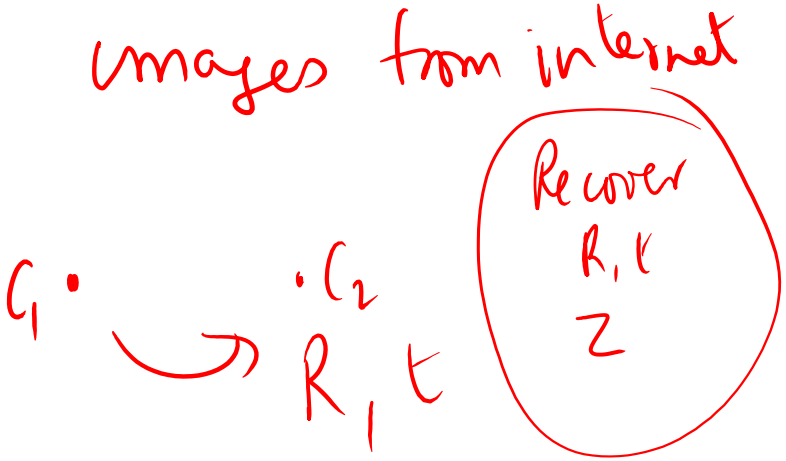
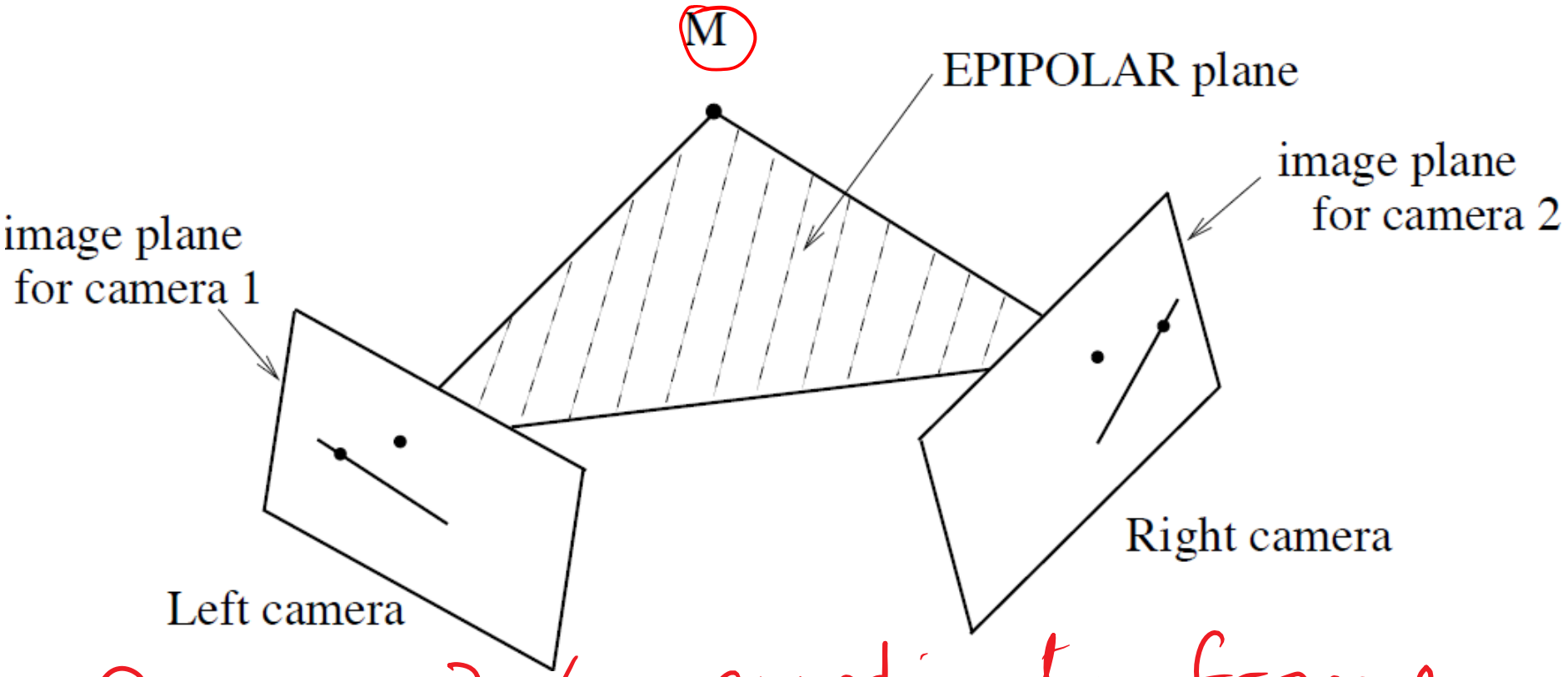
Various camera configurations

- Single point of fixation where optical axes intersect
- Optical axes parallel (fixation at infinity)
- General case

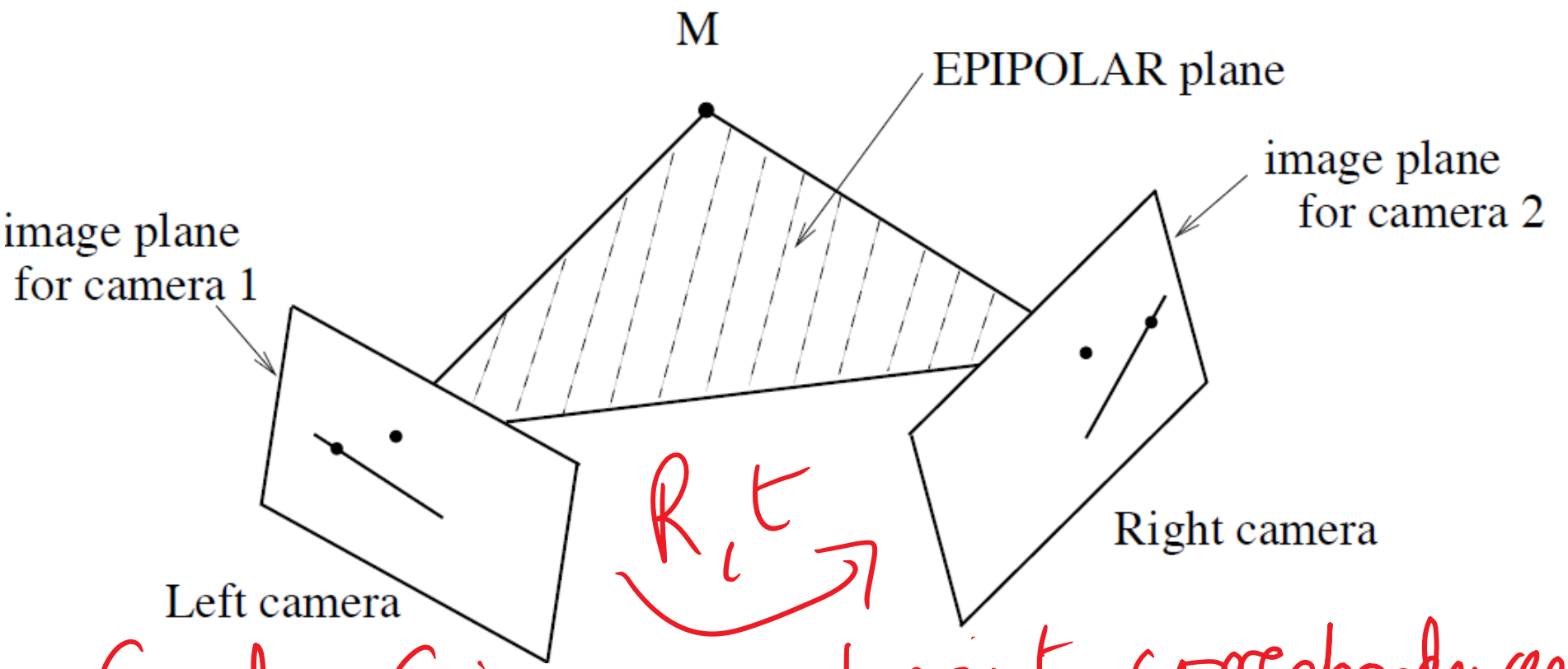


Epipolar geometry for cameras in general position



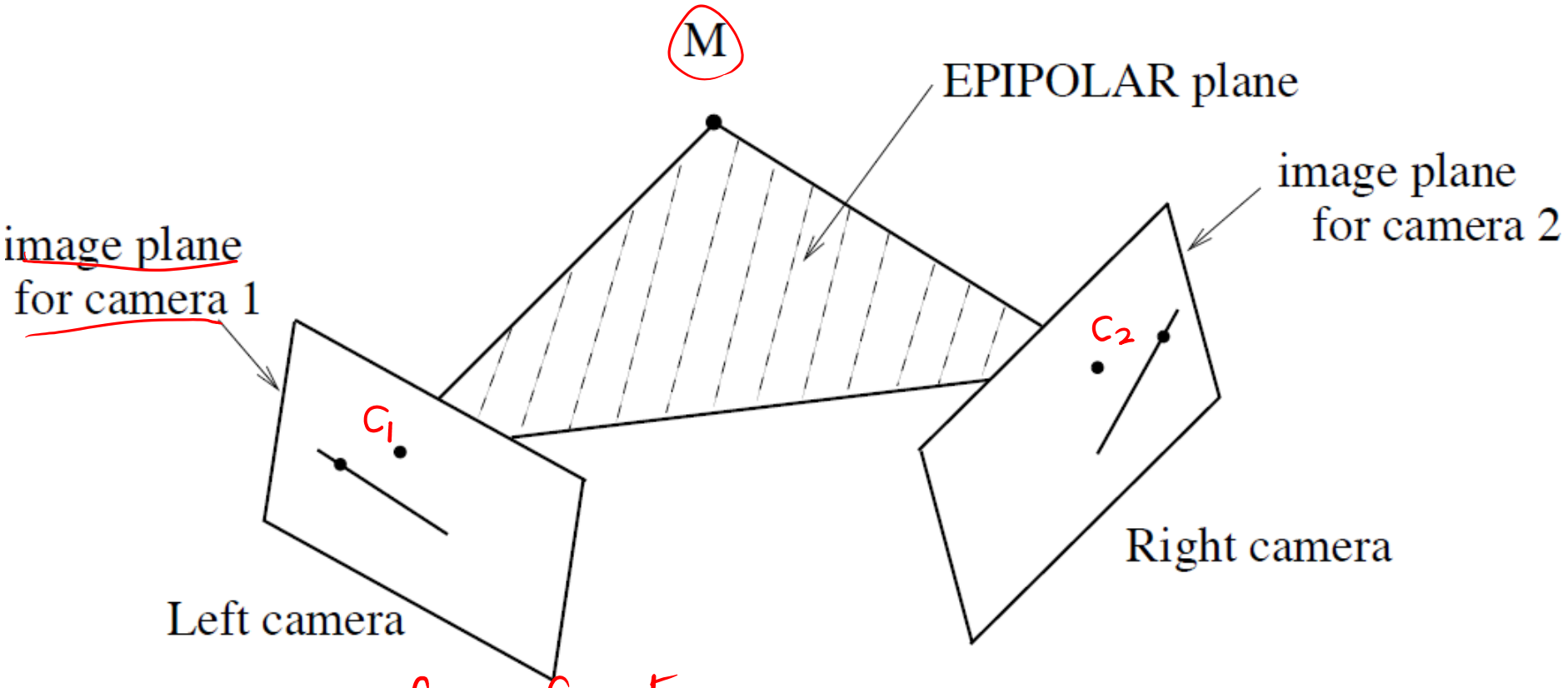
Camera 2's coordinate frame is related to that of Camera 1 by rotation R and translation t

Epipolar geometry for cameras in general position



Goal: Given n point correspondences, estimate R, t and depths at the n points

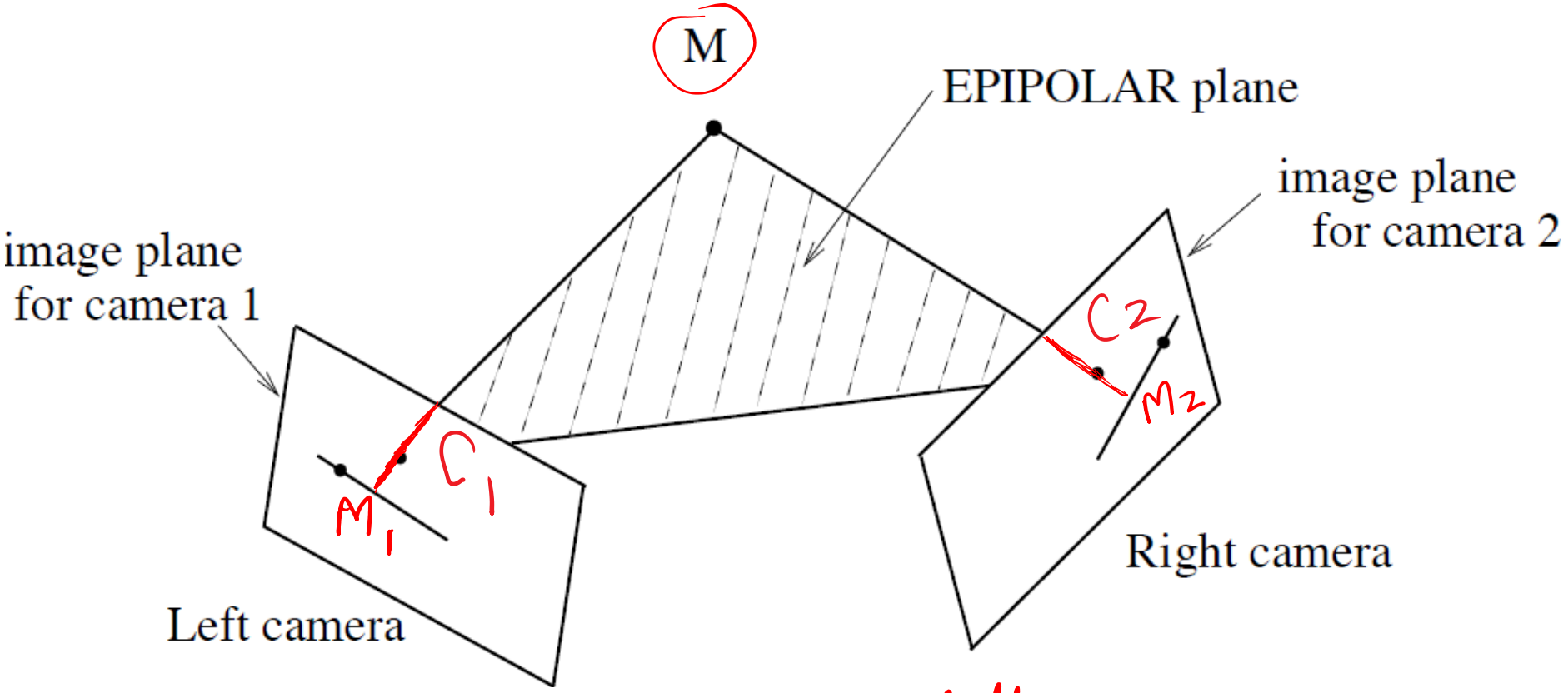
Epipolar geometry for cameras in general position



C_1 is f in front of the image plane of the left camera

Similarly for C_2

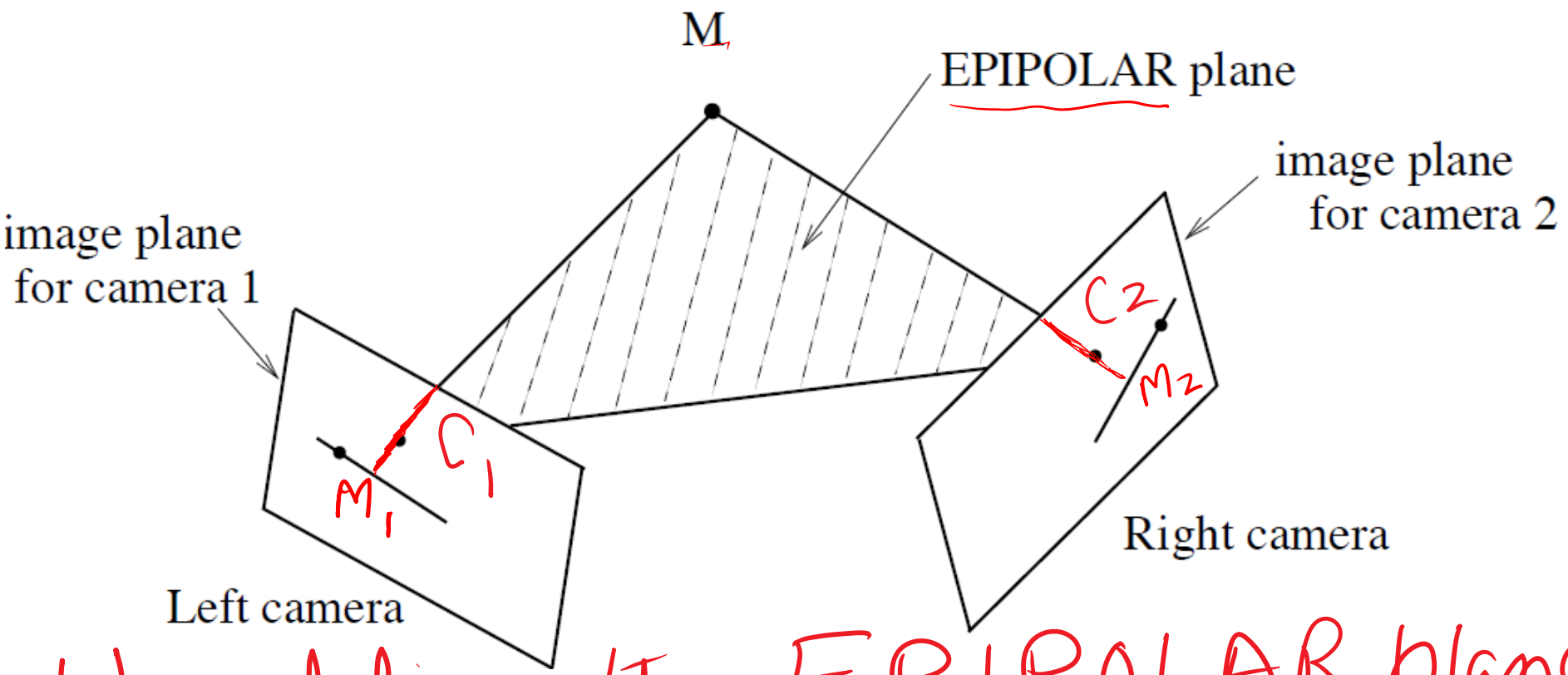
Epipolar geometry for cameras in general position



M_1 is projection in left camera

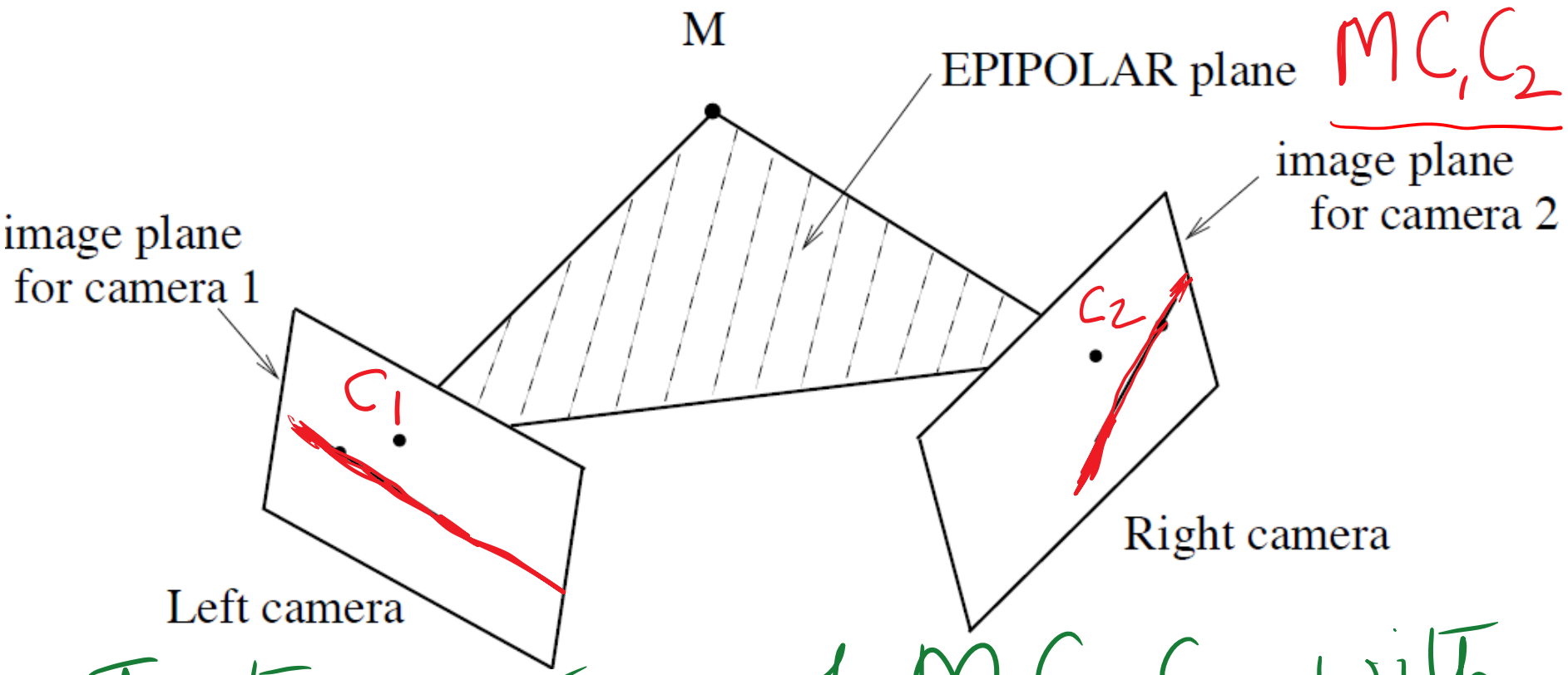
M_2 is projection in right camera

Epipolar geometry for cameras in general position



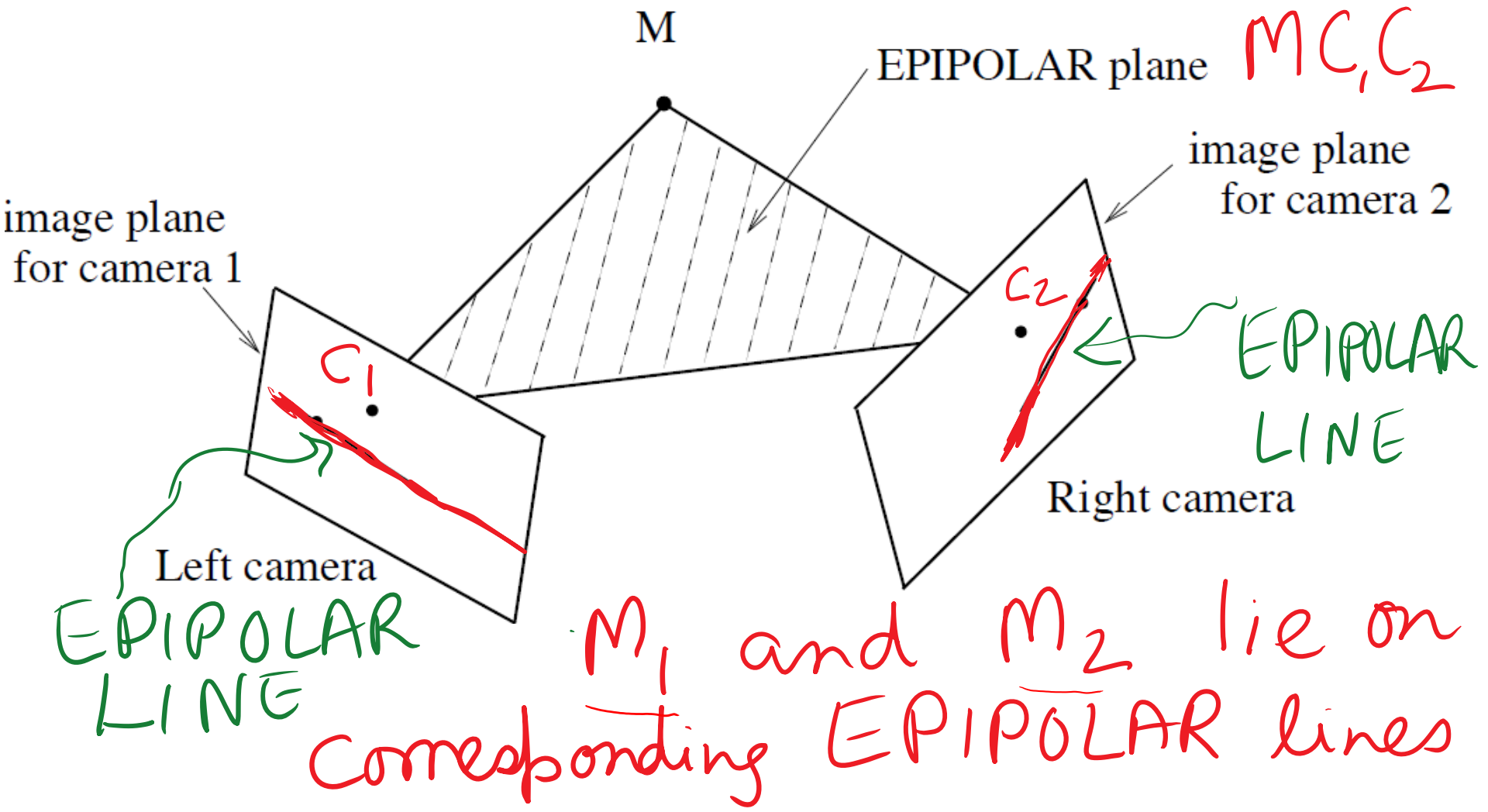
We define the EPIPOLAR plane through M to be MC_1C_2 (or MM_1M_2)

Epipolar geometry for cameras in general position

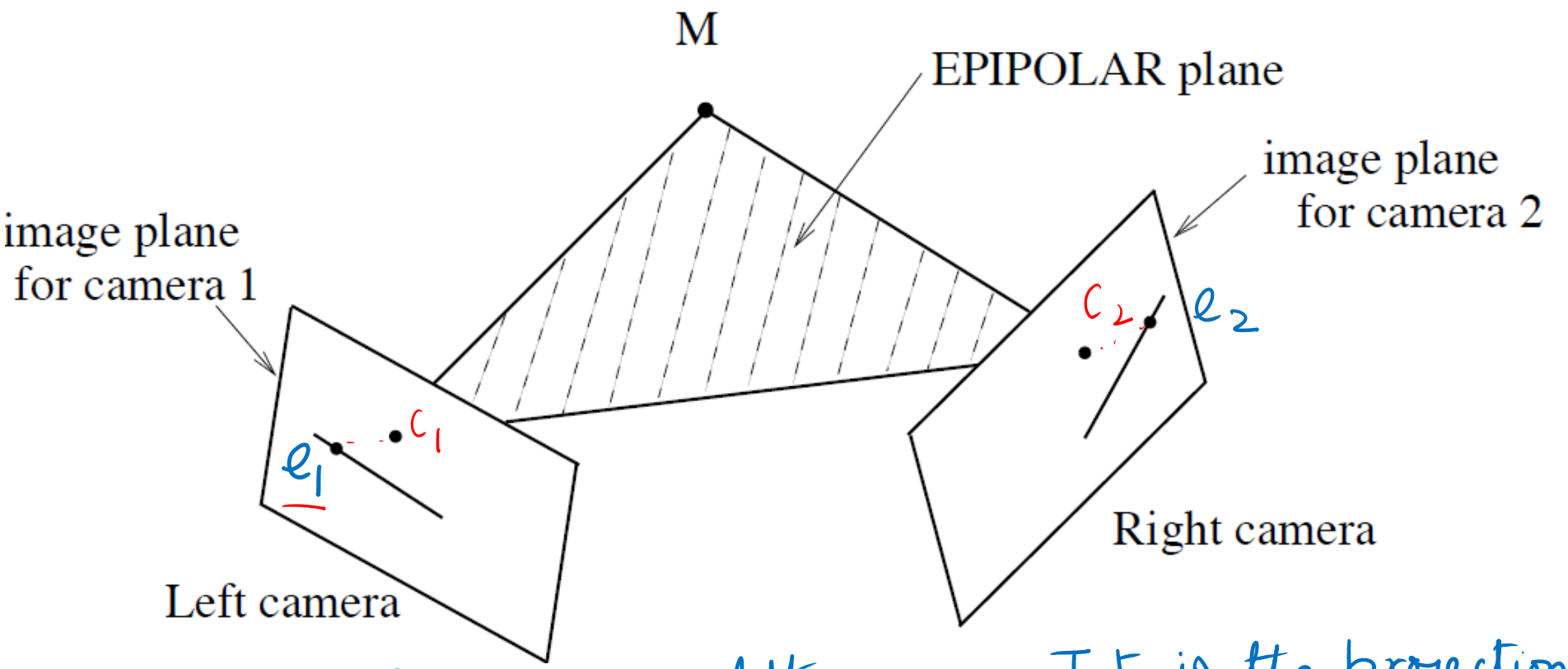


Intersection of MC_1C_2 with image plane = EPIPOLAR line

Epipolar geometry for cameras in general position



Epipolar geometry for cameras in general position



e_1 : EPIPOLE in left camera. It is the projection of C_2 on image plane of left camera

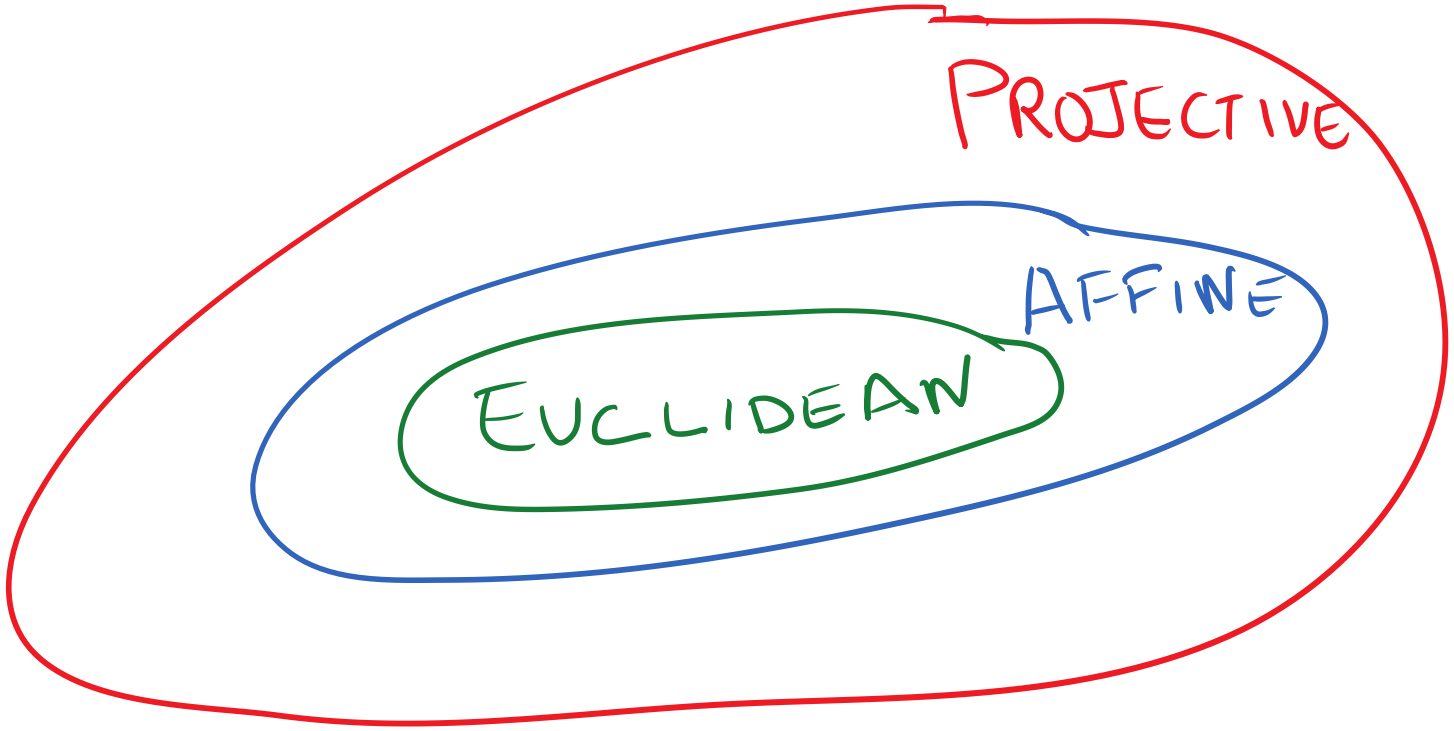
Applications

- Reconstructing a three dimensional object given two or more photos. The relative orientation of the cameras is unknown, such as when using images downloaded from the internet, taken by different people at different times.
- Some examples from Univ. Washington/Microsoft Research:
  - phototour.cs.washington.edu
  - photosynth.net
  - grail.cs.washington.edu/rome/

The approach

- The basic module is recovering 3D structure from 2 views with relative orientation (R, t) of cameras unknown. This has several steps:
  - Finding n corresponding points in the 2 views, i.e. image points which are the projections of the same point in the scene.
  - Estimate the E matrix (= $\hat{T}R$) from these point correspondences.
  - Extract (R, t).
  - Recover depth by triangulation.
- The outer loop combines information from all the cameras in a global coordinate system. Note that not all points will be seen by all cameras. This process is a nonlinear least squares optimization, called bundle adjustment. The error that is minimized is the re-projection error.
- For example, the 3D reconstruction of the Colosseum in Rome was based on 2 K images, and 800 K points.

But first, let us review projective transformations



Projective Transformations

- Under perspective projection, parallel lines can map to lines that intersect. Therefore, this cannot be modeled by an affine transform!
- Projective transformations are a more general family which includes affine transforms and perspective projections.
- Projective transformations are linear transformations using homogeneous coordinates

Homogeneous coordinates

- Instead of using n coordinates for n-dimensional space, we use n+1 coordinates.

$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ for P^1, the projective line $\mathbb{R}^1 \cup \{\text{point at } \infty\}$

$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ for P^2, the projective plane $\mathbb{R}^2 \cup \{\text{line at } \infty\}$

$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$ for P^3 and so on...

Key rule

- $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, $\begin{bmatrix} \lambda x_1 \\ \lambda x_2 \\ \vdots \\ \lambda x_n \end{bmatrix}$ $\lambda \neq 0$

represent the same point in P^{n-1}

(thus only $n-1$ degrees of freedom)

Think of this as a line through the origin in n-dimensional space

Note: $x_1, x_2 \ldots x_n$ may not all be 0

Picking a canonical representative

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \longmapsto \begin{bmatrix} x_1/x_3 \\ x_2/x_3 \\ 1 \end{bmatrix}$$

(only if $x_3 \neq 0$)

$$\therefore (x, y) = \left(\frac{x_1}{x_3}, \frac{x_2}{x_3}\right)$$

Conversely

$$\begin{bmatrix} x \\ y \end{bmatrix} \longmapsto \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

augmented vector

The projective line

- Any finite point x can be represented as

$$\begin{bmatrix} x \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} 2x \\ 2 \end{bmatrix} \text{ or } \begin{bmatrix} 6.3x \\ 6.3 \end{bmatrix} \text{ or } \ldots$$

- Any infinite point can be expressed as

$$\begin{bmatrix} x \\ 0 \end{bmatrix}$$

(Note: there is only one such point)

The projective plane

- Any finite point can be represented as

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix}$$

- Any infinite point can be represented as

$$\begin{bmatrix} x \\ y \\ 0 \end{bmatrix}$$

Thus there is a line at infinity
Different ratios $x : y$ give different points.

Lines in homogeneous coordinates

Consider $a_1 x + a_2 y + a_3 = 0$

Note $\lambda a_1 x + \lambda a_2 y + \lambda a_3 = 0$ is the same line.

$$\begin{bmatrix} x \\ y \end{bmatrix} \longleftrightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \text{with } x = \frac{x_1}{x_3}, \; y = \frac{x_2}{x_3}$$

$$a_1 \frac{x_1}{x_3} + a_2 \frac{x_2}{x_3} + a_3 = 0$$

$$a_1 x_1 + a_2 x_2 + a_3 x_3 = 0$$

Incidence of points on lines

Q. When does a point $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ lie on a line $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$?

A. $$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

Q. Where do the lines $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$, $\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$ intersect?

Incidence of points on lines

A. $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \wedge \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$

Example:

- $x = 1$ and $y = 1$

$$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (1, 1)$$

Incidence of points on lines

A. $\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \wedge \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$

Example:

- $x = 1$ and $x = 2$

$$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \wedge \begin{bmatrix} -1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Line incident on two points

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} \quad \text{is given by}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \wedge \begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} \quad !$$

An example of DUALITY

Representing affine transformations

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = a_{11}x + a_{12}y + t_x$$

$$y' = a_{21}x + a_{22}y + t_y$$

$$w' = 1$$

which does the right thing!!

Euclidean transforms are affine, so the same trick works when $A = R$

Perspective Projection

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z/f \end{bmatrix}$$

$$= \begin{bmatrix} fX/Z \\ fY/Z \\ 1 \end{bmatrix}$$

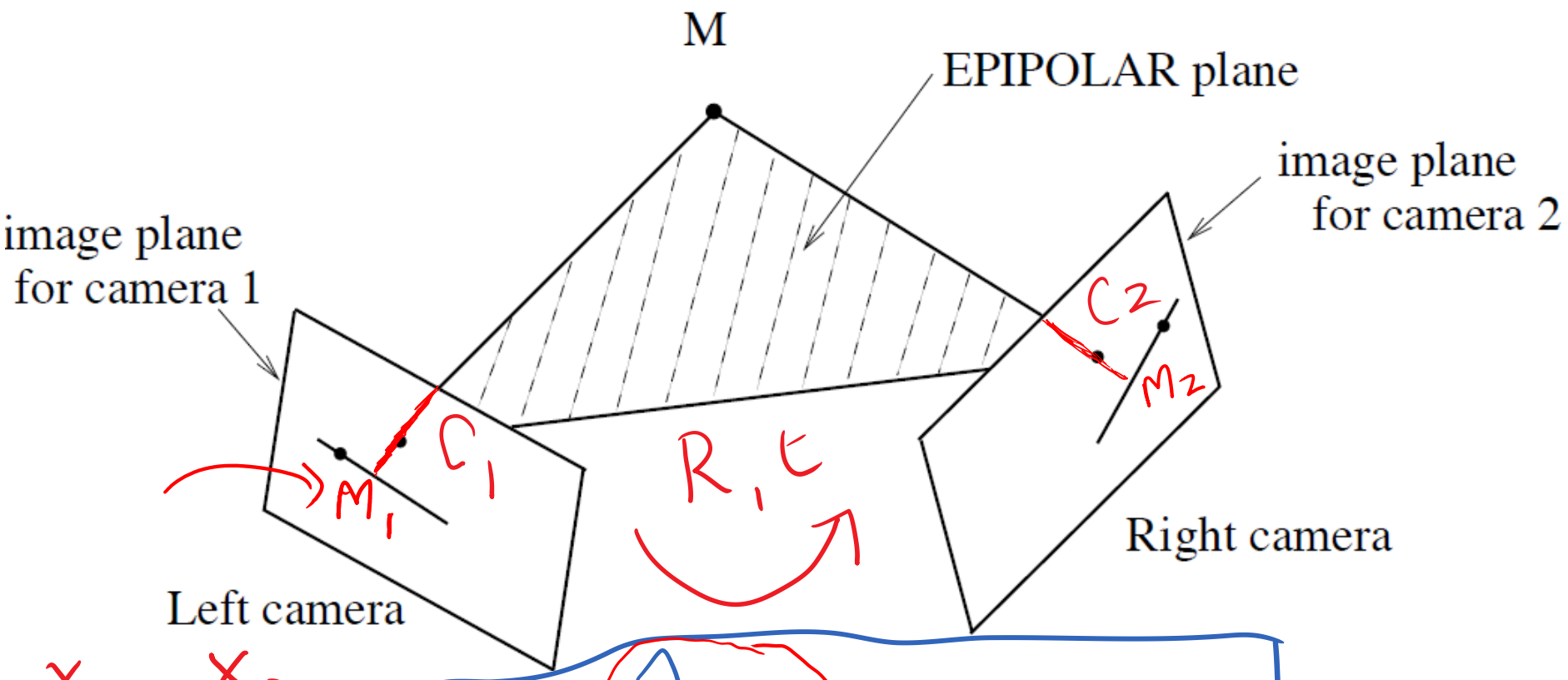
Projective transformations

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \text{in } P^2$$

Notes: 8 independent parameters
matrix required to be non-singular

For projective transforms in
P^1 3 independent parameters
P^3 15 independent parameters

The Essential Matrix Constraint



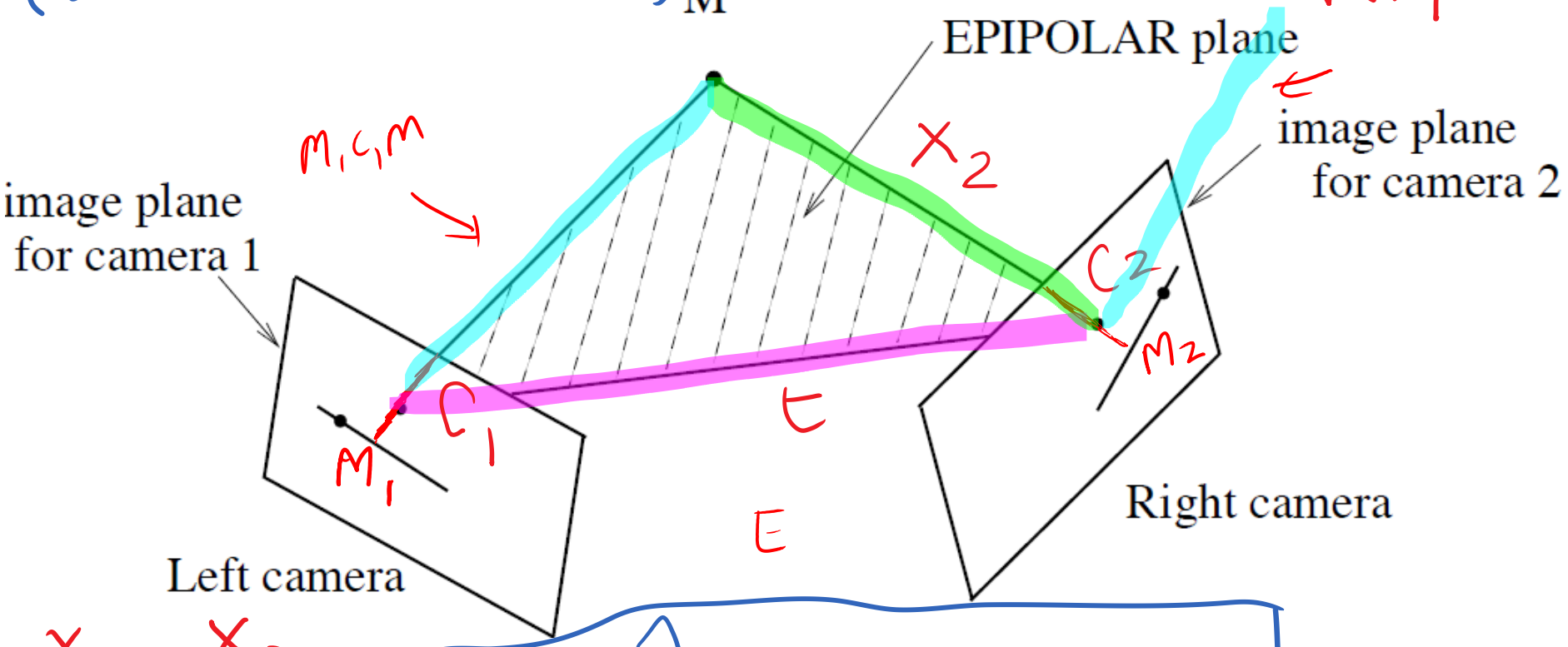
x_1, x_2 homogeneous coordinates of M_1, M_2

$$x_2^T \hat{T} R x_1 = 0$$

Proof is based on the co-planarity of three rays

X_2, t, RX_1 $\Rightarrow$ triple product is 0

(in second camera frame)



X_1, X_2 homogeneous coordinates of M_1, M_2

$$\hat{X}_2^T T R X_1 = 0$$

Coplanarity of 3-vectors implies triple product is zero

v_1, v_2, v_3 are coplanar $\Rightarrow$

$v_1 \cdot (v_2 \wedge v_3) = 0$

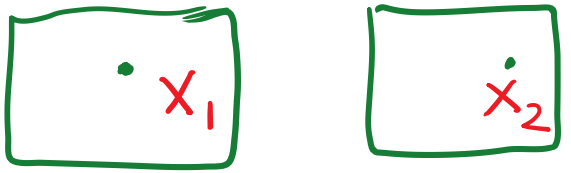
$v_1^T \hat{v}_2 v_3 = 0$

Longuet-Higgins 8 point algorithm

(1981)

- Find n (≥ 8) corresponding points in the 2 views
- Estimate the E matrix (= $\hat{T}R$) from these point correspondences.
- Extract (R, t).
- Recover depth by triangulation.

$E \rightarrow \lambda E$



measured in each camera's coordinates

Given projections x_1, x_2

$$x_2^T E x_1 = 0$$

$$\begin{bmatrix} - & - & - \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} - \\ - \\ - \end{bmatrix} = 0$$

Each point gives a linear equation for entries of E

Summary

- The basic module of recovering 3D structure from 2 views with relative orientation (R, t) of cameras unknown can be implemented using the Longuet-Higgins 8 point algorithm.
- The outer loop combines information from all the cameras in a global coordinate system using bundle adjustment. The error that is minimized is the re-projection error. The big idea is that given the guessed 3d positions of a point, one can predict image plane 2d positions in any camera where it is visible. We wish to minimize the squared error between this predicted position and the actual position, summed over all cameras and over all points.
- Lots of engineering has gone into making these approaches work. Read Szeliski's book, Chapter 7, for more.