

Введение в фотограмметрию

Structure from motion (SFM)

Фотограмметрия. Лекция 6



- Эпиполярная геометрия
- Фундаментальная матрица
- Триангуляция, резекция

Симиютин Борис

simiyutin.boris@yandex.ru

Мотивация

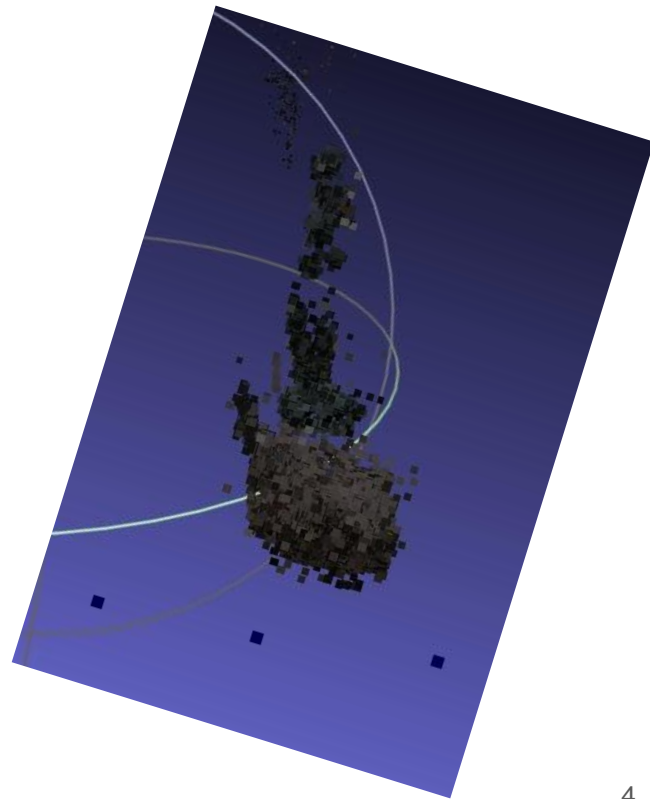
- 1) Хотим по матчам
определять взаимное
расположение камер

Мотивация

- 1) Хотим по матчам
определять взаимное
расположение камер
 - a) В случае двух камер
 - b) В случае трех и более камер

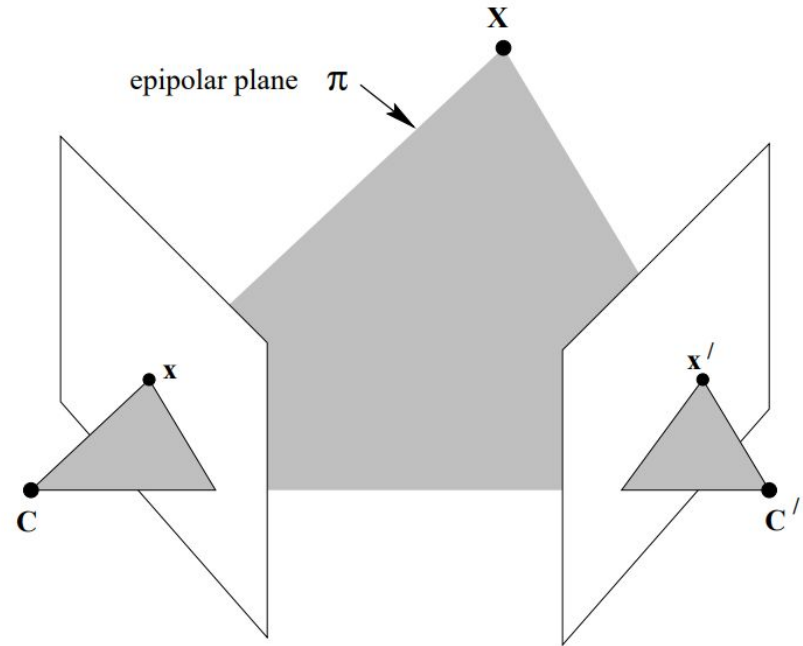
Мотивация

- 1) Хотим по матчам определять взаимное расположение камер
 - а) В случае двух камер
 - б) В случае трех и более камер
- 2) Хотим научиться строить облако точек, соответствующее наблюдаемой геометрии (можно покрутить в 3D вьювере)



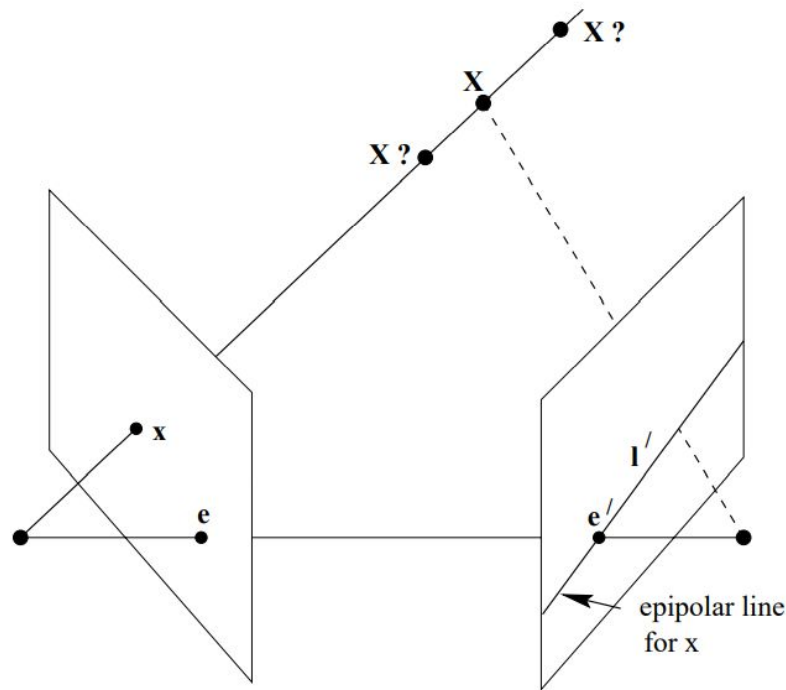
Эпиполярная геометрия

- 1) Каждая пара камер порождает пучок плоскостей, проходящих через ось (C, C')



Эпиполярная геометрия

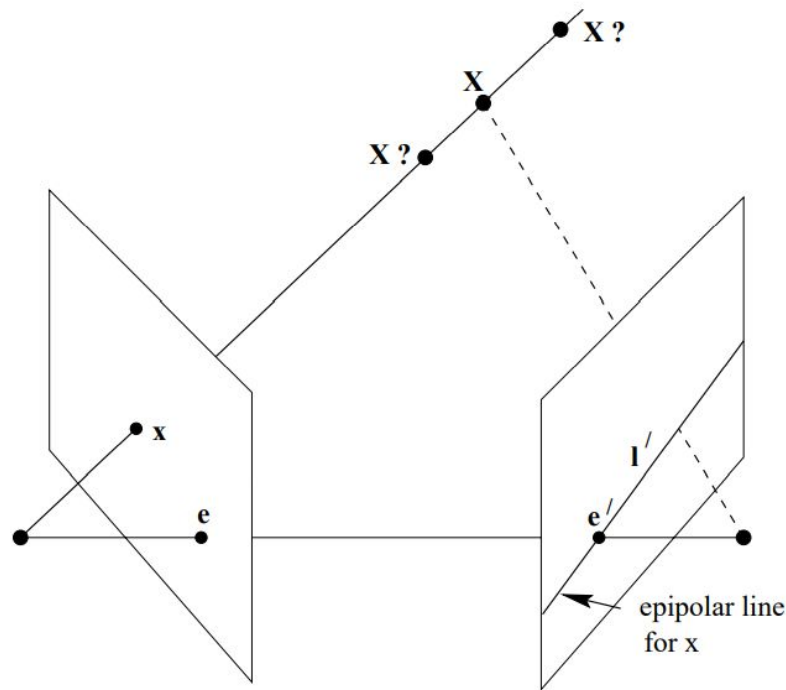
- 1) Каждая пара камер порождает пучок плоскостей, проходящих через ось (C, C')
- 2) e, e' - эпиполюсы
- 3) l' - эпиполярная линия



Эпиполярная геометрия

- 1) Каждая пара камер порождает пучок плоскостей, проходящих через ось (C, C')
- 2) e, e' - эпиполюсы
- 3) l' - эпиполярная линия

$$x \xrightarrow{?} l' ; l' \cdot x' = 0$$
$$l' = Fx ; x'^T Fx = 0$$



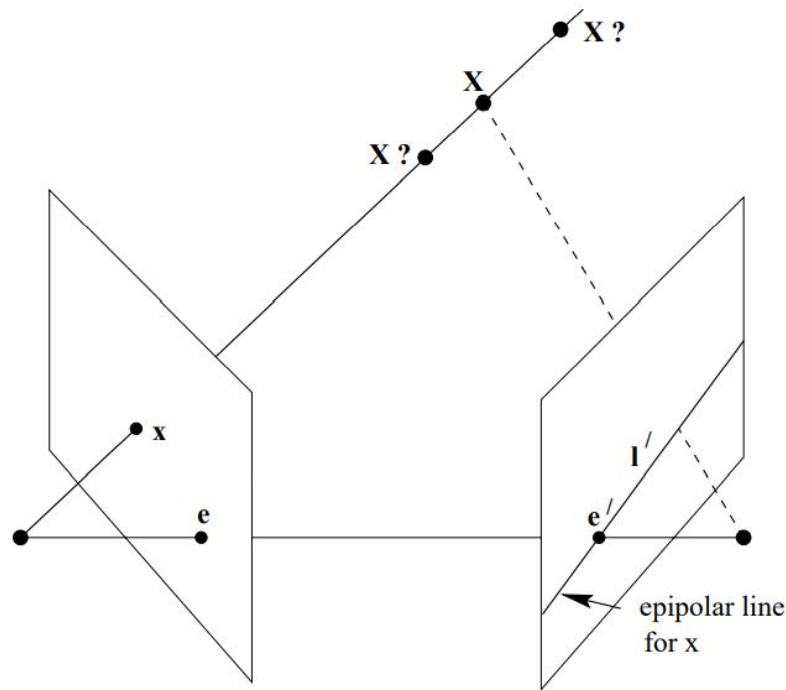
Эпиполярная геометрия

- 1) Каждая пара камер порождает пучок плоскостей, проходящих через ось (C, C')
- 2) e, e' - эпиллюсы
- 3) l' - эпиполярная линия

$$x \xrightarrow{?} l' ; l' \cdot x' = 0$$

$$l' = Fx ; x'^T Fx = 0$$

 фундаментальная матрица



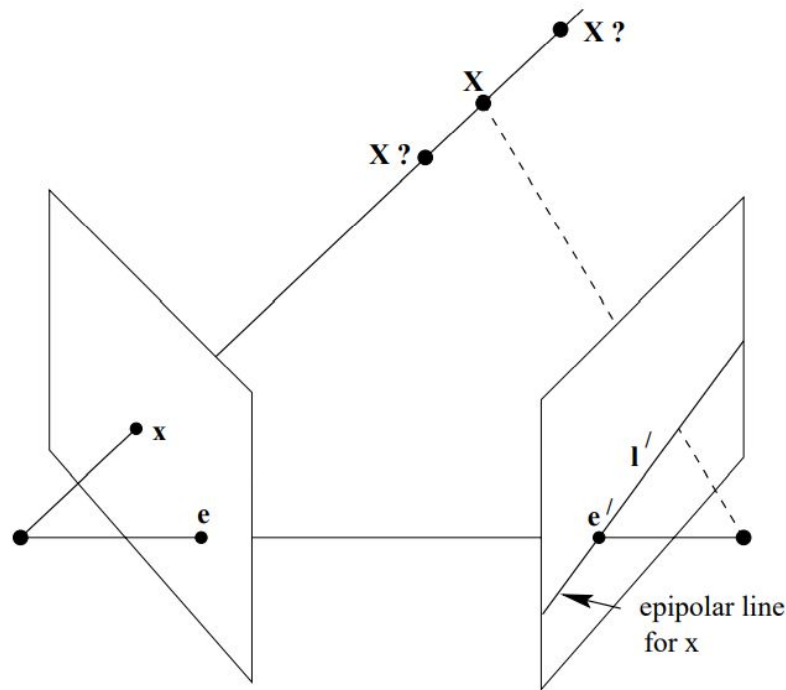
Эпиполярная геометрия

- 1) Каждая пара камер порождает пучок плоскостей, проходящих через ось (C, C')
- 2) e, e' - эпицентры
- 3) l' - эпиполярная линия

$$x \xrightarrow{?} l' ; l' \cdot x' = 0$$

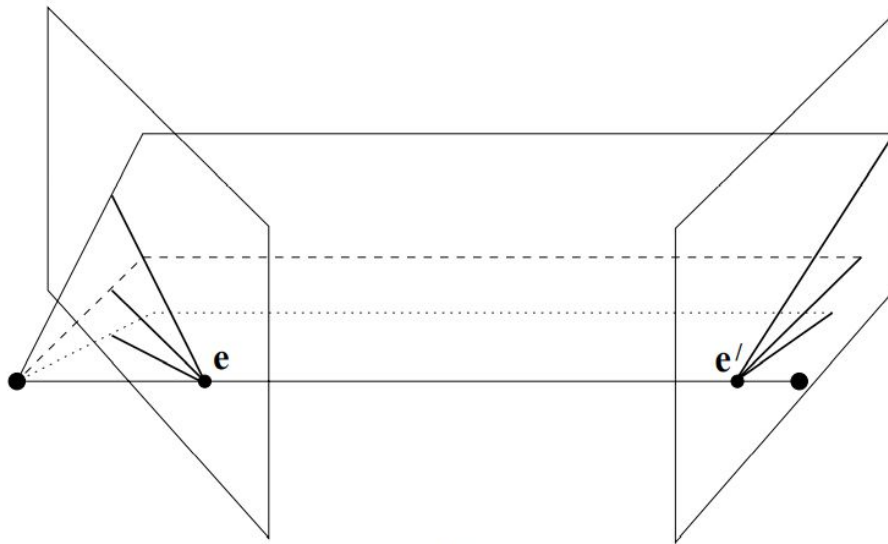
$$l' = Fx ; x'^T Fx = 0$$

 фундаментальная матрица



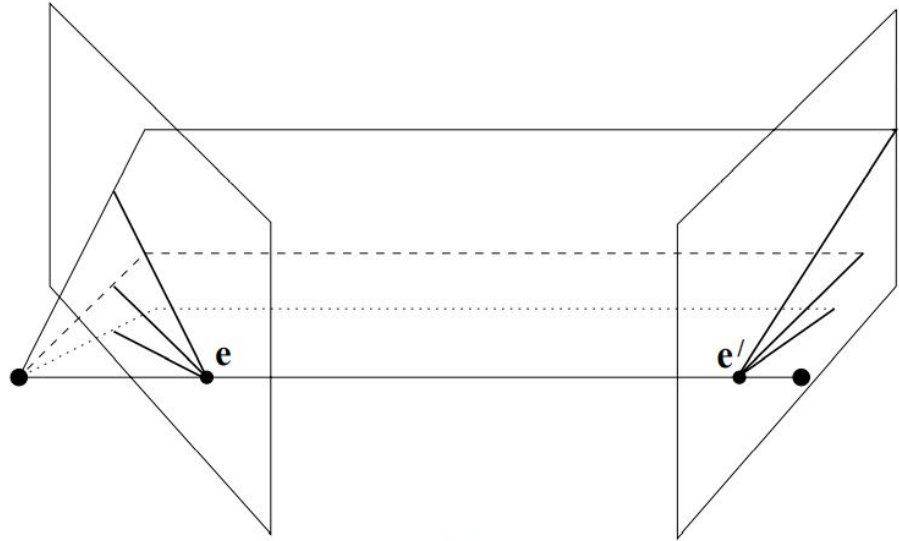
Эпилярная геометрия

1) Эпилярные линии и эпицентр



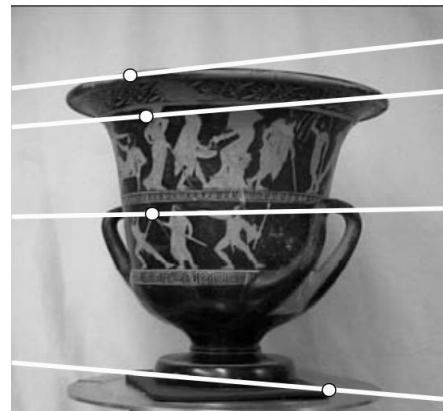
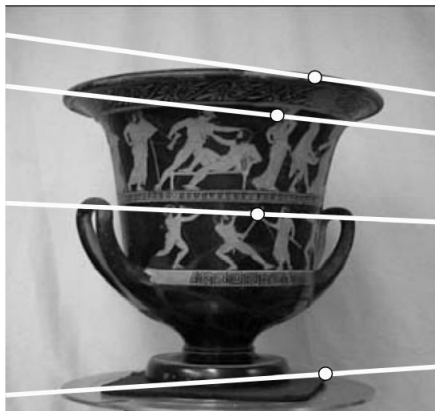
Эпилярная геометрия

- 1) Эпилярные линии и эпицентр
- 2) Как найти эпицентр по матчам и \mathbf{F} ?
- 3) Что нам дает знание эпицентра?



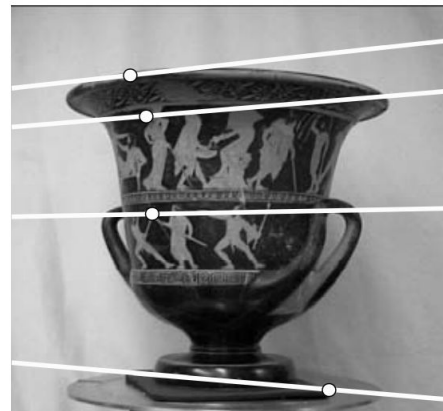
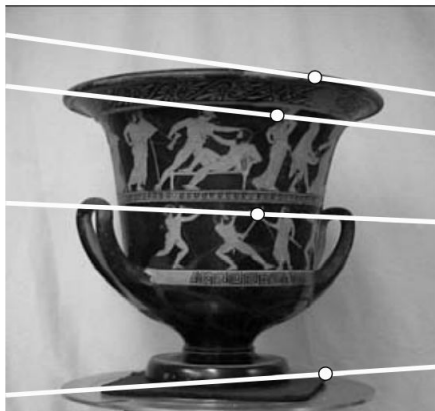
Эпиполярная геометрия

- 1) Эпиполярные линии и эпиполлюс
- 2) Как найти эпиполлюс по матчам и \mathbf{F} ?
- 3) Что нам дает знание эпиполлюса?
- 4) Можно определить направление на вторую камеру



Эпиполярная геометрия

- 1) Эпиполярные линии и эпиполюс
- 2) Как найти эпиполюс по матчам и F ?
- 3) Что нам дает знание эпиполюса?
- 4) Можно определить направление на вторую камеру
- 5) Что если линии не пересекаются?

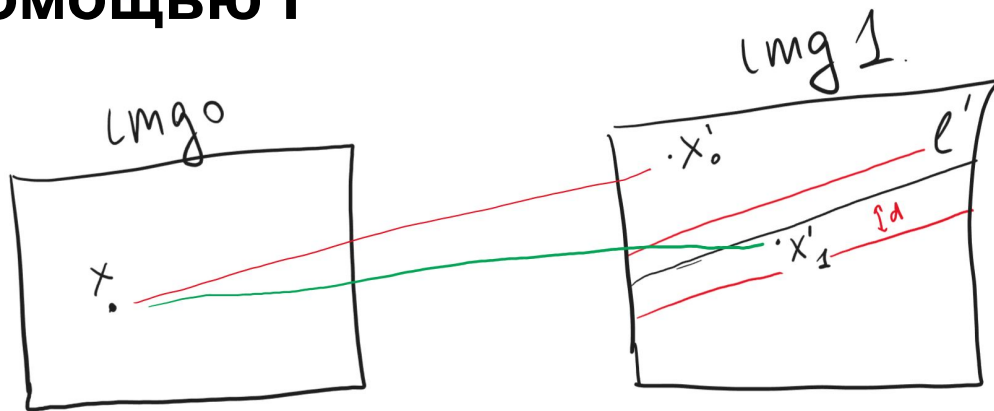


Фильтрация матчей с помощью F

- 1) Если знаем F , можем использовать для фильтрации сопоставлений

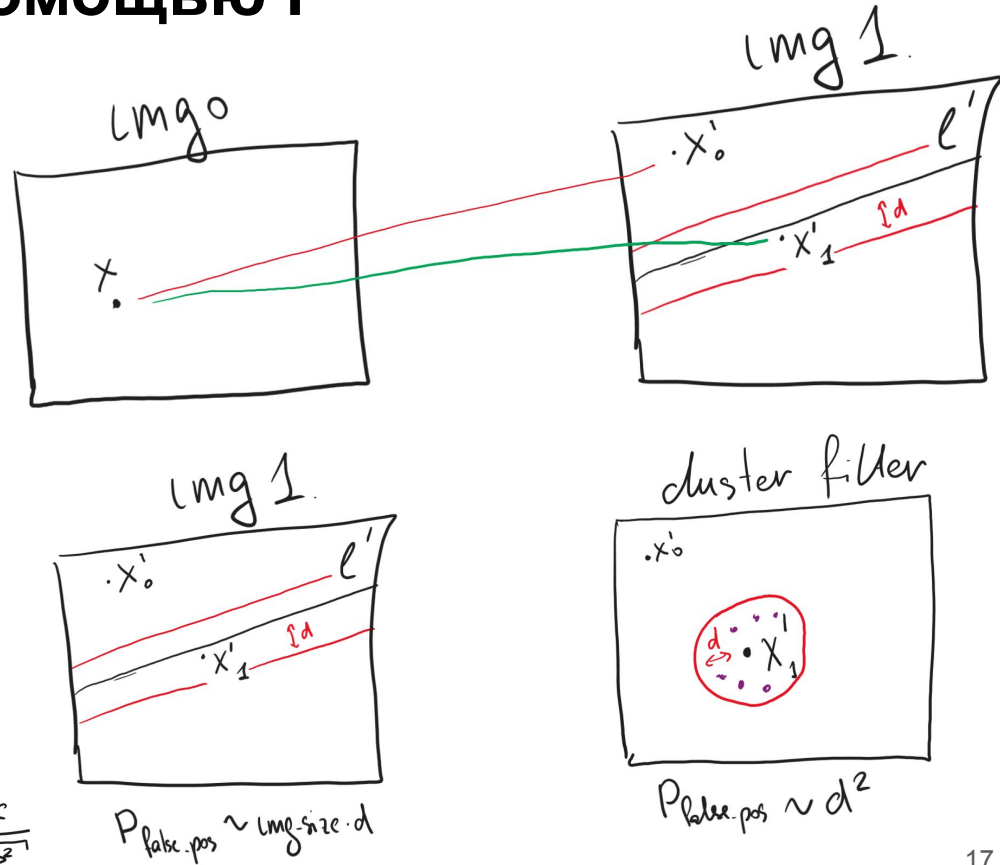
Фильтрация матчей с помощью F

- 1) Если знаем F , можем использовать для фильтрации сопоставлений
- 2) $+$: используем геометрию
- 3) $-$: если сильные искажения, отсеит слишком много матчей



Фильтрация матчей с помощью F

- 1) Если знаем **F**, можем использовать для фильтрации сопоставлений
- 2) +: используем геометрию
- 3) -: если сильные искажения, отсеит слишком много матчей
- 4) -: пропускает слишком много плохих матчей по сравнению с cluster filtering



Фундаментальная матрица

- 1) Как искать если знаем матчи?

Фундаментальная матрица

1) Как искать если знаем матчи? DLT!

11.1 Basic equations

The fundamental matrix is defined by the equation

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (11.1)$$

for any pair of matching points $\mathbf{x} \leftrightarrow \mathbf{x}'$ in two images. Given sufficiently many point matches $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ (at least 7), equation (11.1) can be used to compute the unknown matrix \mathbf{F} . In particular, writing $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$ each point match gives rise to one linear equation in the unknown entries of \mathbf{F} . The coefficients of this equation are easily written in terms of the known coordinates \mathbf{x} and \mathbf{x}' . Specifically, the equation corresponding to a pair of points $(x, y, 1)$ and $(x', y', 1)$ is

$$x'x f_{11} + x'y f_{12} + x' f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0. \quad (11.2)$$

Denote by \mathbf{f} the 9-vector made up of the entries of \mathbf{F} in row-major order. Then (11.2) can be expressed as a vector inner product

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1) \mathbf{f} = 0.$$

From a set of n point matches, we obtain a set of linear equations of the form

$$\mathbf{A} \mathbf{f} = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}. \quad (11.3)$$

Фундаментальная матрица

- 1) Как искать если знаем матчи? **DLT!**
- 2) Однородная система, решаем через **SVD**

11.1 Basic equations

The fundamental matrix is defined by the equation

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (11.1)$$

for any pair of matching points $\mathbf{x} \leftrightarrow \mathbf{x}'$ in two images. Given sufficiently many point matches $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ (at least 7), equation (11.1) can be used to compute the unknown matrix \mathbf{F} . In particular, writing $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$ each point match gives rise to one linear equation in the unknown entries of \mathbf{F} . The coefficients of this equation are easily written in terms of the known coordinates \mathbf{x} and \mathbf{x}' . Specifically, the equation corresponding to a pair of points $(x, y, 1)$ and $(x', y', 1)$ is

$$x'x f_{11} + x'y f_{12} + x' f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0. \quad (11.2)$$

Denote by \mathbf{f} the 9-vector made up of the entries of \mathbf{F} in row-major order. Then (11.2) can be expressed as a vector inner product

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1) \mathbf{f} = 0.$$

From a set of n point matches, we obtain a set of linear equations of the form

$$\mathbf{A} \mathbf{f} = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}. \quad (11.3)$$

Фундаментальная матрица

- 1) Как искать если знаем матчи? **DLT!**
- 2) Однородная система, решаем через **SVD**
- 3) Если нет точного решения, **SVD** дает приближенное (соотв. мин. синг. значению)

11.1 Basic equations

The fundamental matrix is defined by the equation

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (11.1)$$

for any pair of matching points $\mathbf{x} \leftrightarrow \mathbf{x}'$ in two images. Given sufficiently many point matches $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ (at least 7), equation (11.1) can be used to compute the unknown matrix \mathbf{F} . In particular, writing $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$ each point match gives rise to one linear equation in the unknown entries of \mathbf{F} . The coefficients of this equation are easily written in terms of the known coordinates \mathbf{x} and \mathbf{x}' . Specifically, the equation corresponding to a pair of points $(x, y, 1)$ and $(x', y', 1)$ is

$$x'x f_{11} + x'y f_{12} + x' f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0. \quad (11.2)$$

Denote by \mathbf{f} the 9-vector made up of the entries of \mathbf{F} in row-major order. Then (11.2) can be expressed as a vector inner product

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1) \mathbf{f} = 0.$$

From a set of n point matches, we obtain a set of linear equations of the form

$$\mathbf{A} \mathbf{f} = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}. \quad (11.3)$$

Фундаментальная матрица

- 1) Как искать если знаем матчи? **DLT!**
- 2) Однородная система, решаем через **SVD**
- 3) Если нет точного решения, **SVD** дает приближенное (соотв. мин. синг. значению)
- 4) Если матчи шумные?

11.1 Basic equations

The fundamental matrix is defined by the equation

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (11.1)$$

for any pair of matching points $\mathbf{x} \leftrightarrow \mathbf{x}'$ in two images. Given sufficiently many point matches $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ (at least 7), equation (11.1) can be used to compute the unknown matrix \mathbf{F} . In particular, writing $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$ each point match gives rise to one linear equation in the unknown entries of \mathbf{F} . The coefficients of this equation are easily written in terms of the known coordinates \mathbf{x} and \mathbf{x}' . Specifically, the equation corresponding to a pair of points $(x, y, 1)$ and $(x', y', 1)$ is

$$x'x f_{11} + x'y f_{12} + x' f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0. \quad (11.2)$$

Denote by \mathbf{f} the 9-vector made up of the entries of \mathbf{F} in row-major order. Then (11.2) can be expressed as a vector inner product

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1) \mathbf{f} = 0.$$

From a set of n point matches, we obtain a set of linear equations of the form

$$\mathbf{A} \mathbf{f} = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}. \quad (11.3)$$

Фундаментальная матрица

- 1) Как искать если знаем матчи? **DLT!**
- 2) Однородная система, решаем через **SVD**
- 3) Если нет точного решения, **SVD** дает приближенное (соотв. мин. синг. значению)
- 4) Если матчи шумные? **RANSAC!**

11.1 Basic equations

The fundamental matrix is defined by the equation

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \quad (11.1)$$

for any pair of matching points $\mathbf{x} \leftrightarrow \mathbf{x}'$ in two images. Given sufficiently many point matches $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ (at least 7), equation (11.1) can be used to compute the unknown matrix \mathbf{F} . In particular, writing $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{x}' = (x', y', 1)^T$ each point match gives rise to one linear equation in the unknown entries of \mathbf{F} . The coefficients of this equation are easily written in terms of the known coordinates \mathbf{x} and \mathbf{x}' . Specifically, the equation corresponding to a pair of points $(x, y, 1)$ and $(x', y', 1)$ is

$$x'x f_{11} + x'y f_{12} + x' f_{13} + y'x f_{21} + y'y f_{22} + y' f_{23} + x f_{31} + y f_{32} + f_{33} = 0. \quad (11.2)$$

Denote by \mathbf{f} the 9-vector made up of the entries of \mathbf{F} in row-major order. Then (11.2) can be expressed as a vector inner product

$$(x'x, x'y, x', y'x, y'y, y', x, y, 1) \mathbf{f} = 0.$$

From a set of n point matches, we obtain a set of linear equations of the form

$$\mathbf{A} \mathbf{f} = \begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_n x_n & x'_n y_n & x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}. \quad (11.3)$$

Фундаментальная матрица

- 1) Singularity constraint:
 $\text{rank}(\mathbf{F}) = 2$

Фундаментальная матрица

- 1) Singularity constraint:
rank(F) = 2

$$x'^T F x = 0$$

$\forall l$ проходящая через $e \Rightarrow$
 $\Rightarrow \forall x: e'^T F x = 0$
 $(e'^T F) x = 0$
 $g x = 0 \Rightarrow g = 0$

Фундаментальная матрица

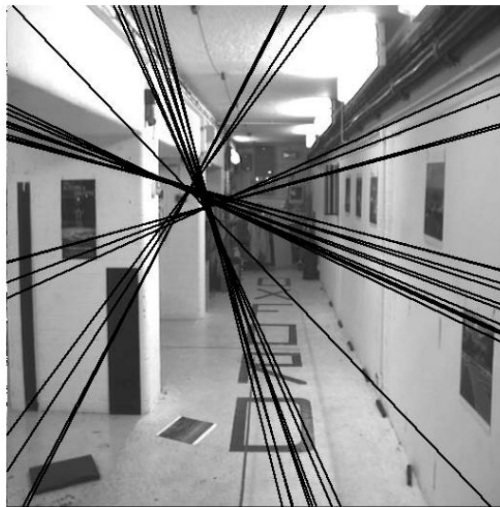
- 1) Singularity constraint:
 $\text{rank}(F) = 2$
- 2) Иначе эпилептические
линии не будут
пересекаться в
эпиполюсе

$$x'^T F x = 0$$

$\forall l$ проходящая через $e \Rightarrow$
 $\Rightarrow \forall x: e'^T F x = 0$
 $(e'^T F) x = 0$
 $g x = 0 \Rightarrow g = 0$

Фундаментальная матрица

- 1) Singularity constraint:
 $\text{rank}(\mathbf{F}) = 2$
- 2) Иначе эпиллярные
линии не будут
пересекаться в
эпиполюсе



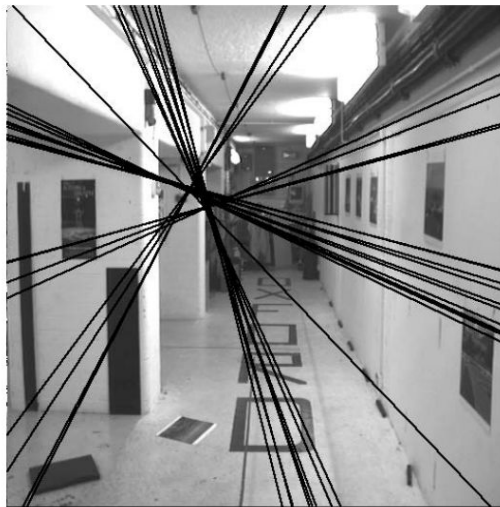
non-singular



singular

Фундаментальная матрица

- 1) Singularity constraint:
 $\text{rank}(\mathbf{F}) = 2$
- 2) Иначе эпиллярные линии не будут пересекаться в эпицентре
- 3) Что делать если решили систему, но **\mathbf{F}** получилась не сингулярная?



non-singular



singular

Фундаментальная матрица

- 1) Singularity constraint:
 $\text{rank}(\mathbf{F}) = 2$
- 2) Иначе эпиллярные линии не будут пересекаться в эпицентре
- 3) Что делать если решили систему, но \mathbf{F} получилась не сингулярная? SVD!



non-singular



singular

Фундаментальная матрица

- 1) Singularity constraint:
 $\text{rank}(\mathbf{F}) = 2$
- 2) Иначе эпиллярные линии не будут пересекаться в эпицентре
- 3) Что делать если решили систему, но **\mathbf{F}** получилась не сингулярная? **SVD!**



\mathbf{F}



\mathbf{F}'

$$\mathbf{F} = \mathbf{U} \mathbf{D} \mathbf{V}^T = \mathbf{U} \begin{pmatrix} r & s & t \end{pmatrix} \mathbf{V}^T \rightarrow \mathbf{U} \begin{pmatrix} r & s & 0 \end{pmatrix} \mathbf{V}^T = \mathbf{F}'$$

$r \geq s \geq t$

Фундаментальная матрица

- 1) Разрешение камер большое,
в системе присутствуют
члены с большим разбросом

$$x'x_{f_{11}} + x'y_{f_{12}} + x'f_{13} + \underline{y'x_{f_{21}}} + y'y_{f_{22}} + \underline{y'f_{23}} + xf_{31} + yf_{32} + \underline{f_{33}} = 0. \quad (11.2)$$

Фундаментальная матрица

- 1) Разрешение камер большое, в системе присутствуют члены с большим разбросом
- 2) Приводит к плохой численной точности решения линейной системы

$$x'x_{f11} + x'y_{f12} + x'f_{13} + \underline{y'x_{f21}} + y'y_{f22} + \underline{y'f_{23}} + xf_{31} + yf_{32} + \underline{f_{33}} = 0. \quad (11.2)$$

Фундаментальная матрица

- 1) Разрешение камер большое, в системе присутствуют члены с большим разбросом
- 2) Приводит к плохой численной точности решения линейной системы
- 3) Нужно нормализовать!

$$x'x_{f11} + x'y_{f12} + x'f_{13} + \underline{y'x_{f21}} + \underline{y'y_{f22}} + \underline{y'f_{23}} + xf_{31} + yf_{32} + \underline{f_{33}} = 0. \quad (11.2)$$

$$\{x\} \rightarrow \text{centroid dispersion} \rightarrow TN: x \rightarrow \hat{x} : \text{centroid}(\hat{x})=0, \text{dispersion}(\hat{x})=\sqrt{2}$$

$$\hat{x}'^T \hat{F} \hat{x} = 0 \Leftrightarrow (TN' x')^T \hat{F} (TN x) = 0$$

$$\Leftrightarrow x'^T \underbrace{(TN'^T \hat{F} TN)}_F x = 0$$

Фундаментальная матрица

1) \mathbf{F} можно разложить на матрицы камер (\mathbf{P} , \mathbf{P}')

$$\mathbf{P} = \mathbf{K}(\mathbf{R} | \mathbf{t})$$

внутр. внешн.

Фундаментальная матрица

- 1) **F** можно разложить на матрицы камер (**P**, **P'**)
- 2) То есть определить положение камер относительно друг друга!

$$P = K(R | t)$$

внутр. внешн.

Фундаментальная матрица

- 1) **F** можно разложить на матрицы камер (**P**, **P'**)
- 2) То есть определить положение камер относительно друг друга!
- 3) Если не повезет, то может выиграть **F** соответствующая неверной калибровке (а есть еще дисторсии..)

$$P = K(R | t)$$

внутр. внешн.

Фундаментальная матрица

- 1) F можно разложить на матрицы камер (P, P')
- 2) То есть определить положение камер относительно друг друга!
- 3) Если не повезет, то может выиграть F соответствующая неверной калибровке
- 4) Из-за того, что матрица F включает в себя калибровку, разложение получается с точностью до проективного преобразования (Hartley, Zisserman, теорема 9.10)

$$P = K(R|t)$$

внутр. внешн.

Theorem 9.10. *Let F be a fundamental matrix and let (P, P') and (\tilde{P}, \tilde{P}') be two pairs of camera matrices such that F is the fundamental matrix corresponding to each of these pairs. Then there exists a non-singular 4×4 matrix H such that $\tilde{P} = PH$ and $\tilde{P}' = P'H$.*

Существенная (Essential) матрица

1) Существенная (Essential) матрица E :

$$\tilde{x}^T E \tilde{x} = 0$$

$$\tilde{x}' = K^{-1} x'$$

$$\tilde{x} = K^{-1} x$$

$$(K^{-1} x')^T E (K^{-1} x) = 0$$

$$x'^T \underbrace{(K^{-1T} E K^{-1})}_F x = 0$$

$$E = K^T F K$$

Существенная (Essential) матрица

- 1) Существенная (Essential) матрица E :
- 2) Позволяет определить (P, P') с точностью до **масштаба**!

$$\begin{aligned}\tilde{x}^T E \tilde{x} &= 0 \\ \tilde{x}' &= K'^{-1} x' \\ \tilde{x} &= K^{-1} x\end{aligned}$$

$$\begin{aligned}(K'^{-1} x')^T E (K^{-1} x) &= 0 \\ x'^T (\underbrace{K'^{-1T} E K^{-1}}_F) x &= 0 \\ E &= K^T F K\end{aligned}$$

Существенная (Essential) матрица

- 1) Существенная (Essential) матрица **E**:
- 2) Позволяет определить (**P**, **P'**) с точностью до **масштаба**!
- 3) Утверждение: существует (нелинейный) алгоритм поиска матрицы **E** напрямую, по пяти сопоставлениям (а не 8 как в случае с **DLT**) (Nister 5 point algorithm)

$$\begin{aligned}\tilde{x}^T E \tilde{x} &= 0 \\ \tilde{x}' &= K'^{-1} x' \\ \tilde{x} &= K^{-1} x\end{aligned}$$

$$\begin{aligned}(K'^{-1} x')^T E (K^{-1} x) &= 0 \\ x'^T (\underbrace{K'^{-T} E K^{-1}}_F) x &= 0 \\ E &= K^T F K\end{aligned}$$

Существенная (Essential) матрица

- 1) Существенная (Essential) матрица **E**:
- 2) Позволяет определить (P, P') с точностью до **масштаба**!
- 3) Утверждение: существует (нелинейный) алгоритм поиска матрицы **E** напрямую, по пяти сопоставлениям (а не 8 как в случае с **DLT**) (Nister 5 point algorithm)
 - a) Удобно для RANSAC
 - b) Не пытается подогнать калибровку под данные

$$\begin{aligned}\tilde{x}^T E \tilde{x} &= 0 \\ \tilde{x}' &= K'^{-1} x' \\ \tilde{x} &= K^{-1} x\end{aligned}$$

$$\begin{aligned}(K'^{-1} x')^T E (K^{-1} x) &= 0 \\ x'^T (\underbrace{K'^{-T} E K^{-1}}_F) x &= 0 \\ E &= K^T F K\end{aligned}$$

Разложение существенной матрицы

- 1) Выберем систему координат так, чтобы первая камера имела тривиальную матрицу \mathbf{P}
- 2) Тогда нужно найти только относительный сдвиг и поворот

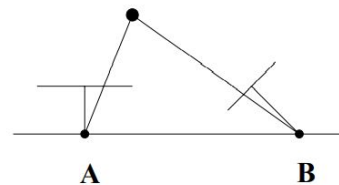
Result 9.19. *For a given essential matrix $\mathbf{E} = \mathbf{U} \text{diag}(1, 1, 0) \mathbf{V}^T$, and first camera matrix $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$, there are four possible choices for the second camera matrix \mathbf{P}' , namely*

$$\mathbf{P}' = [\mathbf{U}\mathbf{W}\mathbf{V}^T \mid +\mathbf{u}_3] \text{ or } [\mathbf{U}\mathbf{W}\mathbf{V}^T \mid -\mathbf{u}_3] \text{ or } [\mathbf{U}\mathbf{W}^T\mathbf{V}^T \mid +\mathbf{u}_3] \text{ or } [\mathbf{U}\mathbf{W}^T\mathbf{V}^T \mid -\mathbf{u}_3].$$

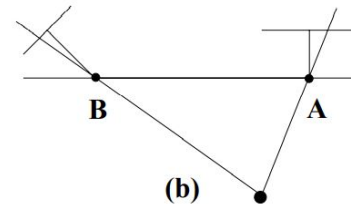
(см. 9.6.2 Extraction of cameras from the essential matrix)

Разложение существенной матрицы

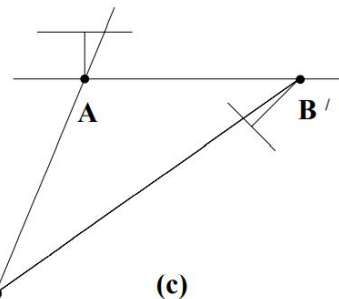
- 1) Матрица \mathbf{P}' определяется с точностью до четырех вариантов
- 2) Выбрать правильный можно с помощью depth-теста: триангулированная точка должна быть спереди для обеих камер



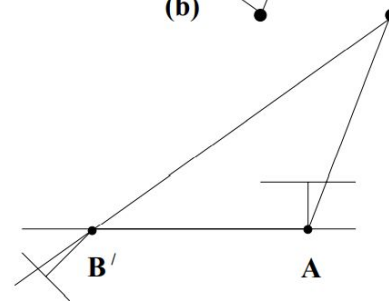
(a)



(b)



(c)



(d)

Fig. 9.12. The four possible solutions for calibrated reconstruction from E. Between the left and right sides there is a baseline reversal. Between the top and bottom rows camera B rotates 180° about the baseline. Note, only in (a) is the reconstructed point in front of both cameras.

Триангуляция

- 1) Знаем положение хотя бы двух камер и матчи, хотим получить 3D точки

Триангуляция

- 1) Знаем положение хотя бы двух камер и матчи, хотим получить 3D точки
- 2) Что делать?

Триангуляция

- 1) Знаем положение хотя бы двух камер и матчи, хотим получить 3D точки
- 2) Что делать? **DLT!**

12.2 Linear triangulation methods

In this section, we describe simple linear triangulation methods. As usual the estimated point does not exactly satisfy the geometric relations, and is not an optimal estimate.

The linear triangulation method is the direct analogue of the DLT method described in section 4.1(p88). In each image we have a measurement $\mathbf{x} = \mathbf{P}\mathbf{X}$, $\mathbf{x}' = \mathbf{P}'\mathbf{X}$, and these equations can be combined into a form $\mathbf{A}\mathbf{X} = \mathbf{0}$, which is an equation linear in \mathbf{X} .

First the homogeneous scale factor is eliminated by a cross product to give three equations for each image point, of which two are linearly independent. For example for the first image, $\mathbf{x} \times (\mathbf{P}\mathbf{X}) = \mathbf{0}$ and writing this out gives

$$x(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{1\top}\mathbf{X}) = 0$$

$$y(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{2\top}\mathbf{X}) = 0$$

$$x(\mathbf{p}^{2\top}\mathbf{X}) - y(\mathbf{p}^{1\top}\mathbf{X}) = 0$$

where $\mathbf{p}^{i\top}$ are the rows of \mathbf{P} . These equations are *linear* in the components of \mathbf{X} .

An equation of the form $\mathbf{A}\mathbf{X} = \mathbf{0}$ can then be composed, with

$$\mathbf{A} = \begin{bmatrix} x\mathbf{p}^{3\top} - \mathbf{p}^{1\top} \\ y\mathbf{p}^{3\top} - \mathbf{p}^{2\top} \\ x'\mathbf{p}^{3\top} - \mathbf{p}'^{1\top} \\ y'\mathbf{p}^{3\top} - \mathbf{p}'^{2\top} \end{bmatrix}$$

where two equations have been included from each image, giving a total of four equations in four homogeneous unknowns. This is a redundant set of equations, since the solution is determined only up to scale. Two ways of solving the set of equations of the form $\mathbf{A}\mathbf{X} = \mathbf{0}$ were discussed in section 4.1(p88) and will be considered again here.

Триангуляция

- 1) Знаем положение хотя бы двух камер и матчи, хотим получить 3D точки
- 2) Что делать? **DLT!**
- 3) Получили однородную систему вида **$AX=0$**
- 4) **SVD**

12.2 Linear triangulation methods

In this section, we describe simple linear triangulation methods. As usual the estimated point does not exactly satisfy the geometric relations, and is not an optimal estimate.

The linear triangulation method is the direct analogue of the DLT method described in section 4.1(p88). In each image we have a measurement $\mathbf{x} = P\mathbf{X}$, $\mathbf{x}' = P'\mathbf{X}$, and these equations can be combined into a form $A\mathbf{X} = 0$, which is an equation linear in \mathbf{X} .

First the homogeneous scale factor is eliminated by a cross product to give three equations for each image point, of which two are linearly independent. For example for the first image, $\mathbf{x} \times (P\mathbf{X}) = 0$ and writing this out gives

$$x(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{1T}\mathbf{X}) = 0$$

$$y(\mathbf{p}^{3T}\mathbf{X}) - (\mathbf{p}^{2T}\mathbf{X}) = 0$$

$$x(\mathbf{p}^{2T}\mathbf{X}) - y(\mathbf{p}^{1T}\mathbf{X}) = 0$$

where \mathbf{p}^{iT} are the rows of P . These equations are *linear* in the components of \mathbf{X} .

An equation of the form $A\mathbf{X} = 0$ can then be composed, with

$$A = \begin{bmatrix} x\mathbf{p}^{3T} - \mathbf{p}^{1T} \\ y\mathbf{p}^{3T} - \mathbf{p}^{2T} \\ x'\mathbf{p}^{3T} - \mathbf{p}'^{1T} \\ y'\mathbf{p}^{3T} - \mathbf{p}'^{2T} \end{bmatrix}$$

where two equations have been included from each image, giving a total of four equations in four homogeneous unknowns. This is a redundant set of equations, since the solution is determined only up to scale. Two ways of solving the set of equations of the form $A\mathbf{X} = 0$ were discussed in section 4.1(p88) and will be considered again here.

Триангуляция

- 1) Знаем положение хотя бы двух камер и матчи, хотим получить 3D точки
- 2) Что делать? **DLT!**
- 3) Получили однородную систему вида **$AX=0$**
- 4) **SVD**
- 5) Если наблюдений больше двух, получим наилучшее решение в смысле наименьших квадратов

12.2 Linear triangulation methods

In this section, we describe simple linear triangulation methods. As usual the estimated point does not exactly satisfy the geometric relations, and is not an optimal estimate.

The linear triangulation method is the direct analogue of the DLT method described in section 4.1(p88). In each image we have a measurement $\mathbf{x} = \mathbf{P}\mathbf{X}$, $\mathbf{x}' = \mathbf{P}'\mathbf{X}$, and these equations can be combined into a form $\mathbf{A}\mathbf{X} = \mathbf{0}$, which is an equation linear in \mathbf{X} .

First the homogeneous scale factor is eliminated by a cross product to give three equations for each image point, of which two are linearly independent. For example for the first image, $\mathbf{x} \times (\mathbf{P}\mathbf{X}) = \mathbf{0}$ and writing this out gives

$$x(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{1\top}\mathbf{X}) = 0$$

$$y(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{2\top}\mathbf{X}) = 0$$

$$x(\mathbf{p}^{2\top}\mathbf{X}) - y(\mathbf{p}^{1\top}\mathbf{X}) = 0$$

where $\mathbf{p}^{i\top}$ are the rows of \mathbf{P} . These equations are *linear* in the components of \mathbf{X} .

An equation of the form $\mathbf{A}\mathbf{X} = \mathbf{0}$ can then be composed, with

$$\mathbf{A} = \begin{bmatrix} x\mathbf{p}^{3\top} - \mathbf{p}^{1\top} \\ y\mathbf{p}^{3\top} - \mathbf{p}^{2\top} \\ x'\mathbf{p}'^{3\top} - \mathbf{p}'^{1\top} \\ y'\mathbf{p}'^{3\top} - \mathbf{p}'^{2\top} \end{bmatrix}$$

where two equations have been included from each image, giving a total of four equations in four homogeneous unknowns. This is a redundant set of equations, since the solution is determined only up to scale. Two ways of solving the set of equations of the form $\mathbf{A}\mathbf{X} = \mathbf{0}$ were discussed in section 4.1(p88) and will be considered again here.

Предварительный итог

- 1) Научились выравнивать две камеры и строить для них трехмерное облако точек с помощью триангуляции

Предварительный итог

- 1) Научились выравнивать две камеры и строить для них трехмерное облако точек с помощью триангуляции
- 2) Как выровнять три камеры и больше?

Предварительный итог

- 1) Научились выравнивать две камеры и строить для них трехмерное облако точек с помощью триангуляции
- 2) Как выровнять три камеры и больше?
- 3) Нужно научиться определять матрицу камеры P по облаку точек

Резекция

- 1) Есть облако точек, какие-то из них нашлись в матчах текущей камеры

Резекция

- 1) Есть облако точек, какие-то из них нашлись в матчах текущей камеры
- 2) **Резекция:** выравнивание камеры по матчам и облаку точек

Резекция

- 1) Есть облако точек, какие-то из них нашлись в матчах текущей камеры
- 2) **Резекция:** выравнивание камеры по матчам и облаку точек
- 3) Как делать?

Резекция

- 1) Есть облако точек, какие-то из них нашлись в матчах текущей камеры
- 2) **Резекция:** выравнивание камеры по матчам и облаку точек
- 3) Как делать? DLT!

7.1 Basic equations

We assume a number of point correspondences $X_i \leftrightarrow x_i$ between 3D points X_i and 2D image points x_i are given. We are required to find a camera matrix P , namely a 3×4 matrix such that $x_i = PX_i$ for all i . The similarity of this problem with that of computing a 2D projective transformation H , treated in chapter 4, is evident. The only difference is the dimension of the problem. In the 2D case the matrix H has dimension 3×3 , whereas in the present case, P is a 3×4 matrix. As one may expect, much of the material from chapter 4 applies almost unchanged to the present case.

As in section 4.1(p88) for each correspondence $X_i \leftrightarrow x_i$ we derive a relationship

$$\begin{bmatrix} 0^T & -w_i X_i^T & y_i X_i^T \\ w_i X_i^T & 0^T & -x_i X_i^T \\ -y_i X_i^T & x_i X_i^T & 0^T \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = 0. \quad (7.1)$$

where each P^{i^T} is a 4-vector, the i -th row of P . Alternatively, one may choose to use only the first two equations:

$$\begin{bmatrix} 0^T & -w_i X_i^T & y_i X_i^T \\ w_i X_i^T & 0^T & -x_i X_i^T \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = 0 \quad (7.2)$$

since the three equations of (7.1) are linearly dependent. From a set of n point correspondences, we obtain a $2n \times 12$ matrix A by stacking up the equations (7.2) for each correspondence. The projection matrix P is computed by solving the set of equations $Ap = 0$, where p is the vector containing the entries of the matrix P .

Резекция

- 1) Есть облако точек, какие-то из них нашлись в матчах текущей камеры
- 2) **Резекция:** выравнивание камеры по матчам и облаку точек
- 3) Как делать? **DLT, SVD...**

7.1 Basic equations

We assume a number of point correspondences $X_i \leftrightarrow x_i$ between 3D points X_i and 2D image points x_i are given. We are required to find a camera matrix P , namely a 3×4 matrix such that $x_i = PX_i$ for all i . The similarity of this problem with that of computing a 2D projective transformation H , treated in chapter 4, is evident. The only difference is the dimension of the problem. In the 2D case the matrix H has dimension 3×3 , whereas in the present case, P is a 3×4 matrix. As one may expect, much of the material from chapter 4 applies almost unchanged to the present case.

As in section 4.1(p88) for each correspondence $X_i \leftrightarrow x_i$ we derive a relationship

$$\begin{bmatrix} 0^T & -w_i X_i^T & y_i X_i^T \\ w_i X_i^T & 0^T & -x_i X_i^T \\ -y_i X_i^T & x_i X_i^T & 0^T \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = 0. \quad (7.1)$$

where each P^{i^T} is a 4-vector, the i -th row of P . Alternatively, one may choose to use only the first two equations:

$$\begin{bmatrix} 0^T & -w_i X_i^T & y_i X_i^T \\ w_i X_i^T & 0^T & -x_i X_i^T \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = 0 \quad (7.2)$$

since the three equations of (7.1) are linearly dependent. From a set of n point correspondences, we obtain a $2n \times 12$ matrix A by stacking up the equations (7.2) for each correspondence. The projection matrix P is computed by solving the set of equations $Ap = 0$, where p is the vector containing the entries of the matrix P .

Резекция

- 1) Есть облако точек, какие-то из них нашлись в матчах текущей камеры
- 2) **Резекция:** выравнивание камеры по матчам и облаку точек
- 3) Как делать? **DLT, SVD...**
- 4) Получили **P**, теперь триангулируем старые и новые точки

7.1 Basic equations

We assume a number of point correspondences $X_i \leftrightarrow x_i$ between 3D points X_i and 2D image points x_i are given. We are required to find a camera matrix P , namely a 3×4 matrix such that $x_i = PX_i$ for all i . The similarity of this problem with that of computing a 2D projective transformation H , treated in chapter 4, is evident. The only difference is the dimension of the problem. In the 2D case the matrix H has dimension 3×3 , whereas in the present case, P is a 3×4 matrix. As one may expect, much of the material from chapter 4 applies almost unchanged to the present case.

As in section 4.1(p88) for each correspondence $X_i \leftrightarrow x_i$ we derive a relationship

$$\begin{bmatrix} 0^T & -w_i X_i^T & y_i X_i^T \\ w_i X_i^T & 0^T & -x_i X_i^T \\ -y_i X_i^T & x_i X_i^T & 0^T \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = 0. \quad (7.1)$$

where each P^{i^T} is a 4-vector, the i -th row of P . Alternatively, one may choose to use only the first two equations:

$$\begin{bmatrix} 0^T & -w_i X_i^T & y_i X_i^T \\ w_i X_i^T & 0^T & -x_i X_i^T \end{bmatrix} \begin{pmatrix} P^1 \\ P^2 \\ P^3 \end{pmatrix} = 0 \quad (7.2)$$

since the three equations of (7.1) are linearly dependent. From a set of n point correspondences, we obtain a $2n \times 12$ matrix A by stacking up the equations (7.2) for each correspondence. The projection matrix P is computed by solving the set of equations $Ap = 0$, where p is the vector containing the entries of the matrix P .

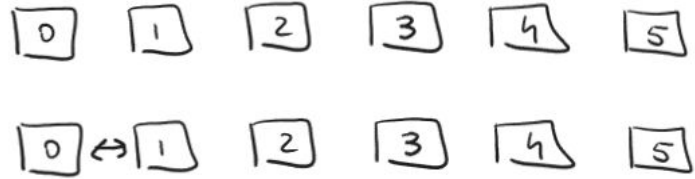
Общий алгоритм SFM

1) На вход кадры и матчи



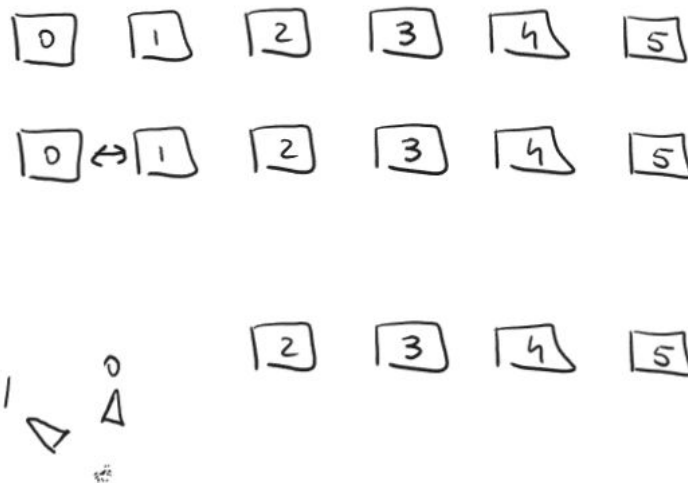
Общий алгоритм SFM

- 1) На вход кадры и матчи
- 2) Выбор первой пары (часто нетривиальный)



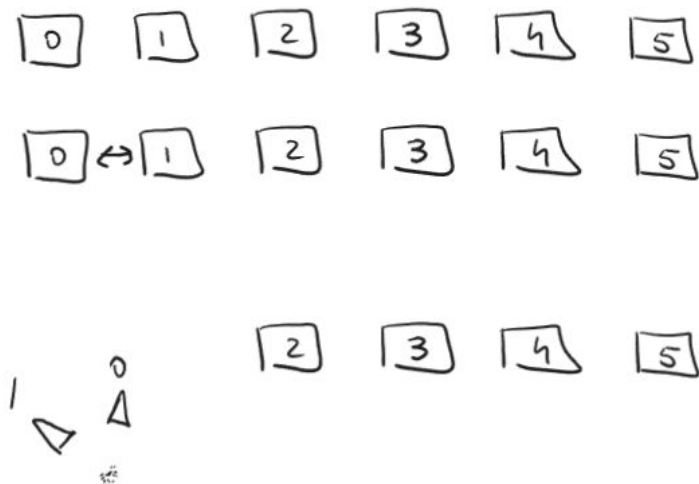
Общий алгоритм SFM

- 1) На вход кадры и матчи
- 2) Выбор первой пары (часто нетривиальный)
- 3) Выравнивание через essential матрицу, тривиальная калибровка



Общий алгоритм SFM

- 1) На вход кадры и матчи
- 2) Выбор первой пары (часто нетривиальный)
- 3) Выравнивание через essential матрицу, тривиальная калибровка
- 4) Триангуляция (добавление точек)
- 5) *Уточнение положения и калибровки (TBD)*
- 6) Резекция (добавление камер)
- 7) goto шаг 4



Общий алгоритм SFM

- 1) На вход кадры и матчи
- 2) Выбор первой пары (часто нетривиальный)
- 3) Выравнивание через essential матрицу, тривиальная калибровка
- 4) Триангуляция (добавление точек)
- 5) *Уточнение положения и калибровки (TBD)*
- 6) Резекция (добавление камер)
- 7) goto шаг 4



Общий алгоритм SFM

- 1) На вход кадры и матчи
- 2) Выбор первой пары (часто нетривиальный)
- 3) Выравнивание через essential матрицу, тривиальная калибровка
- 4) Триангуляция (добавление точек)
- 5) *Уточнение положения и калибровки (TBD)*
- 6) Резекция (добавление камер)
- 7) goto шаг 4



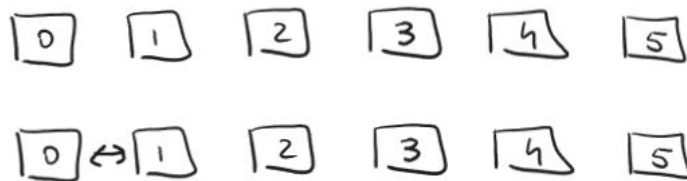
Общий алгоритм SFM

- 1) На вход кадры и матчи
- 2) Выбор первой пары (часто нетривиальный)
- 3) Выравнивание через essential матрицу, тривиальная калибровка
- 4) Триангуляция (добавление точек)
- 5) *Уточнение положения и калибровки (TBD)*
- 6) Резекция (добавление камер)
- 7) goto шаг 4



Общий алгоритм SFM

- 1) На вход кадры и матчи
- 2) Выбор первой пары (часто нетривиальный)
- 3) Выравнивание через essential матрицу, тривиальная калибровка
- 4) Триангуляция (добавление точек)
- 5) *Уточнение положения и калибровки (TBD)*
- 6) Резекция (добавление камер)
- 7) goto шаг 4



Выбор первой пары

- 1) Как можно выбрать пару, с которой начинать процесс?

Выбор первой пары

- 1) Как можно выбрать пару, с которой начинать процесс?
- 2) Наибольшее количество матчей?

Выбор первой пары

- 1) Как можно выбрать пару, с которой начинать процесс?
- 2) Наибольшее количество матчей?
- 3) Наибольшее количество матчей + minimal baseline angle

Выбор первой пары

- 1) Как можно выбрать пару, с которой начинать процесс?
- 2) Наибольшее количество матчей?
- 3) Наибольшее количество матчей + minimal baseline angle
- 4) Заглядывание в будущее: как хорошо будут выравниваться несколько первых кадров если выберем эту пару?

Global SFM

- 1) Существуют подходы,
избавленные от ответственного
выбора первой пары

Global SFM

- 1) Существуют подходы,
избавленные от ответственного
выбора первой пары
- 2) Более устойчивы ко входным
данным

Global SFM

- 1) Существуют подходы, избавленные от ответственного выбора первой пары
- 2) Более устойчивы ко входным данным
- 3) Необходима хорошая начальная калибровка

Global SFM

- 1) Существуют подходы, избавленные от ответственного выбора первой пары
- 2) Более устойчивы ко входным данным
- 3) Необходима хорошая начальная калибровка
- 4) На практике аккуратно реализованный выбор первой пары работает хорошо

ДЗ

- 1) Реализовать общий алгоритм SFM

ДЗ

- 1) Реализовать общий алгоритм SFM
 - а) без уточнения положений и калибровки

ДЗ

- 1) Реализовать общий алгоритм SFM
 - a) без уточнения положений и калибровки
 - b) перед каждой системой полезно сесть с листочком и аккуратно пройти по шагам из Hartley & Zisserman

ДЗ

- 1) Реализовать общий алгоритм SFM
 - a) без уточнения положений и калибровки
 - b) перед каждой системой полезно сесть с листочком и аккуратно пройти по шагам из Hartley & Zisserman
 - c) довольно объемно, лучше слишком сильно не откладывать

Ссылки

- 1) [Awesome SFM](#) (в том числе про Global SFM можно почитать там)
- 2) [OpenMVG](#)
- 3) [Hartley, Zisserman](#)

Следующая лекция онлайн!

Вопросы?

