


FACULTY OF MATHEMATICS,
PHYSICS AND INFORMATICS
Comenius University
Bratislava

3D Vision

# Lecture 3: Vanishing Points and Lines, Homography Estimation

Ing. Viktor Kocur, PhD.

7.3.2023

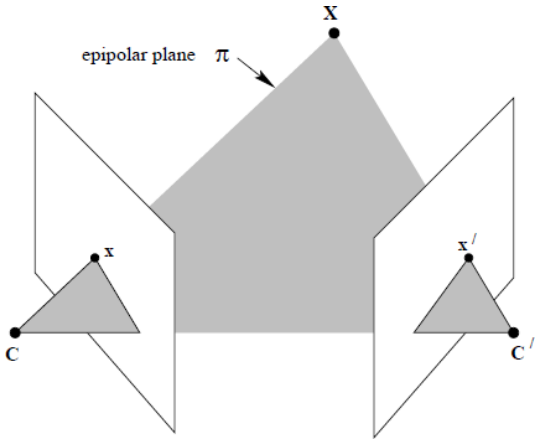
# Contents

# How are images in two views related?

Suppose we have two cameras with $P = K[I|\mathbf{0}]$ and $P' = K'[R, \boldsymbol{t}]$. (Note that we will keep using this notation on future slides.) We want to know how a single point $\boldsymbol{X}$ from the real scene is imaged in both images as $\boldsymbol{x}$ and $\boldsymbol{x'}$ respectively.

What do we know about the relationship of $\boldsymbol{x}, \boldsymbol{x'}$ and $\boldsymbol{X}$?

# Epipolar Geometry




We can see that $\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{X}$ and both of the camera centers are coplanar. We denote this plane as $\pi$.

Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

# Epipolar Lines

Now suppose that we only know $\boldsymbol{x}$. What can we tell about the position of $\boldsymbol{x'}$?

Since the point $\boldsymbol{x'}$ has to lie on $\pi$ and also the center of the second camera lies on $\pi$ too, the point $\boldsymbol{x'}$ has to lie on line $\boldsymbol{l'}$ which is the image of $\pi$.

# Epipolar Geometry

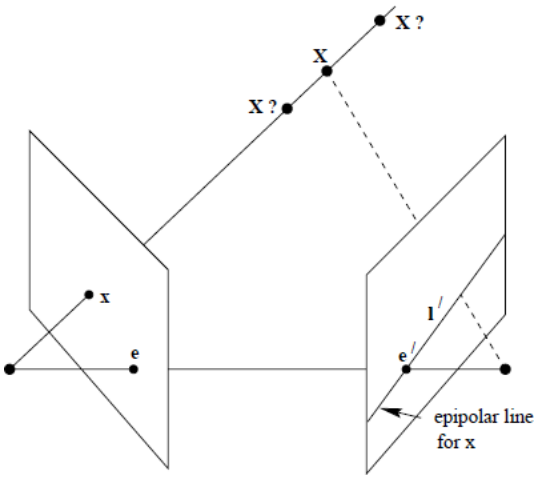



Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

# Epipolar Plane

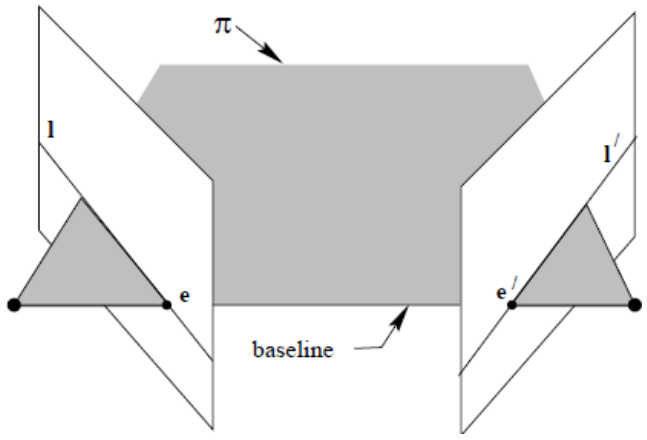



Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

# Pencil of Epipolar Lines

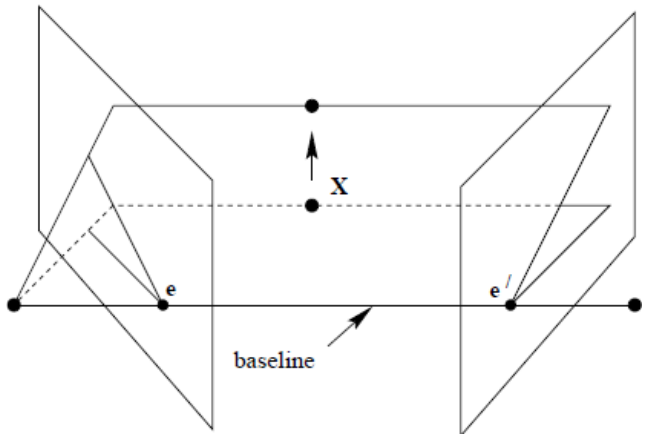


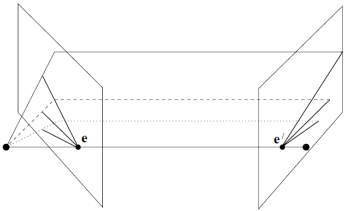

Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

# Epipolar Geometry - Objects

Let us now define some of the geometrical objects we used:

- The **epipole** is the point of instersection of line joining the camera centers and the image plane. Equivalently, the epipole is the image in one view of the camera centre of the other view. It is also the vanishing point of the baseline (translation) direction.
- An **epipolar plane** is a plane containing the baseline. There is a one-parameter family (a pencil) of epipolar planes.
- An **epipolar line** is the intersection of an epipolar plane with the image plane. All epipolar lines intersect at the epipole. An epipolar plane intersects the left and right image planes in epipolar lines, and defines the correspondence between the lines.

# Pencil of Epipolar Lines




a

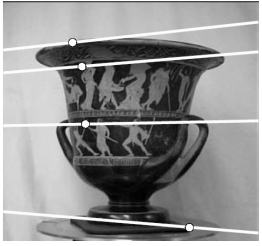


Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

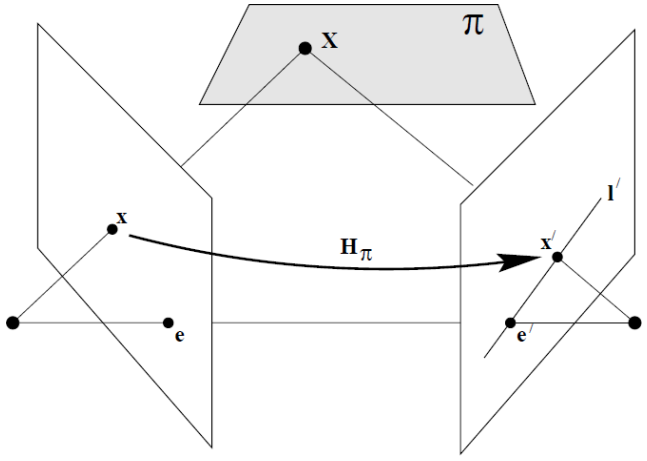
# The Fundamental Matrix

We thus know that there is a map $\boldsymbol{x} \mapsto \boldsymbol{l'}$. It turns out that this mapping will have the form of:

$$\boldsymbol{l'} \sim F\boldsymbol{x'}. \tag{1}$$

The matrix $F$ is the **fundamental matrix**. It encompases all information about the relative geometry of the two cameras. The rank of $F$ is 2 and it is thus singular.

# Geometric Derivation




Consider a plane $\pi$ which contains $\boldsymbol{X}$ and does not pass through any of the two camera centers.

Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

# Geometric Derivation

In order to obtain the line $\boldsymbol{l'}$ we want to form the line from $\boldsymbol{e'}$ to $\boldsymbol{x'}$. We can use transfer via plane by defining a homography $H_\pi$ between the two views. Thus we can get $\boldsymbol{x'} = H_\pi \boldsymbol{x}$. Now, recall that we can construct lines using the vector product of two points and we can also rewrite vector product as matrix multiplication using skew-symmetric matrices. We can thus obtain the final expression:

$$\boldsymbol{l'} = [\boldsymbol{e'}]_\times H_\pi \boldsymbol{x} = F\boldsymbol{x}. \tag{2}$$

This also shows that $F$ is of rank 2, because $[\boldsymbol{e'}]_\times$ is rank 2 and $H_\pi$ is regular. Note that we have freedom in choosing the plane $\pi$ and also thus $H_\pi$. The plane does not have to be present in the scene in reality.

# The Fundamental Matrix

Suppose we have two cameras with $P = K[I|\mathbf{0}]$ and $P' = K'[R, \boldsymbol{t}]$. Then it is possible to derive the following expressions:

$$F \sim [\boldsymbol{e}]_\times K' R K^{-1} \sim K'^{-T}[\boldsymbol{t}]_\times R K^{-}1 \sim K'^{-T} R [\boldsymbol{R^T t}]_\times K^{-}1 \sim K'^{-T} R K^T [\boldsymbol{e}]_\times. \qquad (3)$$

Note that $F$ contains information about rotation and translation and both intrinsic camera matrices.

# Correspondence Condition

Consider again two cameras $P$ and $P'$. We can now consider a relationship between two images $\boldsymbol{x}$ and $\boldsymbol{x'}$. Since $\boldsymbol{x'}$ has to lie on $\boldsymbol{l'} \sim F\boldsymbol{x}$ we can write the correspondence condition:

$$\boldsymbol{x'}^T F \boldsymbol{x} = 0. \tag{4}$$

Note that this enables us to characterize the $F$ only using points without the need to knowing $K, K', R$ or $\boldsymbol{t}$. This is similar to how we defined homographies and then showed that they can be described by 4 correspondences.

# Properties of the Fundamental Matrix

- Transpose - if $F$ is the fundamental matrix for pair $(P, P')$ then $F^T$ is the fundamental matrix for $(P', P)$.
- Epipolar lines - for any point $\boldsymbol{x}$ in the first image $F\boldsymbol{x}$ represent the epipolar line $\boldsymbol{l'}$ in the second image. Equivalently $F^T\boldsymbol{x'}$ represents $\boldsymbol{l}$.
- The epipoles - It holds that $F\boldsymbol{e} = \boldsymbol{0}$ and $\boldsymbol{e'}^T F = \boldsymbol{0}$.
- F has seven degrees of freedom. It has 9 entries. One degree of freedom is reduced by scaling and another one is reduced due to rank$(F)$.

# Pure Translation

We may consider a special case of motion of single camera ($K' = K$) which is pure translation (e.g. $R = I$). This type of motion leads to simplification of $F$ to:

$$F = [\boldsymbol{e}']_\times K K^{-1} = [\boldsymbol{e}']_\times. \tag{5}$$

This leaves $F$ with only two degrees of freedom as we only need to find $\boldsymbol{e}'$.

This also gives us insight into $F$ for general motion. We can use a homography $H$ to rotate the second image so that $R = I$ and then we have $F = [\boldsymbol{e}']_\times H$.

# Pure Translation

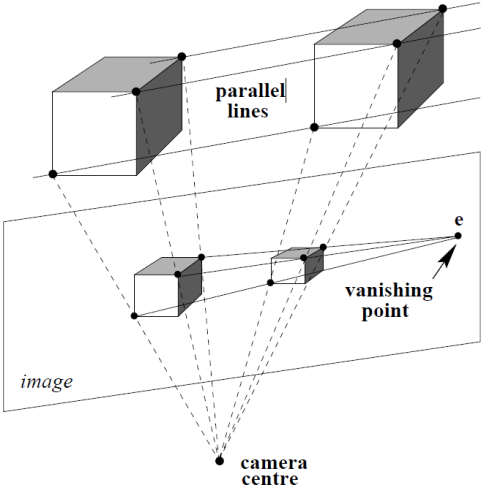



Image adopted from: Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003

# The Essential Matrix

Let us now consider a case when the two cameras are calibrated. We can then define $\hat{\boldsymbol{x}} = K^{-1}\boldsymbol{x}$ and $\hat{\boldsymbol{x}}' = K'^{-1}\boldsymbol{x}'$. This is equivalent to cameras $\hat{P} = [I|\mathbf{0}]$ and $\hat{P}' = [R|\boldsymbol{t}]$, where $\hat{\boldsymbol{x}} \sim \hat{P}\boldsymbol{X}$ and $\hat{\boldsymbol{x}}' \sim \hat{P}'\boldsymbol{X}$. From (3) we can see that for a pair of correspondences we have:

$$0 = \boldsymbol{x}'^T F \boldsymbol{x} = \hat{\boldsymbol{x}}'^T K'^T F K \hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}'^T K'^T K'^{-T} [\boldsymbol{t}]_\times R K^{-1} K \hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}'^T [\boldsymbol{t}]_\times R \hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}'^T E \hat{\boldsymbol{x}} = 0, \quad (6)$$

where matrix $E = [\boldsymbol{t}]_\times R$ is called the **essential matrix**. It relates two views with calibrated cameras. We can see from the derivation above that:

$$E = K'^T F K. \quad (7)$$

# Properties of the Essential Matrix

$E$ has to be of rank two. However a further condition applies. The singular values of $E$ have to be $(\sigma, \sigma, 0)$. In other words, the two non-zero singular values have to be equal, thus SVD of $E$ is:

$$E = U\text{diag}(\sigma, \sigma, 0)V^T. \tag{8}$$

This leaves $E$ with 5 degrees of freedom as opposed to the 7 degrees for $F$.

# Extracting $R$ and $\boldsymbol{t}$ from $E$

We can factorize $E$ to the form $E = [\boldsymbol{t}]_\times R$ where $R$ is a rotation matrix and $\boldsymbol{t}$ is the translation vector. First let us define two matrices $Z$ and $W$:

$$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, Z = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{9}$$

then considering SVD of $E = U\text{diag}(1, 1, 0)V^T$ we can obtain:

$$[\boldsymbol{t}]_\times = UZU^T, \tag{10}$$

$$R = UWV^T \text{ or } UW^TV^T. \tag{11}$$

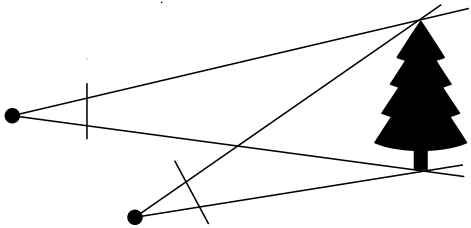
This yields 4 configurations - 2x from choice of $R$ and 2x from changing signs of $\boldsymbol{t}$. Only one of these will be valid such that the points are in front both cameras.

# Scale Ambiguity

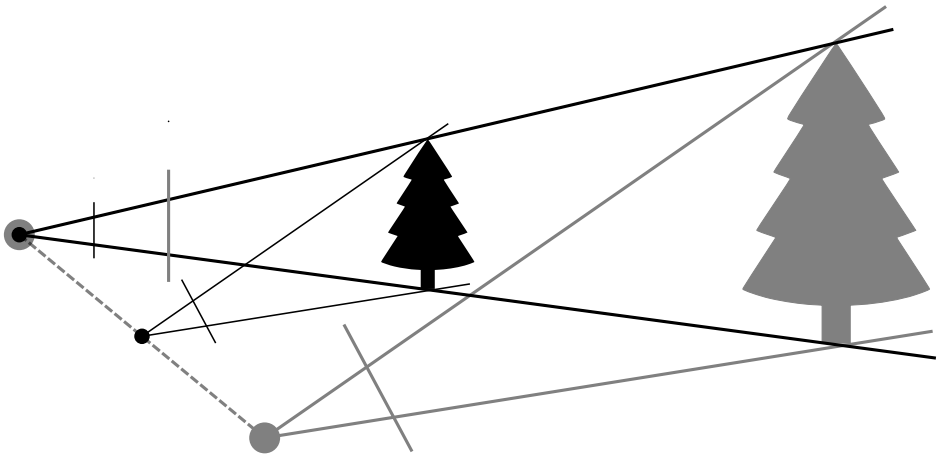
The vector $\boldsymbol{t}$ we obtained using (10) is always such that $|\boldsymbol{t}| = 1$. This is due to scale ambiguity.

Suppose that we have a scene with two cameras 10 cm apart imaging an object with size of 5 cm. And the same scene but the cameras are 10 m apart an object with size of 5 m. The images created by the two cameras will be the same. It is therefore impossible to determine the scale of the scene from point correspondences even if we know the camera intrinsics!

# Scale Ambiguity



# Scale Ambiguity



# Why do we care about $F$?

You may ask why do we even try to estimate $F$ instead of $E$ in the first place? This is because we can extract additional information about the camera intrinsics from $F$ that are not present in $E$. For example if you consider that pinhole cameras with zero skew and known principal points then it is possible to obtain the focal lengths of both cameras.

Though in many applications we only calculate $E$ even if the camera intrinsics are not known exactly and we further refine them using non-linear optimization.

# Estimation of $F$

We will now discuss algorithms for estimating $F$. From point correspondences $\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}_i'$ we obtain equations:

$$\boldsymbol{x}_i'^T F \boldsymbol{x}_i = 0. \tag{12}$$

The fundamental matrix also has to satisfy the following:

$$\det(F) = 0, \tag{13}$$

which is a third order polynomial in the entries of $F$. We will now derive two different approaches: the 7-point minimal algorithm and the 8-point algorithm.

We can rewrite (12) to the form:

$$\boldsymbol{a}_i^T \boldsymbol{f} = \begin{pmatrix} x_i' x_i, & x' y, & x', & y' x, & y' y, & y', & x, & y, & 1 \end{pmatrix} \boldsymbol{f} = 0, \tag{14}$$

where $\boldsymbol{x} = (x, y, 1)^T, \boldsymbol{x'} = (x', y', 1)^T$ and $\boldsymbol{f}$ is the vector representation of $F$ such that:

$$F = \begin{pmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{pmatrix}. \tag{15}$$

# Linear System

We can use $n$ equations (14) for $n$ correspondences and rewrite them into:

$$A\boldsymbol{f} = \begin{pmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_n^T \end{pmatrix} \boldsymbol{f} = \boldsymbol{0}. \tag{16}$$

The matrix $A$ has size $n \times 9$ and would have at most rank 8 if we used 8 or more correspondences and they were exact. In that case we could find $F$ via the null-vector of $A$. However this is not possible in practice.

# 8-point algorithm

We will use $A$ generated using 8 or more correspondences. Then we will find the singular vector $\hat{\boldsymbol{f}}$ corresponding to the smallest eigenvalue using SVD of $A$. Recall that this minimizes $|A\hat{\boldsymbol{f}}|$ while $|\hat{\boldsymbol{f}}| = 1$.

This does not guarantee that the resulting matrix $\hat{F}$ will be singular. We therefore want to find singular $F$ such that $||F - \hat{F}||$ is minimized.

# 8-point algorithm

We will use $A$ generated using 8 or more correspondences. Then we will find the singular vector $\hat{\boldsymbol{f}}$ corresponding to the smallest eigenvalue using SVD of $A$. Recall that this minimizes $|A\hat{\boldsymbol{f}}|$ while $|\hat{\boldsymbol{f}}| = 1$.

This does not guarantee that the resulting matrix $\hat{F}$ will be singular. We therefore want to find singular $F$ such that $||F - \hat{F}||$ is minimized. We can do this using SVD of $\hat{F} = U\text{diag}(\sigma_1, \sigma_2, \sigma_3)V^T$ (we assume $\sigma_1 > \sigma_2 > \sigma_3$). Then the minimizing $F$ is obtained as

$$F = U\text{diag}(\sigma_1, \sigma_2, 0)V^T. \tag{17}$$

# Normalized 8-point Algorithm

The stability of the resulting algorithm depends on the correspondences. It is therefore advisable to perform normalization of the coordinates to improve stability. Instead of using the original correspondences $\boldsymbol{x}_i' \leftrightarrow \boldsymbol{x}_i$ we use $\boldsymbol{q}_i' \leftrightarrow \boldsymbol{q}_i$ where $\boldsymbol{q}' = T'\boldsymbol{x}'$ and $\boldsymbol{q} = T\boldsymbol{x}$, where $T$ and $T'$ are transformations performing rotation and translation of the vectors. We then proceed to obtain the fundamental matrix $Q$. And then we calculate:

$$F = T'^{T} Q T. \tag{18}$$

$T$ and $T'$ are selected that the centroids of the points are at origin and the root mean square distance of points from origin is $\sqrt{2}$.

# 7-point Algorithm

We can also find the minimal solution for $F$ given only 7 point correspondences. We can use the same system $A\boldsymbol{f} = 0$. The matrix $A$ would be rank 7. This means that the null-space is two dimensional. We can use SVD to find two vectors $\boldsymbol{f}_1, \boldsymbol{f}_2$ which form a base of the null-space.

# 7-point Algorithm

We can also find the minimal solution for $F$ given only 7 point correspondences. We can use the same system $A\boldsymbol{f} = 0$. The matrix $A$ would be rank 7. This means that the null-space is two dimensional. We can use SVD to find two vectors $\boldsymbol{f}_1, \boldsymbol{f}_2$ which form a base of the null-space. This means that the final $F$ is a linear combination of $F_1$ and $F_2$. Due to scale ambiguity we can write

$$F = \alpha F + (1 - \alpha)F. \tag{19}$$

To find the final $F$ we solve the rank constraint:

$$\det(F) = \det(\alpha F + (1 - \alpha)F) = 0, \tag{20}$$

which is a cubic equation. This equation may have 1, 3 or no real solutions. In practice we use this algorithm within RANSAC so we can test all of them.

# Degeneracies

During estimation of $F$ we may encounter some degeneracies. Both algorithms may be unstable when input with configurations close to the degenerate ones.

- Critical surfaces - If the points and both camera centers lie on some types of surfaces (some quadrics) then this might result in degeneracy of the configuration. This could result in multiple solutions for $F$.
- Points on a plane - If all points lie on a plane then there are multiple solutions to $F$.
- No translation - If there is no translation then $F$ is not well defined.

In practice we may also have issues when the motion is close to pure translation. In that case it is better to estimate directly for $F = [\boldsymbol{e}']_\times$.

# $F$ Estimation Pipeline

We may use both algorithms to estimate $F$. In practice we proceed in two steps:

1. We use the 7-point algorithm within RANSAC to find the inlier points.
2. We obtain the final $F$ using the (normalized) 8-point algorithm.

We may also use some iterative optimization methods once we have good initial estimate of $F$. However, such methods may be slow compared with the 7-point and 8-point algorithms.

# 5-point Algorithm

We may also want to estimate $E$. Analogous to 8-point and 7-point algorithms we construct a matrix $\hat{A}$ such that we obtain a system:

$$\hat{A}\boldsymbol{e} = \mathbf{0}, \tag{21}$$

where $\hat{A}$ is same as $A$ in previous slides but it uses $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{x}}'$ instead of $\boldsymbol{x}$ and $\boldsymbol{x}'$. We then obtain a four dimensional null-space - $E = \alpha E_1 + \beta E_2 + \gamma E_3 + \delta E_4$. We then solve a system of polynomial equations:

$$\det(E) = 0, \tag{22}$$

$$2EE^T E - \mathrm{tr}(EE^T)E = 0. \tag{23}$$

(23) actually represents 9 equations for the entries of the resulting matrix, but only two of them are algebraically independent. Solving this system is not so straightforward and we will not cover it in this lecture.