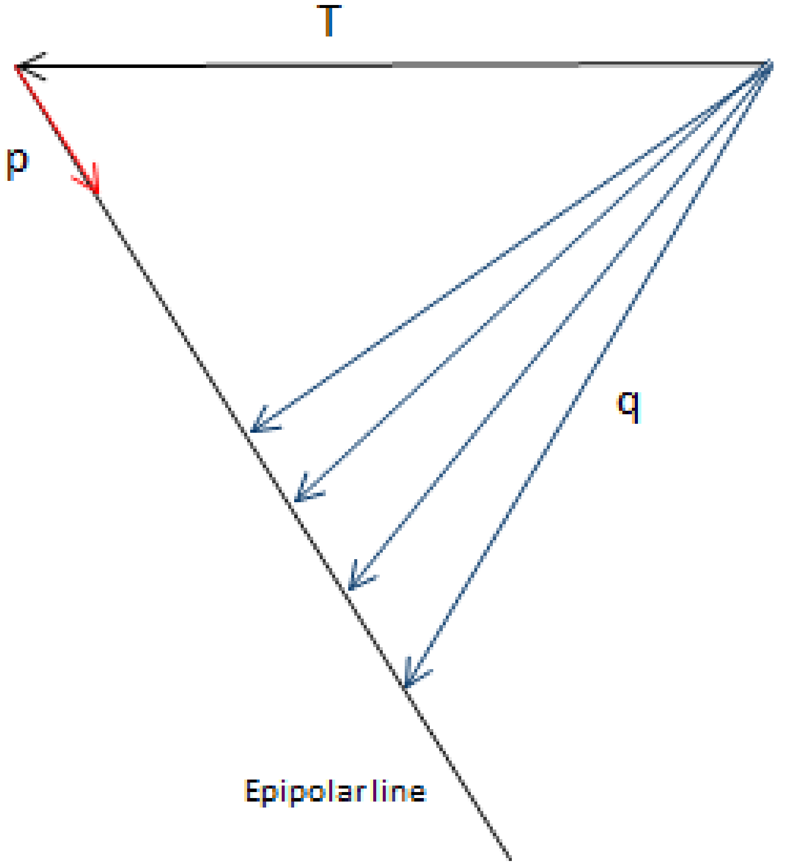
Given a point p = left image of a world point, the right image of that point is constrained to lie on the corresponding right epipolar line, and vice versa. (depending on depth of the world point).

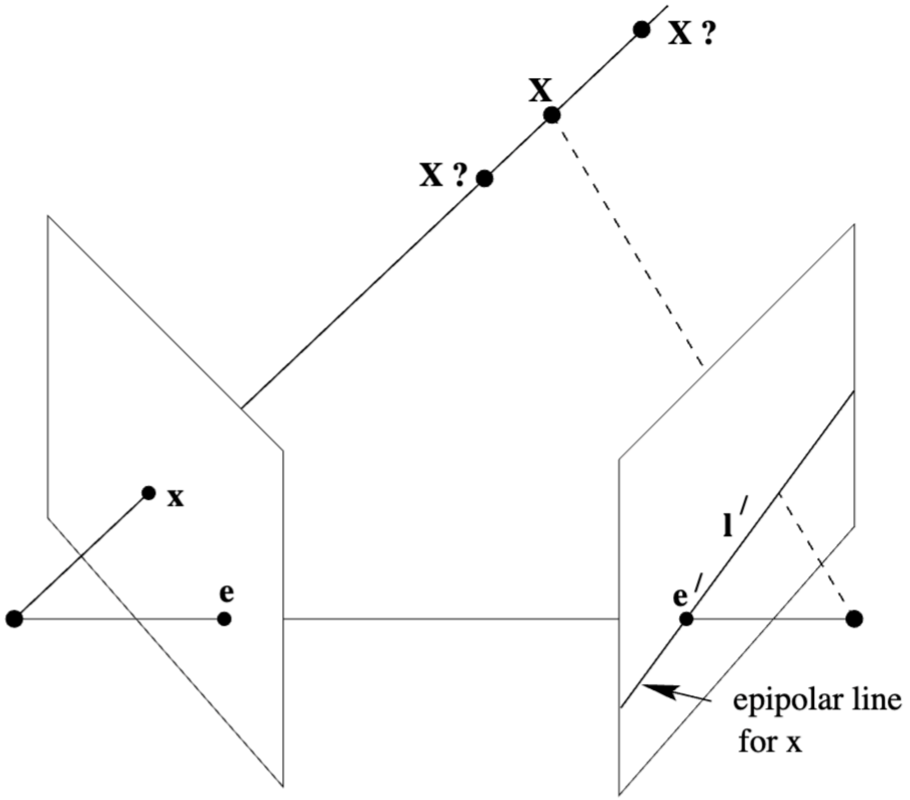
Epipolar Lines Constrain Point Correspondences!



ZH Ch9

Epipolar Lines Constrain Point Correspondences!

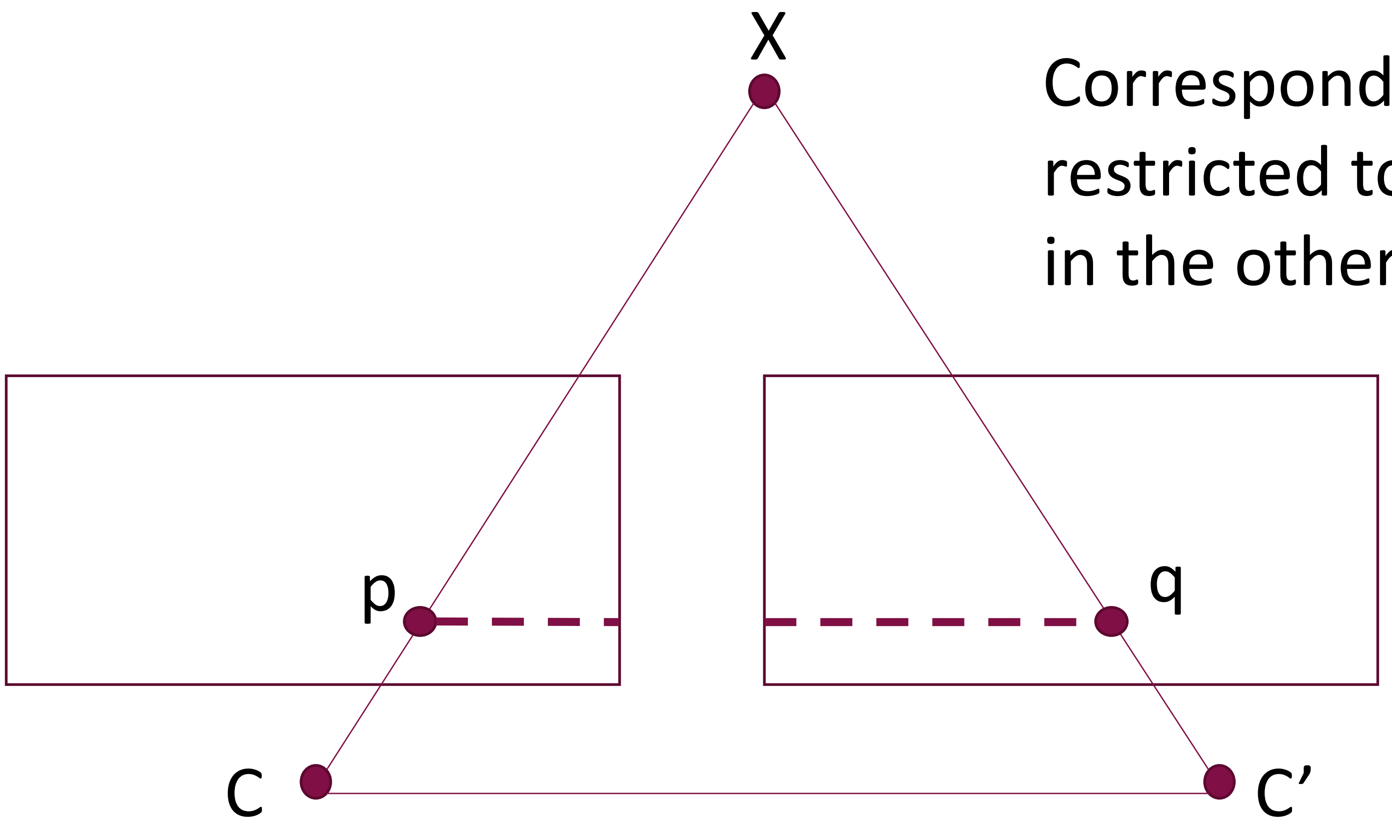




Knowledge of the E-matrix allows us to search for points q corresponding to points p along the epipolar line, reducing correspondence to $1D$-search.

Position of the corresponding point q along epipolar line varies with depth of the 3D points which is still constrained to lie on the ray through p.

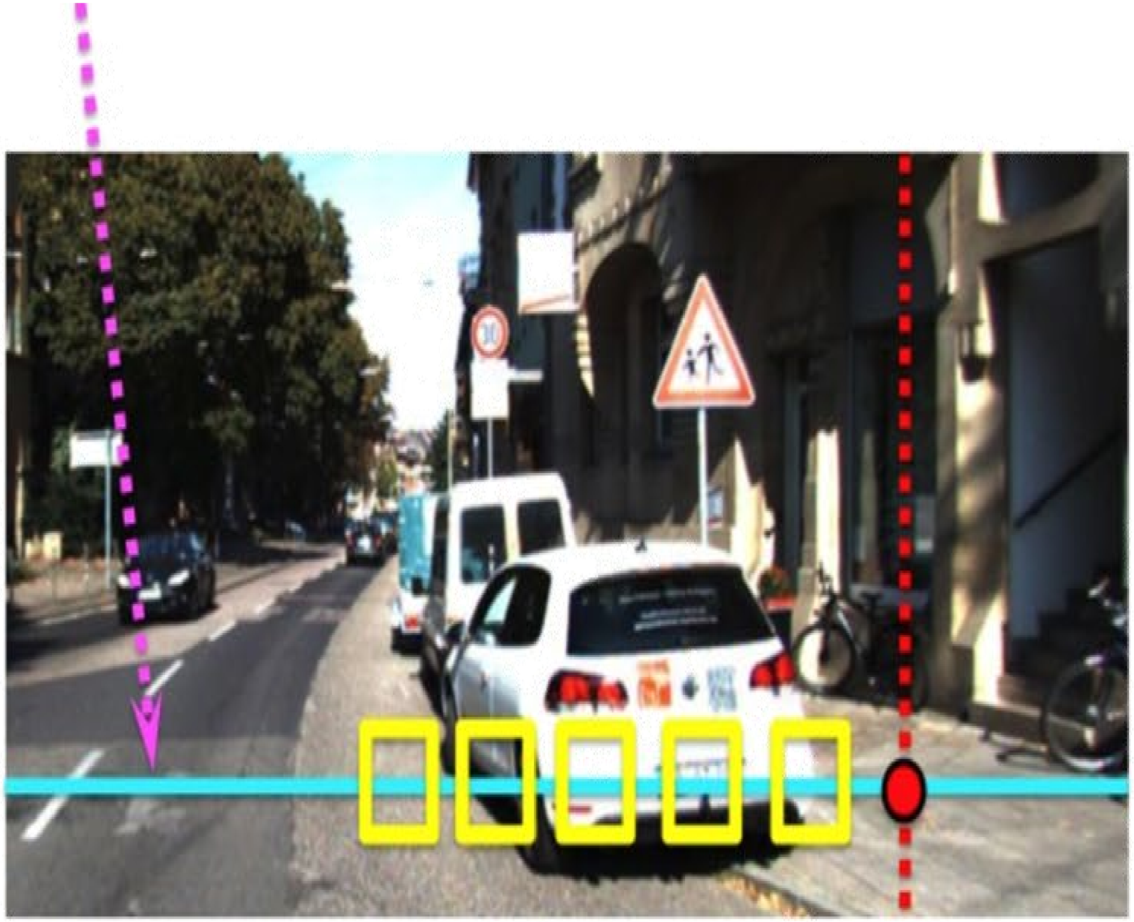
Special case: “frontoparallel” / “parallel stereo” cameras



Correspondences are restricted to the “same” row in the other image!

Q: Where is the epipole = image of other camera center in each image?
A: Baseline is parallel to image plane(s) => epipole is at infty along x, and epipolar line is horizontal.

What if the cameras are “frontoparallel” / “parallel stereo”?



left image

right image

Longuet-Higgins' 8-Point Algorithm

Hugh Christopher Longuet-Higgins, quantum chemist turned cognitive scientist. 1981

Also see: [Hartley 1997: "In defense of the 8-point algorithm"]

We thank D. Cassidy for sampling of the RISP cores, L. Burckle for assistance with diatom taxonomy, K. Austin for sample preparation and G. H. Denton and T. J. Hughes for discussions and for reading drafts of the manuscript. P. Whiting drafted the figure and C. McGowen typed the manuscript. This work was supported by NSF grants DPP-7721083 A01 and DPP-7920112.

Received 7 October 1980; accepted 30 June 1981.

1. Kellogg, T. B., Osterman, L. E. & Stuiver, M. *J. Foram. Res.* **9**, 322–335 (1979).
2. Kellogg, T. B. & Truesdale, R. S. *Mar. Micropalaeont.* **4**, 137–158 (1979).
3. Kellogg, T. B., Truesdale, R. S. & Osterman, L. E. *Geology* **7**, 249–253 (1979).
4. Opdyke, N. D. *Rev. Geophys. Space Phys.* **10**, 213–249 (1972).
5. McCollum, D. W. *Init. Rep. DSDP* **28**, 515–571 (1975).
6. Brady, H. T. & Martin, H. *Science* **203**, 437–438 (1979).
7. Webb, P. N., Ronan, T. E. Jr, Lipps, J. H. & DeLaca, T. E. *Science* **203**, 435–437 (1979).
8. Webb, P. N. *RISP tech. Rep.* 78–1 (University of Nebraska, 1978); *RISP tech. Rep.* 79–1 (University of Nebraska, 1979).
9. Brady, H. T. *Antarct. J. U.S.* **13**, 123–124 (1978); **14**, 130 (1979).
10. Truesdale, R. S. & Kellogg, T. B. *Mar. Micropalaeont.* **4**, 13–31 (1979).
11. Ling, H. Y. & White, R. J. *Antarct. J. U.S.* **14**, 126–127 (1979).
12. Lipps, J. H., Ronan, T. E. Jr & DeLaca, T. E. *Science* **203**, 447–449 (1979).
13. Schrader, H. J. *Nova Hedwigia* **45**, Suppl. 403–409 (1974).
14. Schrader, H. J. *Init. Rep. DSDP* **18**, 673–797 (1973); **35**, 605–671 (1976).
15. Fenner, J. *Init. Rep. DSDP* **39**, 491–623 (1977).
16. Webb, P. N. *Mem. natn. Inst. Polar Res.* **13**, 148–149 (1979).
17. Jacobs, S. S., Gordon, A. L. & Ardai, J. L. Jr *Science* **203**, 439–443 (1979).
18. Stuiver, M., Denton, G. H., Hughes, T. J. & Fastook, J. L. in *The Last Great Ice Sheets* (eds Denton, G. H. & Hughes, T. J.) 319–436 (Wiley, New York, 1981).
19. Kellogg, D. E., Stuiver, M., Kellogg, T. B. & Denton, G. H. *Palaeogeogr. Palaeoclimatol. Palaeoecol.* **30**, 157–189 (1980).
20. Boulton, G. S. *J. Glaciol.* **24**, 244 (1979).
21. Webb, P. N. & Brady, H. T. *EOS* **59**, 309 (1978).
22. Webb, P. N. *Mem. natn. Inst. Polar Res.* **13**, 206–212 (1979).

A computer algorithm for reconstructing a scene from two projections

H. C. Longuet-Higgins

Laboratory of Experimental Psychology, University of Sussex, Brighton BN1 9QG, UK

A simple algorithm for computing the three-dimensional structure of a scene from a correlated pair of perspective projections is described here, when the spatial relationship between the two projections is unknown. This problem is relevant not only to photographic surveying[1] but also to binocular vision[2], where the non-visual information available to the observer about the orientation and focal length of each eye is much less accurate than the optical information supplied by the retinal images themselves. The problem also arises in monocular perception of motion[3], where the two projections represent views which are separated in time as well as space. As Marr and Poggio[4] have noted, the fusing of two images to produce a three-dimensional percept involves two distinct processes: the establishment of a 1:1 correspondence between image points in the two views—the 'correspondence problem'—and the use of the associated disparities for determining the distances of visible elements in the scene. I shall assume that the correspondence problem has been solved; the problem of reconstructing the scene then reduces to that of finding the relative orientation of the two viewpoints.

Photogrammetrists know that if a scene is photographed from two viewpoints, then the relationship between the camera positions is uniquely determined, in general, by the photographic coordinates of just five distinguishable points; but actually calculating the structure of the scene from five sets of image coordinates involves the iterative solution of five simultaneous third-order equations[1]. I show here that if the scene contains as many as eight points whose images can be located in each projection, then the relative orientation of the two projections, and the structure of the scene, can be computed, in general, from the eight sets of image coordinates by a direct method which calls for nothing more difficult than the solution of a set of simultaneous linear equations.

Let P be a visible point in the scene, and let (X_1, X_2, X_3) and (X'_1, X'_2, X'_3) be its three-dimensional cartesian coordinates with respect to the two viewpoints. The 'forward' coordinates X_3 and X'_3 are necessarily positive. The image coordinates of P in the two views may then be defined as

$$(x_1, x_2) = (X_1/X_3, X_2/X_3), \quad (x'_1, x'_2) = (X'_1/X'_3, X'_2/X'_3) \tag{1}$$

and it is convenient to supplement them with the dummy coordinates

$$x_3 = 1, \qquad x'_3 = 1 \tag{2}$$

so that one can then write

$$x_\mu = X_\mu/X_3, \qquad x'_\nu = X'_\nu/X'_3 \qquad (\mu, \nu = 1, 2, 3) \tag{3}$$

As the two sets of three-dimensional coordinates are connected by an arbitrary displacement, we may write

$$X'_\mu = \mathbf{R}_{\mu\nu}(X_\nu - \mathbf{T}_\nu) \tag{4}$$

where $\mathbf{T}$ is an unknown translational vector and $\mathbf{R}$ is an unknown rigid rotation matrix. (In this and subsequent equations I sum over repeated Greek subscripts.) The rotation $\mathbf{R}$ satisfies the relationships

$$\mathbf{R}\tilde{\mathbf{R}} = \mathbf{1} = \tilde{\mathbf{R}}\mathbf{R}, \qquad \det \mathbf{R} = 1 \tag{5}$$

and it is convenient to adopt the length of the vector $\mathbf{T}$ as the unit of distance:

$$\mathbf{T}_\nu^2 (= \mathbf{T}_1^2 + \mathbf{T}_2^2 + \mathbf{T}_3^2) = 1 \tag{6}$$

I begin by establishing a general relationship between the two sets of image coordinates—a relationship which expresses the condition that corresponding rays through the two centres of projection must intersect in space. We define a new matrix $\mathbf{Q}$ by

$$\mathbf{Q} = \mathbf{R}\mathbf{S} \tag{7}$$

where $\mathbf{S}$ is the skew-symmetric matrix

$$\mathbf{S} = \begin{bmatrix} 0 & \mathbf{T}_3 & -\mathbf{T}_2 \\ -\mathbf{T}_3 & 0 & \mathbf{T}_1 \\ \mathbf{T}_2 & -\mathbf{T}_1 & 0 \end{bmatrix} \tag{8}$$

Equation (8) may be written as

$$\mathbf{S}_{\lambda\nu} = \varepsilon_{\lambda\nu\sigma}\mathbf{T}_\sigma \tag{9}$$

where $\varepsilon_{\lambda\nu\sigma} = 0$ unless (λ, ν, σ) is a permutation of (1, 2, 3), in which case $\varepsilon_{\lambda\nu\sigma} = \pm 1$ depending on whether this permutation is even or odd. It follows from equations (4)–(9) that

$$X'_\mu \mathbf{Q}_{\mu\nu} X_\nu = \mathbf{R}_{\mu\kappa}(X_\kappa - \mathbf{T}_\kappa)\mathbf{R}_{\mu\lambda}\varepsilon_{\lambda\nu\sigma}\mathbf{T}_\sigma X_\nu = (X_\lambda - \mathbf{T}_\lambda)\varepsilon_{\lambda\nu\sigma}\mathbf{T}_\sigma X_\nu \tag{10}$$

but because the quantity $\varepsilon_{\lambda\nu\sigma}$ is antisymmetric in every pair of its subscripts, the right-hand side vanishes identically:

$$X'_\mu \mathbf{Q}_{\mu\nu} X_\nu = 0 \tag{11}$$

Dividing equation (11) by $X'_3 X_3$ we arrive at the desired relationship between the image coordinates:

$$x'_\mu \mathbf{Q}_{\mu\nu} x_\nu = 0 \tag{12}$$

The next step is to determine the nine elements $\mathbf{Q}_{\mu\nu}$. There will be one equation of type (12) for every point P_i, namely

$$(x'_\mu x_\nu)_i \mathbf{Q}_{\mu\nu} = 0 \tag{13}$$

and in this equation the nine quantities $(x'_\mu x_\nu)_i$ are presumed to be known. The ratios of the nine unknowns $\mathbf{Q}_{\mu\nu}$ can therefore be obtained, in general, by solving eight simultaneous linear equations of type (13), one for each of eight visible points $P_1, \ldots, P_8$. I shall not yet discuss the special circumstances under which the solution fails; for the present merely note that if the eight equations (13) are independent, their solution is entirely straightforward from a computational point of view.

0028–0836/81/370133—03$01.00

8-Point Algorithm

- Recall that each correspondence gives us one linear equation in the unknowns E

Is this really linear in E?

$q_i^T E_{3\times3} p_i = 0$ is a single equation that is linear in the elements of E
Can write this out explicitly as below.

If

$$E = \begin{pmatrix} e_1 & e_2 & e_3 \end{pmatrix}$$

then epipolar constraint can be rewritten as

$$q^T \begin{pmatrix} e_1 & e_2 & e_3 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = q^T \left(p_x e_1 + p_y e_2 + p_z e_3 \right)$$

$$= \boxed{\begin{pmatrix} p_x q^T & p_y q^T & p_z q^T \end{pmatrix}} \boxed{\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix}} = 0$$

This equation is linear

“$\boldsymbol{a}_{1\times9}$”

“$\boldsymbol{E'}_{9\times1}$”

Longuet-Higgins 1981

8-Point Algorithm

Let $\vec{a} = \begin{pmatrix} p_x q^T & p_y q^T & p_z q^T \end{pmatrix}$

$$\begin{pmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_n^T \end{pmatrix}_{n \times 9} E' = 0$$

One row per point correspondence

where a_i is the known 1 x 9 vector of image points and E' is the essential matrix re-organized into a 9 x 1 column vector.

E' has to be in the null-space of $\begin{pmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_n^T \end{pmatrix}$.

Does this remind you of something?
Hint: 4-Point Collineations, PnP, …

Solution: As before, set E' to the last right singular vector of $A_{n \times 9}$

Longuet-Higgins 1981

Review: EigenValue and SVD

EigenValue, SVD

Eigenvalue $X\vec{v} = \sigma\vec{v}$

Eigenvalue decomposition $X = U\Sigma U^{-1}$ Here, X is square, U is invertible and Σ is diagonal matrix.

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \cdots & \\ & & & \sigma_n \end{bmatrix} \qquad U = [\vec{u}_1 \quad \vec{u}_2 \quad \dots \quad \vec{u}_n]$$

Here, σ is the eigenvalue of X.
$\vec{u}$ is the eigenvector of X, in unit length.

If X is symmetric $X = U\Sigma U^T$ Here U is orthogonal.

What if X is not square ?

$$X^T X = V\Sigma_1 V^T$$

$$XX^T = U\Sigma_2 U^T$$

SVD $X = U\Sigma V^T$

Penn Engineering

EigenValue, SVD

SVD tells you that a linear transformation is consist of :
rotation + stretching + rotation

$$Xv = U\Sigma V^T v$$

Rotation
Stretch
Rotation

$$\underset{A}{\begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}} = \underset{U}{\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}} \underset{\Sigma}{\begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{bmatrix}} \underset{V^T}{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{18}} & \frac{1}{\sqrt{18}} & -\frac{4}{\sqrt{18}} \\ \frac{2}{3} & -\frac{2}{3} & \frac{1}{3} \end{bmatrix}}$$

Why H is the last column of V ?

Penn Engineering

Rank of A in the 8-Point Algorithm

Problem: What if we have >8 points and $A_{n\times 9}$ is full-rank (i.e. rank 9)?

Not a problem! The last singular vector of A already solves for:

$$\underset{E\prime:||E\prime||_2=1}{\operatorname{argmin}} \; ||AE'||_2 \,,$$

which is the closest thing to finding a null vector for a non-singular matrix

Problem: A may be ***too low-rank***. For example, with perfect measurements, A is rank 8 only if points do not all lie on any “quadric” surface*, including planes, cylinders, ellipsoids etc. Otherwise, rank(A) < 8! This can happen quite frequently in practice, e.g., smartphone moving facing a wall.

No real solution. Avoid relying on correspondences exclusively from such surfaces to the extent possible. If still rank < 8, give up on that pair of views.

*https://tutorial.math.lamar.edu/classes/calciii/quadricsurfaces.aspx

After solving for E, not Quite Done Yet!

$E = \hat{T}R$ has fewer than 8 DOF. T has 3 DOF $(+3)$, R has 3 DOF $(+3)$, and E is scale invariant (-1), so total 5 DOF. **So not all 3x3 matrix is a valid essential matrix.**

- Problem: Given the above, how to ensure that the estimated E is a valid essential matrix?
- Problem: How to decompose E into the $\widehat{T}, R$ required in SfM?

We will revisit these problems soon!

*https://tutorial.math.lamar.edu/classes/calciii/quadricsurfaces.aspx

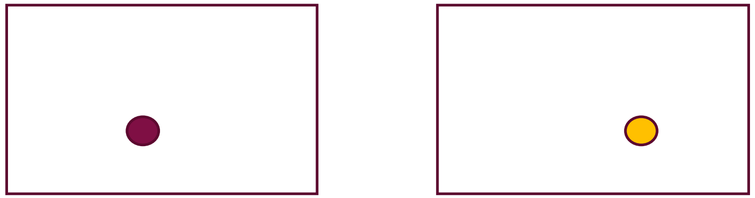
Note: Do we really need 8 Points to solve SfM?

- Actually, there are methods for SfM with as few as 5 point correspondences, corresponding to 5 degrees of freedom in R, T (minus scale ambiguity)

- But much like our P3P Step 1, these involve solving complicated systems of higher degree equations. Instead, the 8-point algorithm offers a “direct” solution by solving a linear system of equations, and is widely used due to its convenience. (Much like the “direct solution” we saw for PnP, which needed $6 > 3$ 2D-3D point correspondences for PnP)

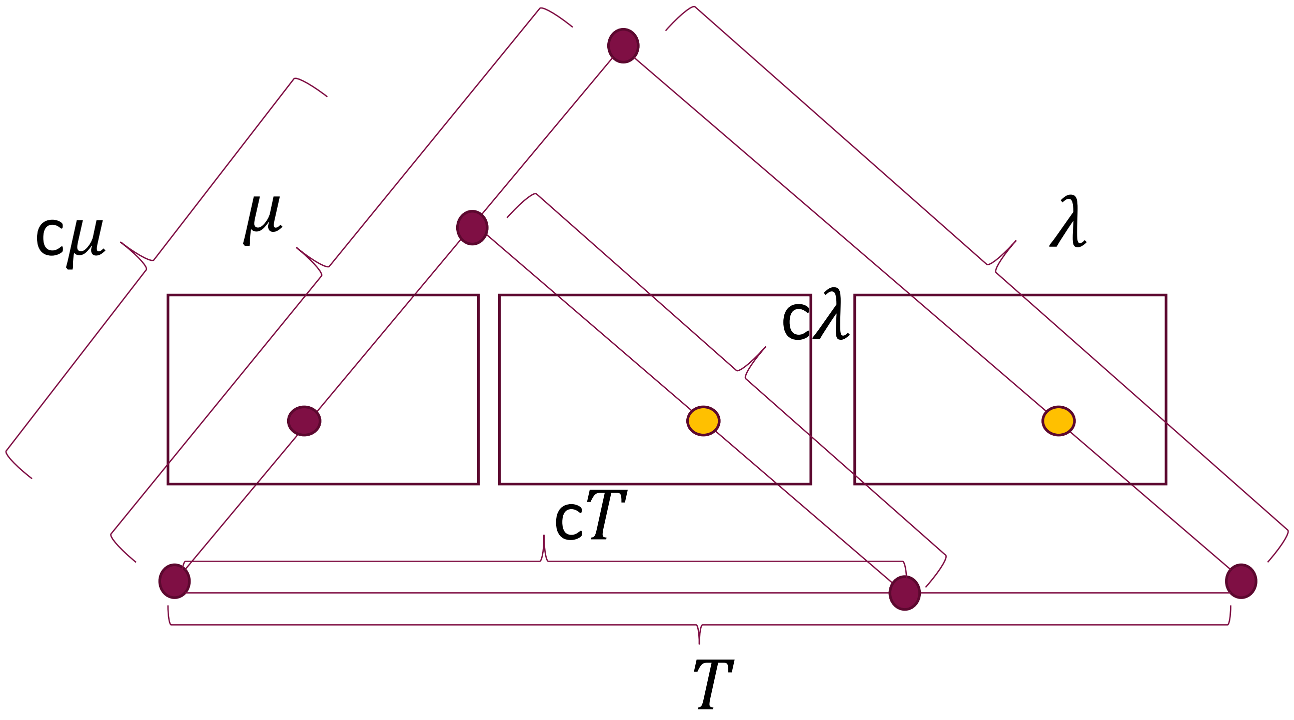
Note: SfM Scale Ambiguity Illustration

Remember, all you are given is calibrated coordinate correspondences between 2 image planes.



So, suppose one solution to SfM is R, T, λ, μ.

Then, an equally valid solution is $R, cT, c\lambda, c\mu$!

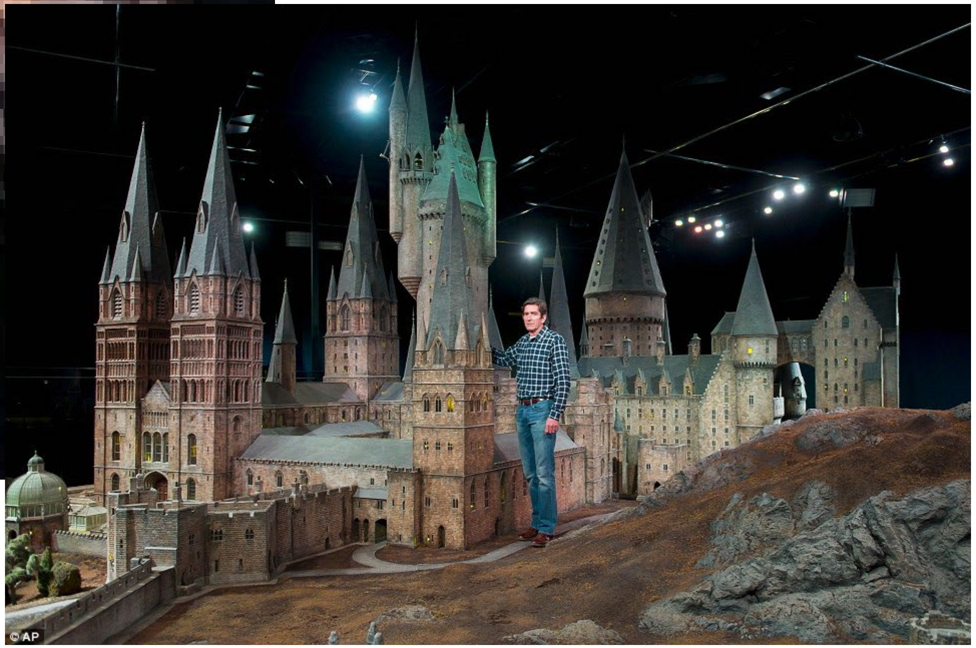


Scale ambiguity applies to purely visual SfM. If you have access to an accelerometer on a moving camera, can resolve scale ambiguity! ("Visual inertial odometry").

SfM Scale Ambiguity and Miniature Movie Sets!

Grand "establishing shots" in movies are often filmed with miniature sets (and green screens for background)

No one can tell, because of scale ambiguity!



https://youtu.be/v59kwB37xQo?t=464

Fundamental matrix

- Essential matrix E connects image plane coordinates i.e. “calibrated coordinates” for points $p = K_p^{-1}\left(u_p, v_p, 1\right)^T$ and $q = K_q{}^{-1}\left(u_q, v_q, 1\right)^T$.

- Fundamental matrix F assumes that we are still in “pixel land”

- So:

$$q^T E p = 0 \qquad \text{can be written as:} \qquad \left(K_q q\right)^T F \left(K_p p\right) = 0$$