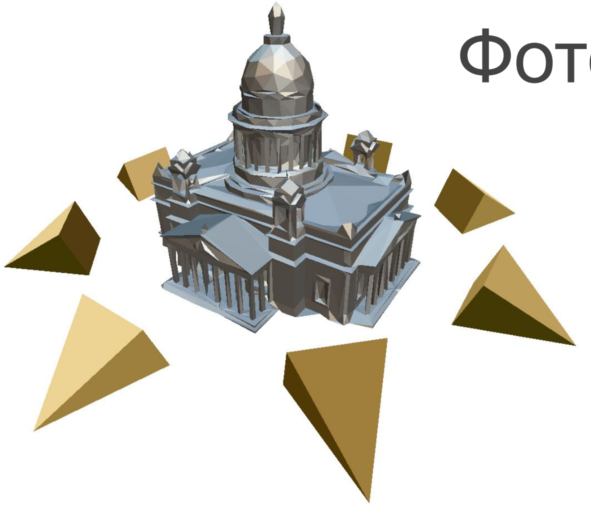
Введение в фотограмметрию Structure from motion (SFM)



Фотограмметрия. Лекция 6

- Эпиполярная геометрия
- Фундаментальная матрица
- Триангуляция, резекция

Симиютин Борис

simiyutin.boris@yandex.ru

Мотивация

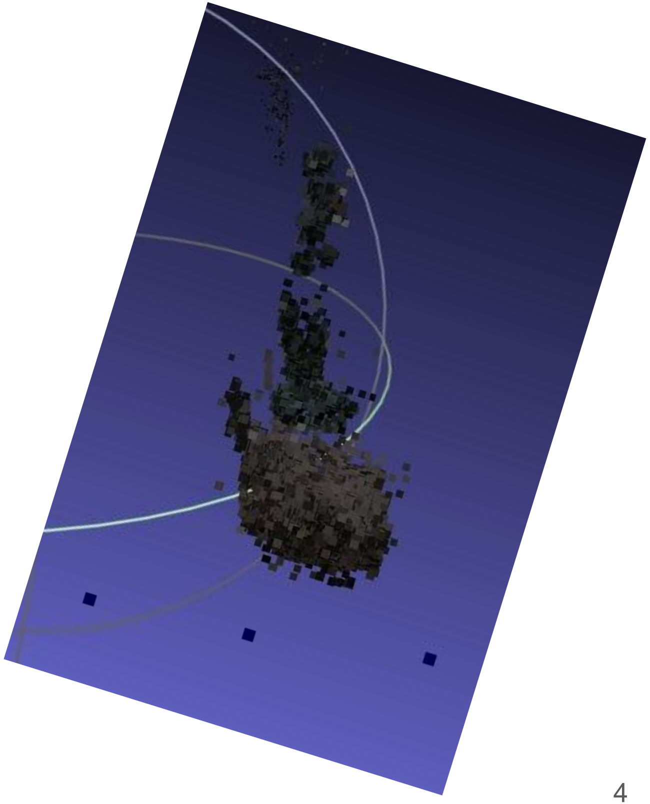
1) Хотим по матчам определять взаимное расположение камер

Мотивация

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

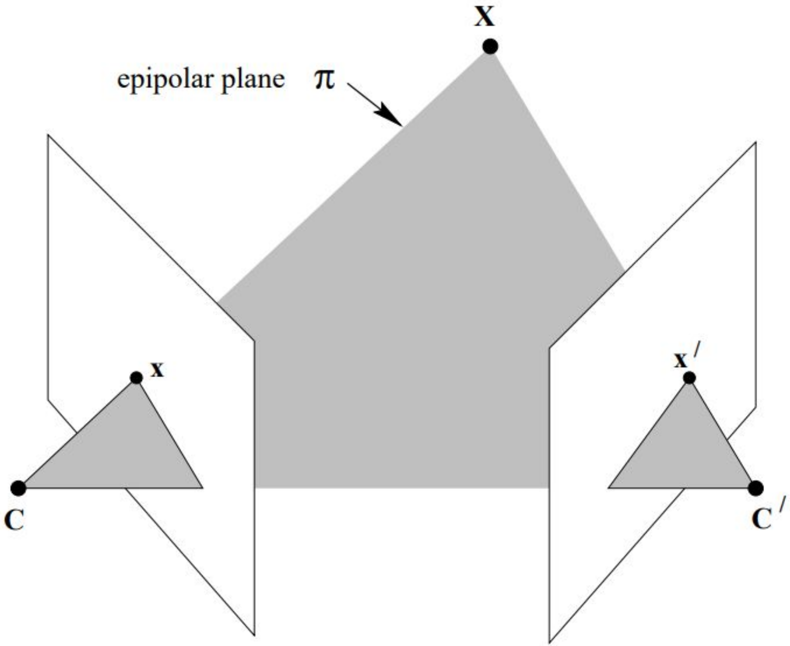
Мотивация

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



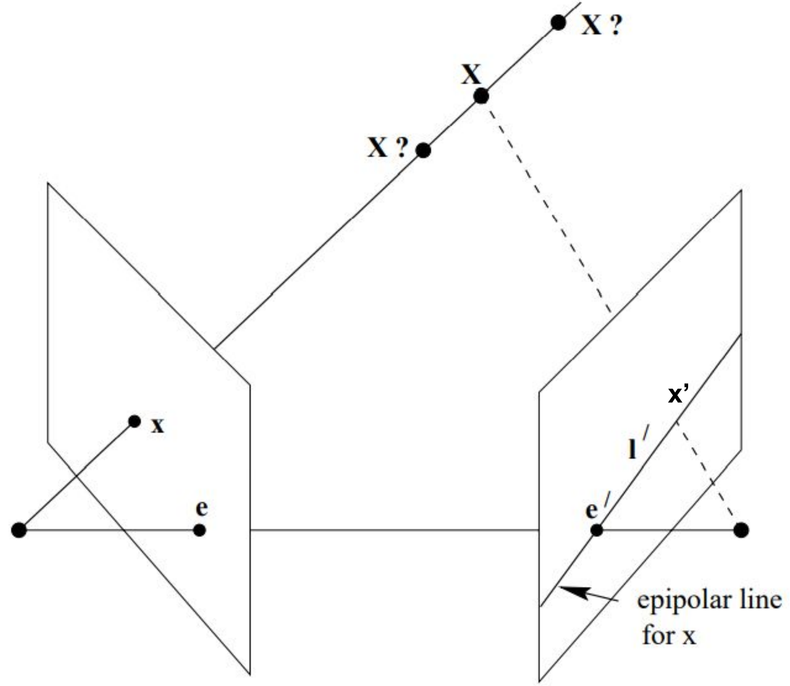
Эпиполярная геометрия

1) Каждая пара камер порождает пучок плоскостей, проходящих через ось **(C, C’)**



Эпиполярная геометрия

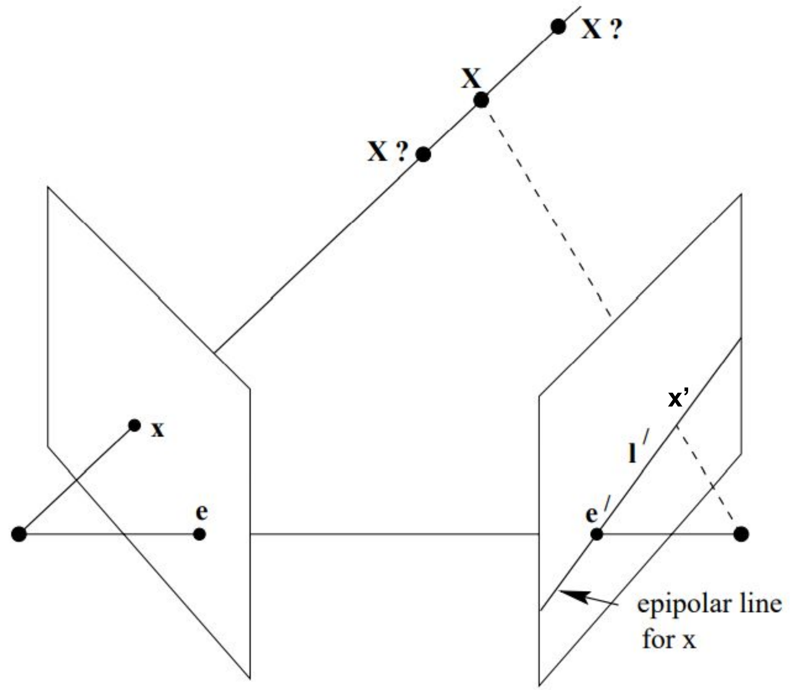
1) Каждая пара камер порождает пучок плоскостей, проходящих через ось **(C, C’)**
2) e,e’ - эпиполюсы
3) l’ - эпиполярная линия



Эпиполярная геометрия

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

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

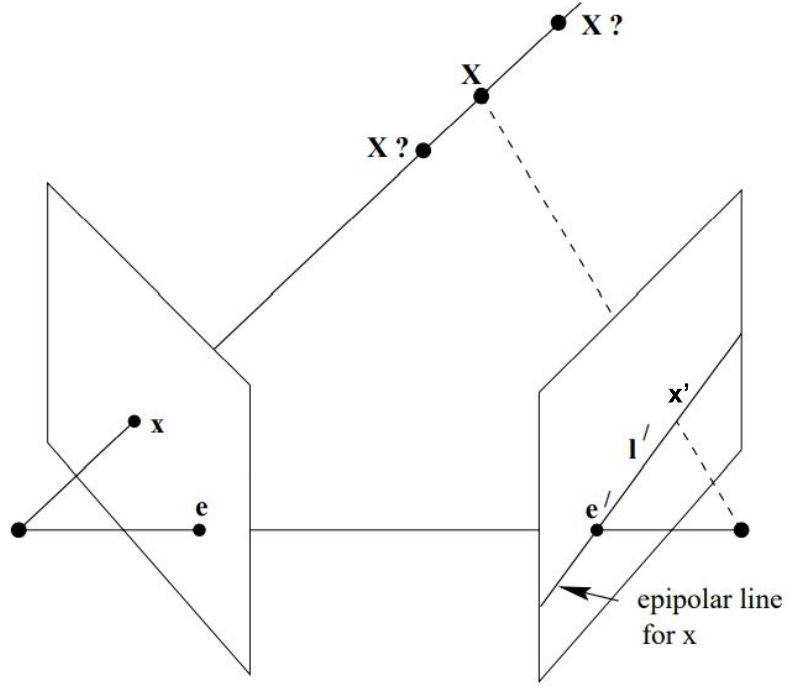


Эпиполярная геометрия

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

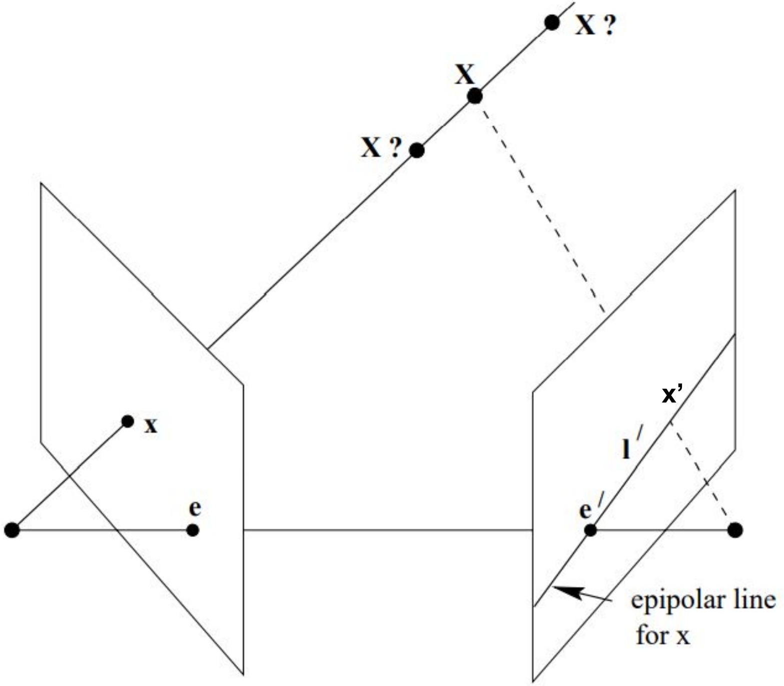
$$x \xrightarrow{?} l' \; ; \quad l' \cdot x' = 0$$

$$l' = Fx \; ; \quad x'^{T} F x = 0$$



Эпиполярная геометрия

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



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

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

фундаментальная матрица

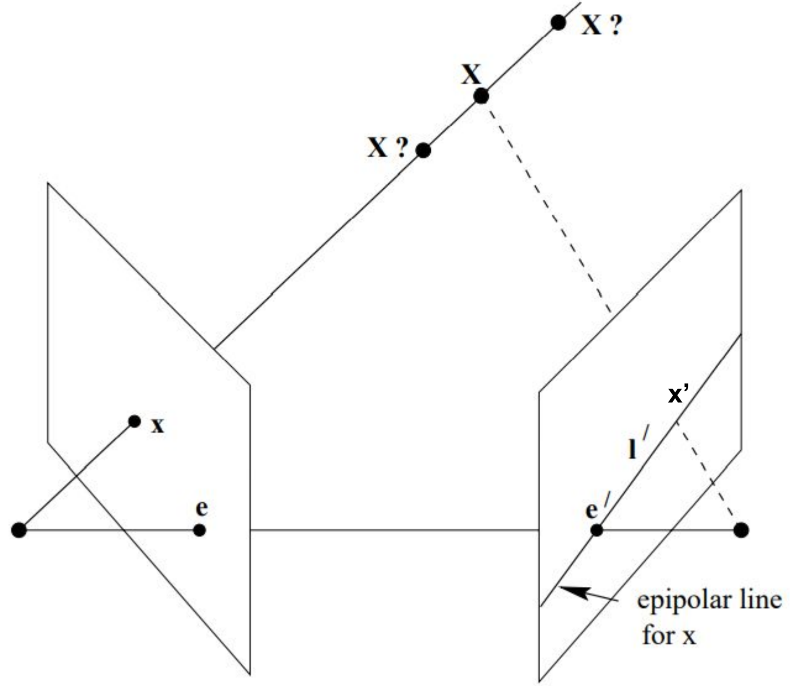
Эпиполярная геометрия

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

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

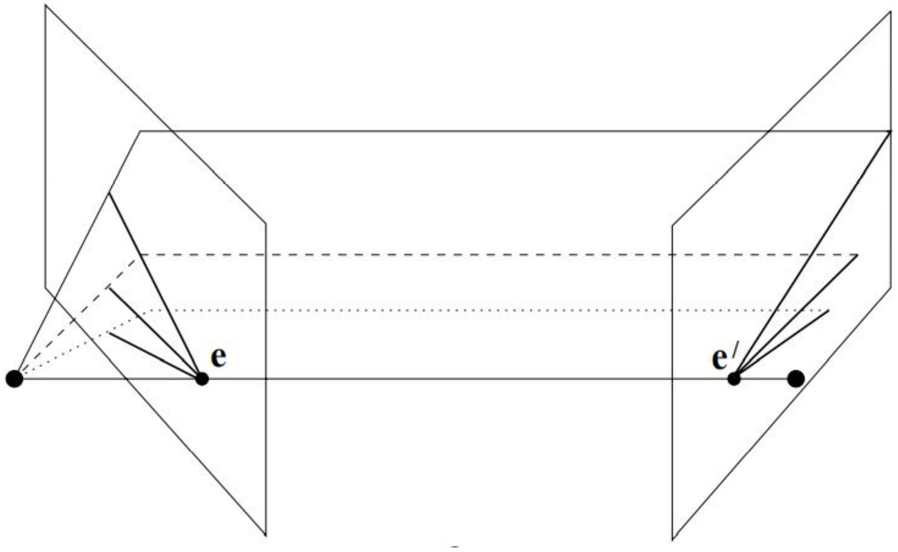
$$\ell' = Fx \; ; \; x'^T F x = 0$$

фундаментальная матрица



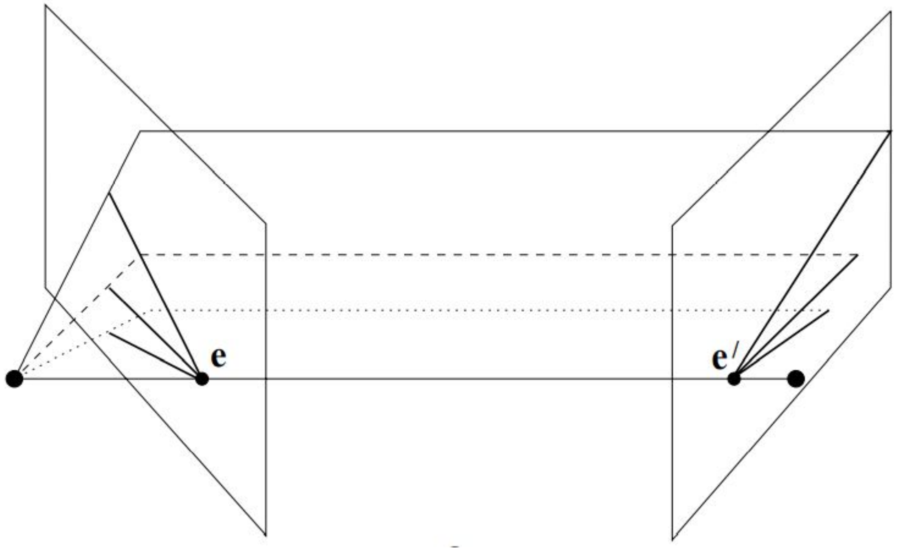
Фундаментальная матрица

1) Эпиполярные линии и эпиполюс



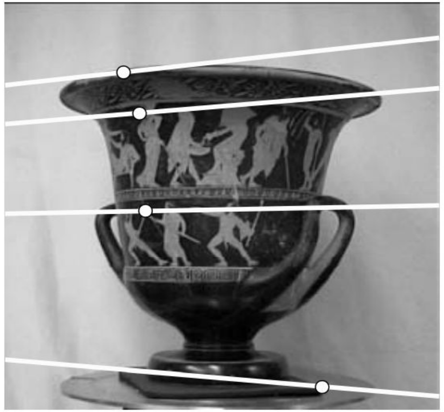
Фундаментальная матрица

1) Эпиполярные линии и эпиполюс
2) (пусть знаем **F**) Как найти эпиполюс по матчам и **F**?



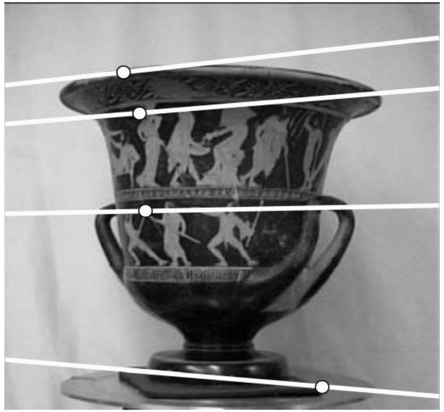
Фундаментальная матрица

1) Эпиполярные линии и эпиполюс
2) Как найти эпиполюс по матчам и **F**?
3) Что нам дает знание эпиполюса?



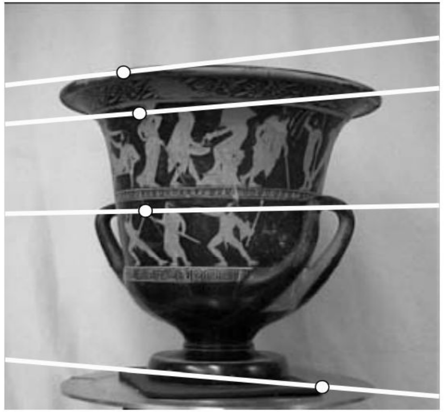
Фундаментальная матрица

1) Эпиполярные линии и эпиполюс
2) Как найти эпиполюс по матчам и **F**?
3) Что нам дает знание эпиполюса?
    - Можно определить луч, на котором лежит центр второй камеры



Фундаментальная матрица

1) Эпиполярные линии и эпиполюс
2) Как найти эпиполюс по матчам и **F**?
3) Что нам дает знание эпиполюса?
    - Можно определить луч, на котором лежит центр второй камеры
4) Чему соответствует ситуация если линии не пересекаются?



Фундаментальная матрица

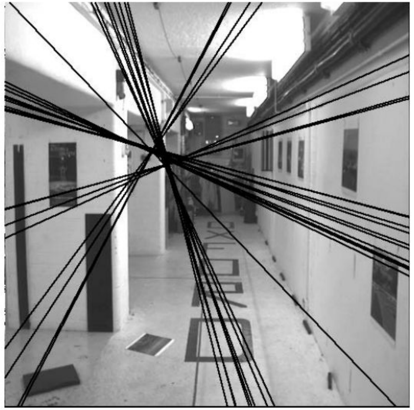
1) Singularity constraint:
   rank(F) = 2

Фундаментальная матрица

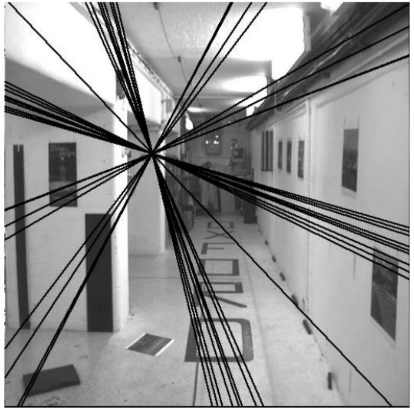
1) Singularity constraint:
   rank(F) = 2
2) Иначе эпиполярные линии не будут пересекаться в эпиполюсе

Фундаментальная матрица

1) Singularity constraint:
   rank(F) = 2
2) Иначе эпиполярные линии не будут пересекаться в эпиполюсе



non-singular



singular

$x_i'^T F x_i = 0 \ \forall i$

интуиция: $e'^T F x_i = 0 \ \forall i \Rightarrow e'^T F = 0 \Rightarrow$

$\Rightarrow F$ имеет нетрив. ядро

$\Rightarrow \mathrm{rank}(F) < 3$

Фундаментальная матрица

1) **F** можно разложить на матрицы камер (**P, P’**) $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]\ \textit{and}\ \mathtt{P}' = [[\mathbf{e}']_\times \mathtt{F} \mid \mathbf{e}']$

Фундаментальная матрица

1) **F** можно разложить на матрицы камер (**P, P’**) $\mathrm{P} = [\mathrm{I} \mid 0]$ *and* $\mathrm{P}' = [[\mathrm{e}']_{\times}\mathrm{F} \mid \mathrm{e}']$
2) То есть определить положение камер относительно друг друга!

$$P = K\underbrace{\overbrace{(R|t)}^{4}}\}3$$

внутр. парам. камеры

внешние парам. камеры

$$\underset{= \begin{pmatrix}x\\y\\z\\1\end{pmatrix}}{PX} = KR\begin{pmatrix}x\\y\\z\end{pmatrix} + Kt$$

ортог. (вращение)

верхнетреугольная: $\begin{pmatrix} f_x & & c_x \\ & f_y & c_y \\ & & 1 \end{pmatrix}$

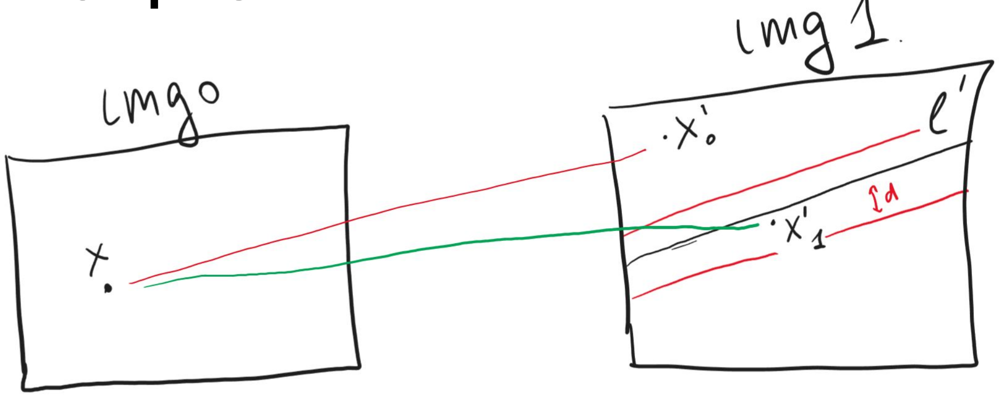
зная KR, можем извлечь K и R с помощью RQ-разлож.

Фильтрация матчей с помощью F

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

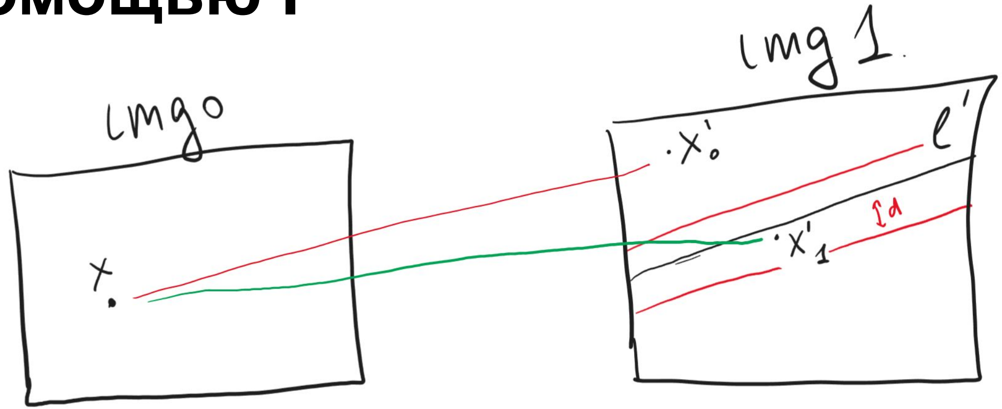
Фильтрация матчей с помощью F

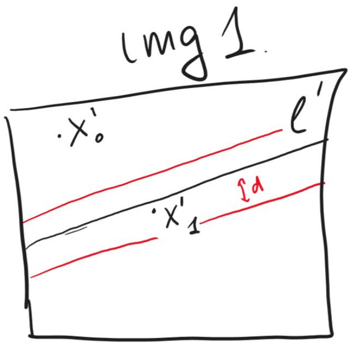
1) Если знаем **F**, можем использовать для фильтрации сопоставлений
2) +: используем геометрию
3) -: если сильные искажения, отсеет слишком много матчей

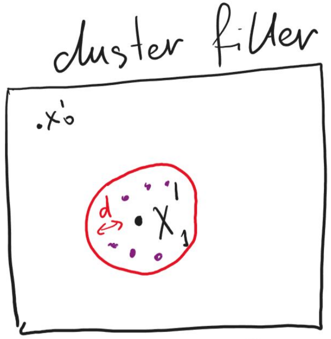


Фильтрация матчей с помощью F

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







$$dist(l, p) = \frac{(l, p)}{\left\| \begin{smallmatrix} l_1 \\ l_2 \end{smallmatrix} \right\|} = \frac{ax + by + c}{\sqrt{a^2 + b^2}}$$

$P_{false.pos} \sim img\text{-}size \cdot d$

$P_{false.pos} \sim d^2$

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

$$\mathbf{x}'^{\mathsf{T}} \mathrm{F} \mathbf{x} = 0 \tag{11.1}$$

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 F. In particular, writing $\mathbf{x} = (x, y, 1)^{\mathsf{T}}$ and $\mathbf{x}' = (x', y', 1)^{\mathsf{T}}$ each point match gives rise to one linear equation in the unknown entries of 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

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

Denote by $\mathbf{f}$ the 9-vector made up of the entries of 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

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

279

Hartley, Zisserman, глава 11

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

$$\mathbf{x}'^{\mathsf{T}} \mathrm{F} \mathbf{x} = 0 \tag{11.1}$$

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 F. In particular, writing $\mathbf{x} = (x, y, 1)^{\mathsf{T}}$ and $\mathbf{x}' = (x', y', 1)^{\mathsf{T}}$ each point match gives rise to one linear equation in the unknown entries of 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

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

Denote by $\mathbf{f}$ the 9-vector made up of the entries of 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

$$\mathrm{A}\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}. \tag{11.3}$$

279

Hartley, Zisserman, глава 11

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

$$\mathbf{x}'^{\mathsf{T}}\mathrm{F}\mathbf{x} = 0 \qquad (11.1)$$

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 F. In particular, writing $\mathbf{x} = (x, y, 1)^{\mathsf{T}}$ and $\mathbf{x}' = (x', y', 1)^{\mathsf{T}}$ each point match gives rise to one linear equation in the unknown entries of 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

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

Denote by $\mathbf{f}$ the 9-vector made up of the entries of 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

$$\mathrm{A}\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}. \qquad (11.3)$$

279

Hartley, Zisserman, глава 11

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

$$\mathbf{x}'^{\mathsf{T}} \mathtt{F} \mathbf{x} = 0 \tag{11.1}$$

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 F. In particular, writing $\mathbf{x} = (x, y, 1)^{\mathsf{T}}$ and $\mathbf{x}' = (x', y', 1)^{\mathsf{T}}$ each point match gives rise to one linear equation in the unknown entries of 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

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0. \tag{11.2}$$

Denote by $\mathbf{f}$ the 9-vector made up of the entries of 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

$$\mathtt{A}\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}. \tag{11.3}$$

279

Hartley, Zisserman, глава 11

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

$$\mathbf{x}'^{\mathsf{T}} \mathrm{F} \mathbf{x} = 0 \tag{11.1}$$

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 F. In particular, writing $\mathbf{x} = (x, y, 1)^{\mathsf{T}}$ and $\mathbf{x}' = (x', y', 1)^{\mathsf{T}}$ each point match gives rise to one linear equation in the unknown entries of 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

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0. \tag{11.2}$$

Denote by $\mathbf{f}$ the 9-vector made up of the entries of 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

$$\mathtt{A}\mathbf{f} = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} \mathbf{f} = \mathbf{0}. \tag{11.3}$$

279

Hartley, Zisserman, глава 11

Фундаментальная матрица

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

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

Фундаментальная матрица

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

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

Фундаментальная матрица

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

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

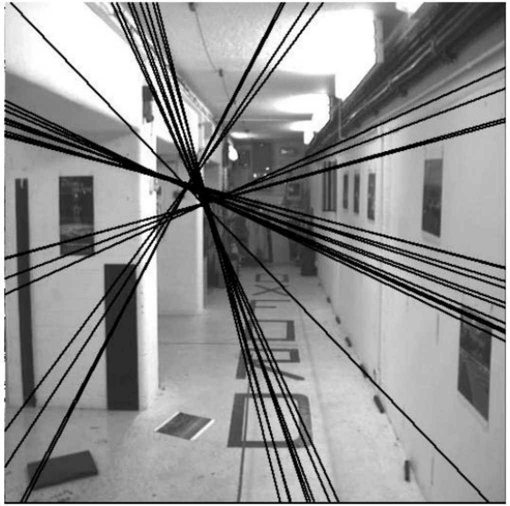
$$\{x\} \to \begin{matrix}\text{centroid} \\ \text{dispersion}\end{matrix} \to TN : x \to \hat{x} : \begin{matrix}\text{centroid}(\hat{x}) = 0 \\ \text{dispersion}(\hat{x}) = \sqrt{2}\end{matrix}$$

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

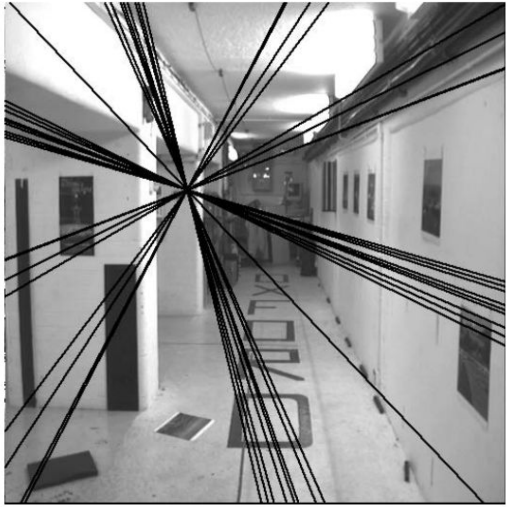
$$\iff x'^T \underbrace{(TN'^T \hat{F}\, TN)}_{F} x = 0$$

Фундаментальная матрица

1) Singularity constraint: Что делать если решили систему, но **F** получилась не сингулярная? (хотим rank(F) = 2)
2) **SVD**!



F



F’

$$F = UDV^T = U\begin{pmatrix} r & & \\ & s & \\ & & t \end{pmatrix}V^T \to U\begin{pmatrix} r & & \\ & s & \\ & & 0 \end{pmatrix}V^T = F'$$

$r \geq s \geq t$

Фундаментальная матрица

1) **F** можно разложить на матрицы камер (**P, P’**) $\mathrm{P} = [\mathrm{I} \mid 0]$ *and* $\mathrm{P}' = [[\mathrm{e}']_{\times}\mathrm{F} \mid \mathrm{e}']$
2) То есть определить положение камер относительно друг друга!

$$P = K\,\underbrace{\overbrace{(R \mid t)}^{4}}_{}\Big\}{\scriptstyle 3}$$

внутр. парам. камеры

внешние парам. камеры

$$P\,X = KR\begin{pmatrix} x \\ y \\ z \end{pmatrix} + Kt, \qquad X = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

ортог. (вращение)

верхнетреугольная: $\begin{pmatrix} f_x & & c_x \\ & f_y & c_y \\ & & 1 \end{pmatrix}$

зная KR, можем извлечь K и R с помощью RQ-разлож.

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 в RANSAC может победить **F**, соответствующая неверной калибровке (а есть еще дисторсии..)
4) Из-за того, что матрица F включает в себя калибровку, разложение получается с точностью до проективного преобразования (Hartley, Zisserman, теорема 9.10)

$$\mathrm{P} = [\mathrm{I} \mid 0] \; and \; \mathrm{P}' = [[\mathrm{e}']_\times \mathrm{F} \mid \mathrm{e}']$$

$$P = K \underbrace{\overbrace{(R \mid t)}^{4}}_{} \}_{3}$$

внутр. парам. камеры (K)

внешние парам. камеры ($(R \mid t)$)

$$P \underset{\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}}{X} = KR \begin{pmatrix} x \\ y \\ z \end{pmatrix} + Kt$$

R: ортог. (вращение)

K: верхнетреугольная: $\begin{pmatrix} f_x & & c_x \\ & f_y & c_y \\ & & 1 \end{pmatrix}$

зная KR, можем извлечь K и R с помощью RQ-разлож.

Фундаментальная матрица

Из-за того, что матрица F включает в себя калибровку, разложение получается с точностью до проективного преобразования (Hartley, Zisserman, теорема 9.10)

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

Более того, делая разложение матрицы камеры **P**, получаем в итоге и калибровку **K**. А что если откуда-то уже знаем внутренние параметры камеры (нередкий случай), и они не совпадают с найденными?

10.2 Reconstruction ambiguity 265

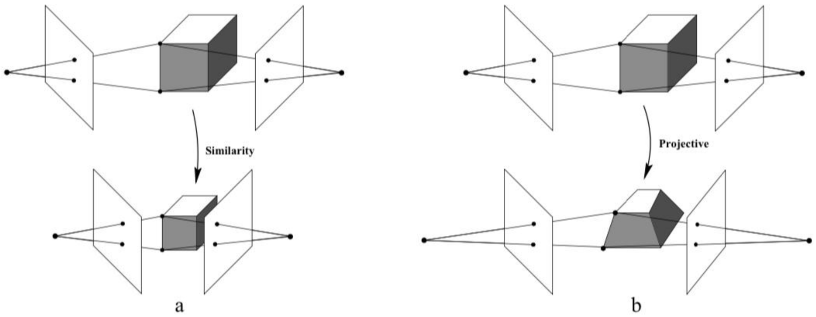


Fig. 10.2. **Reconstruction ambiguity.** *(a) If the cameras are calibrated then any reconstruction must respect the angle between rays measured in the image. A similarity transformation of the structure and camera positions does not change the measured angle. The angle between rays and the baseline (epipoles) is also unchanged. (b) If the cameras are uncalibrated then reconstructions must only respect the image points (the intersection of the rays with the image plane). A projective transformation of the structure and camera positions does not change the measured points, although the angle between rays is altered. The epipoles are also unchanged (intersection with baseline).*

пример проективной неоднозначности

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!**

$$\tilde{x}'^{T} E \tilde{x} = 0$$

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

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

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

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

$$E = K^{T} F K$$

Существенная (Essential) матрица

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

$$\tilde{x}'^T E \tilde{x} = 0$$

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

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

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

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

$$E = K^T F K$$

Существенная (Essential) матрица

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

$$\tilde{x}'^{T} E \tilde{x} = 0$$

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

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

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

$$x'^{T} \underbrace{\left(K^{-1T} E K^{-1}\right)}_{F} x = 0$$

$$E = K^{T} F K$$

Разложение существенной матрицы

1) Выберем систему координат так, чтобы первая камера имела тривиальную матрицу **P**
2) Тогда нужно найти только относительный сдвиг и поворот

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

$$\mathtt{P}' = [\mathtt{UWV}^\mathsf{T} \mid +\mathbf{u}_3] \;\text{ or }\; [\mathtt{UWV}^\mathsf{T} \mid -\mathbf{u}_3] \;\text{ or }\; [\mathtt{UW}^\mathsf{T}\mathtt{V}^\mathsf{T} \mid +\mathbf{u}_3] \;\text{ or }\; [\mathtt{UW}^\mathsf{T}\mathtt{V}^\mathsf{T} \mid -\mathbf{u}_3].$$

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

Разложение существенной матрицы

$\exists\ P = K(I|0)$, $P' = K'(R|t)$

Утв 1: $F \equiv K^{-T}[t]_\times R K^{-1}$, где $[t]_\times$ – матрица векторн. умножения на t

(без док-ва)

NB: $\vec{a} \times \vec{b} = -\vec{b} \times \vec{a}$

$[a]_\times b = -b^T [a]_\times \Longleftrightarrow [a]_\times^T = -[a]_\times$

С другой стороны, $F = K^{-T} E K \Longrightarrow$

$\Longrightarrow E = [t]_\times R \equiv S R$

антисимм. вращение

Разложение существенной матрицы

Утв ② :
(без док-ва)

$$E \overset{SVD}{=} U \begin{pmatrix} 1 & & \\ & 1 & \\ & & 0 \end{pmatrix} V^T$$

Утв ③ :
(без док-ва)

$$\forall S: S^T = -S \quad , \; S = kUZU^T, \; Z = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$Z^T = -Z$$

$$\mathrm{rank}(Z) = 2$$

Разложение существенной матрицы

Утв.: $\exists\ E = SR = U\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T \Longleftrightarrow$

$\Longleftrightarrow S = UZU^T,\ R = UWV^T \ /\ R = UW^TV^T$

$W = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Proof: $\Leftarrow$: $ZW = -ZW^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

$$\underset{Z}{\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}} \underset{W}{\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}} = \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$$

Разложение существенной матрицы

$\Rightarrow$: (нет других разложений)

Знаем, что • $E = U\begin{pmatrix}1 & & \\ & 1 & \\ & & 0\end{pmatrix}V^T$, $\operatorname{rank}(E)=2$

$\operatorname{rank}(V^T)=3$

• $E = SR$

утв. 3

$$\Rightarrow \quad E = \underbrace{[UZU^T]}_{S}\underbrace{[U X V^T]}_{R.} =$$

какая-то вращ.

$$= UZXV^T$$

$$ZX = \begin{pmatrix}1 & & \\ & 1 & \\ & & 0\end{pmatrix} \Rightarrow X = W / W^T$$

т.к. X - вращ., то не подходит

$$X = \begin{pmatrix}0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0\end{pmatrix}$$

(знак и масштаб нигде не важен) $\blacksquare$

Разложение существенной матрицы

Утв: $t = \pm U_3$ (столбец)

Proof: $E = SR$,

$S = UZU^T = [t]_\times \Rightarrow St = 0$

$UZU^T t = 0 \Rightarrow t = U \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = U_3$

$\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

Итог: $\exists\ E,\ P = [I|0] \Rightarrow$

$\Rightarrow P' = [UWV^T | U_3]$ /

$[UWV^T | -U_3]$ /

$[UW^TV^T | U_3]$ /

$[UW^TV^T | -U_3]$

Ура! Извлекли положения камер из матрицы E

Разложение существенной матрицы

1) Матрица **P’** определяется с точностью до четырех вариантов
2) Выбрать правильный можно с помощью depth-теста: триангулированная точка должна быть спереди для обеих камер

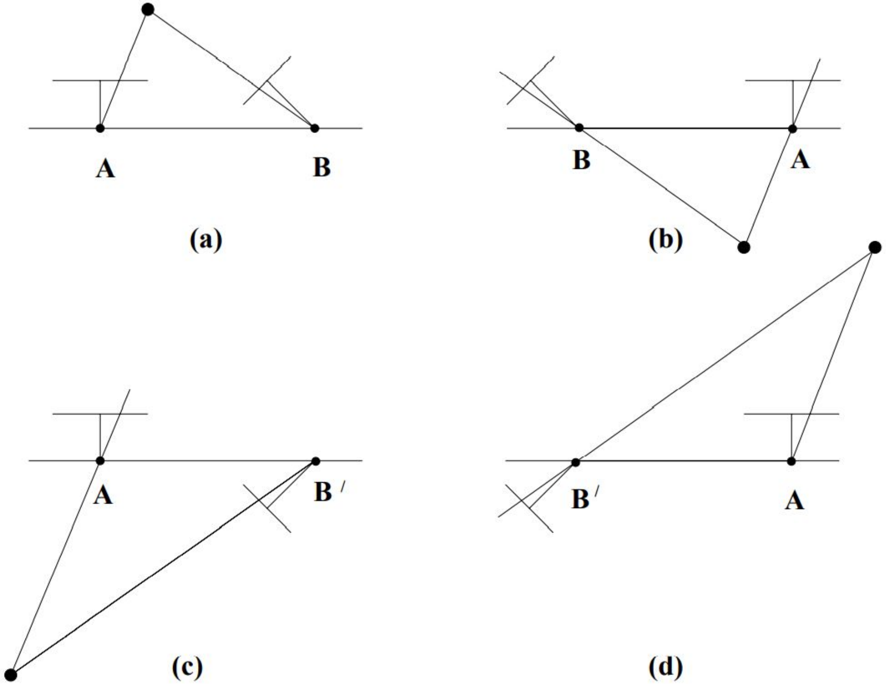


Fig. 9.12. **The four possible solutions for calibrated reconstruction from E.** *Between the left and right sides there is a baseline reversal. Between the top and bottom rows camera B rotates 180° about the baseline. Note, only in (a) is the reconstructed point in front of both cameras.*

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(*p*88). In each image we have a measurement $\mathbf{x} = \mathtt{P}\mathbf{X}$, $\mathbf{x}' = \mathtt{P}'\mathbf{X}$, and these equations can be combined into a form $\mathtt{A}\mathbf{X} = \mathbf{0}$, 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, $\boxed{\mathbf{x} \times (\mathtt{P}\mathbf{X}) = \mathbf{0}}$ and writing this out gives

$$\begin{aligned} x(\mathbf{p}^{3\mathsf{T}}\mathbf{X}) - (\mathbf{p}^{1\mathsf{T}}\mathbf{X}) &= 0 \\ y(\mathbf{p}^{3\mathsf{T}}\mathbf{X}) - (\mathbf{p}^{2\mathsf{T}}\mathbf{X}) &= 0 \\ x(\mathbf{p}^{2\mathsf{T}}\mathbf{X}) - y(\mathbf{p}^{1\mathsf{T}}\mathbf{X}) &= 0 \end{aligned}$$

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

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

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

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 $\mathtt{A}\mathbf{X} = \mathbf{0}$ were discussed in section 4.1(*p*88) 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(*p*88). In each image we have a measurement $\mathbf{x} = \mathtt{P}\mathbf{X}$, $\mathbf{x}' = \mathtt{P}'\mathbf{X}$, and these equations can be combined into a form $\mathtt{A}\mathbf{X} = \mathbf{0}$, 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, $\boxed{\mathbf{x} \times (\mathtt{P}\mathbf{X}) = \mathbf{0}}$ and writing this out gives

$$\begin{aligned} x(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{1\top}\mathbf{X}) &= 0 \\ y(\mathbf{p}^{3\top}\mathbf{X}) - (\mathbf{p}^{2\top}\mathbf{X}) &= 0 \\ x(\mathbf{p}^{2\top}\mathbf{X}) - y(\mathbf{p}^{1\top}\mathbf{X}) &= 0 \end{aligned}$$

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

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

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

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 $\mathtt{A}\mathbf{X} = \mathbf{0}$ were discussed in section 4.1(*p*88) 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(*p*88). In each image we have a measurement $\mathbf{x} = \mathtt{P}\mathbf{X}$, $\mathbf{x}' = \mathtt{P}'\mathbf{X}$, and these equations can be combined into a form $\mathtt{A}\mathbf{X} = \mathbf{0}$, 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, $\boxed{\mathbf{x} \times (\mathtt{P}\mathbf{X}) = \mathbf{0}}$ and writing this out gives

$$\begin{aligned} x(\mathbf{p}^{3\mathsf{T}}\mathbf{X}) - (\mathbf{p}^{1\mathsf{T}}\mathbf{X}) &= 0 \\ y(\mathbf{p}^{3\mathsf{T}}\mathbf{X}) - (\mathbf{p}^{2\mathsf{T}}\mathbf{X}) &= 0 \\ x(\mathbf{p}^{2\mathsf{T}}\mathbf{X}) - y(\mathbf{p}^{1\mathsf{T}}\mathbf{X}) &= 0 \end{aligned}$$

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

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

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

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 $\mathtt{A}\mathbf{X} = \mathbf{0}$ were discussed in section 4.1(*p*88) 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! (система уравнений получается похожим образом как для поиска гомографии)

7.1 Basic equations

We assume a number of point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3D points $\mathbf{X}_i$ and 2D image points $\mathbf{x}_i$ are given. We are required to find a camera matrix P, namely a 3×4 matrix such that $\boxed{\mathbf{x}_i = \mathtt{P}\mathbf{X}_i}$ 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(*p*88) for each correspondence $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ we derive a relationship

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \\ -y_i\mathbf{X}_i^\mathsf{T} & x_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0}. \tag{7.1}$$

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

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0} \tag{7.2}$$

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 $\mathtt{A}\mathbf{p} = \mathbf{0}$, where $\mathbf{p}$ 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 $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3D points $\mathbf{X}_i$ and 2D image points $\mathbf{x}_i$ are given. We are required to find a camera matrix $\mathtt{P}$, namely a 3×4 matrix such that $\boxed{\mathbf{x}_i = \mathtt{P}\mathbf{X}_i}$ for all i. The similarity of this problem with that of computing a 2D projective transformation $\mathtt{H}$, treated in chapter 4, is evident. The only difference is the dimension of the problem. In the 2D case the matrix $\mathtt{H}$ has dimension 3×3, whereas in the present case, $\mathtt{P}$ 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(*p*88) for each correspondence $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ we derive a relationship

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \\ -y_i\mathbf{X}_i^\mathsf{T} & x_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0}. \tag{7.1}$$

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

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0} \tag{7.2}$$

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

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 $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3D points $\mathbf{X}_i$ and 2D image points $\mathbf{x}_i$ are given. We are required to find a camera matrix P, namely a 3×4 matrix such that $\boxed{\mathbf{x}_i = \mathtt{P}\mathbf{X}_i}$ 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(*p*88) for each correspondence $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ we derive a relationship

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \\ -y_i\mathbf{X}_i^\mathsf{T} & x_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0}. \tag{7.1}$$

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

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0} \tag{7.2}$$

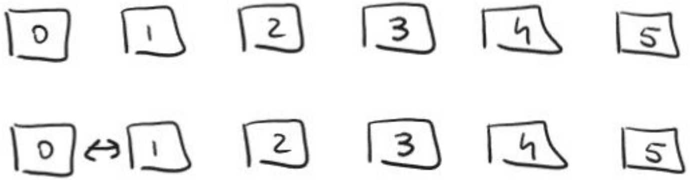
since the three equations of (7.1) are linearly dependent. From a set of n point correspondences, we obtain a $2n \times 12$ matrix A by stacking up the equations (7.2) for each correspondence. The projection matrix P is computed by solving the set of equations $\mathtt{A}\mathbf{p} = \mathbf{0}$, where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

Общий алгоритм SFM

1) На вход кадры и матчи

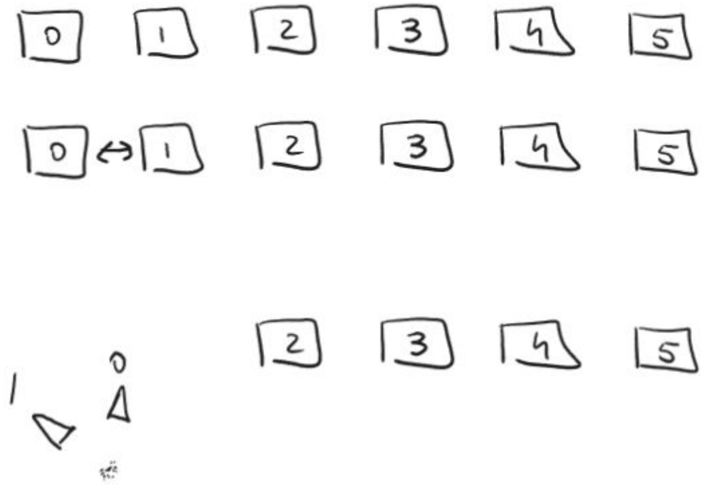
Общий алгоритм SFM

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



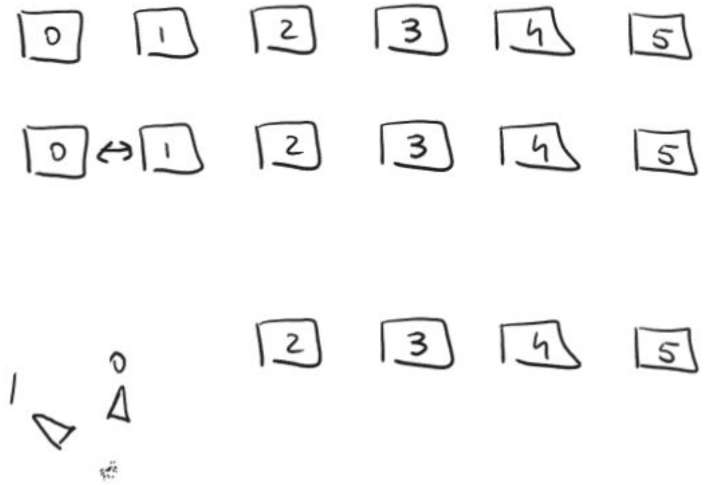
Общий алгоритм SFM

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



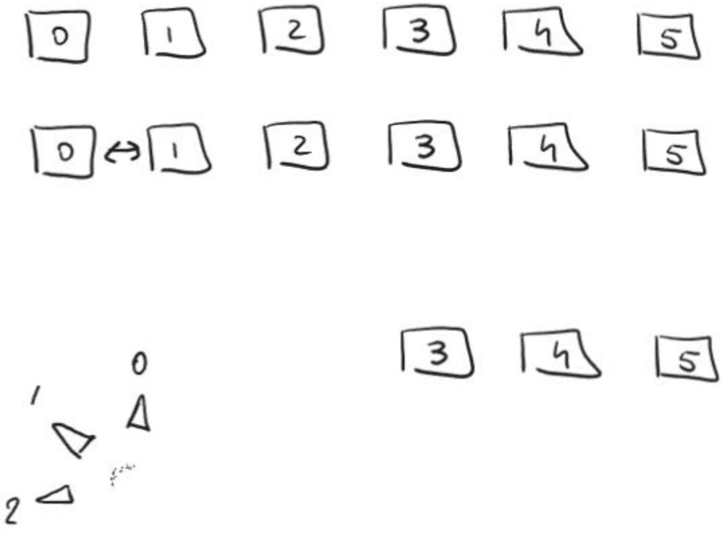
Общий алгоритм SFM

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



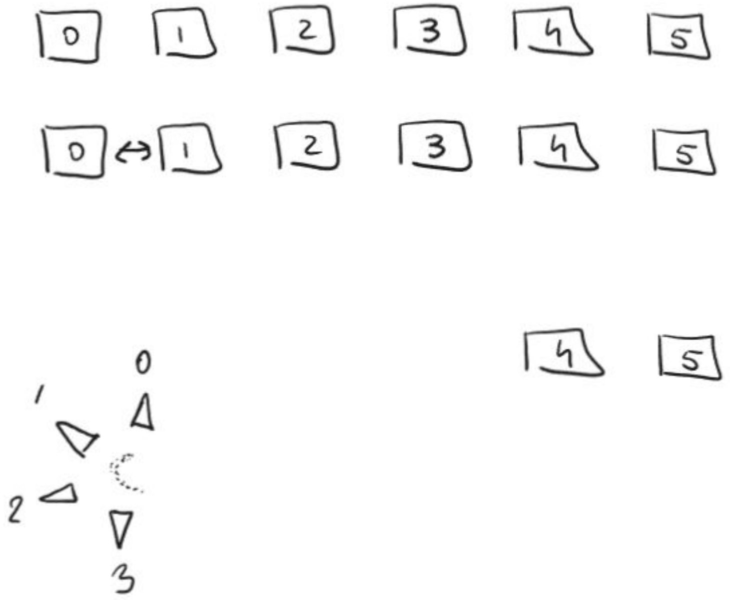
Общий алгоритм SFM

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



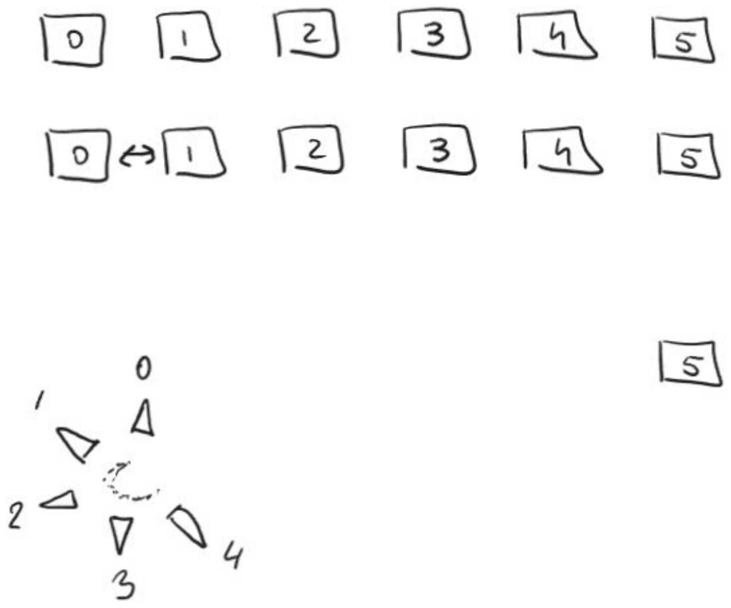
Общий алгоритм SFM

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



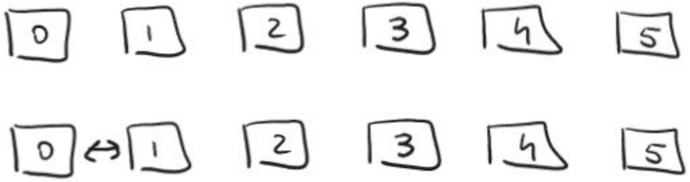
Общий алгоритм SFM

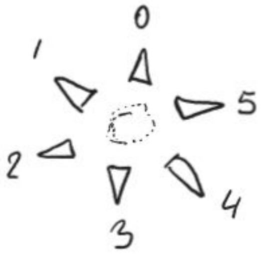
1) На вход кадры и матчи
2) Выбор первой пары (часто нетривиальный)
3) Выравнивание через essential матрицу, тривиальная калибровка
4) Триангуляция (добавление точек)
5) *Уточнение положения и калибровки (TBD)*
6) Резекция (добавление камер)
7) goto шаг 4



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
   a) 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

Вопросы?

Симиютин Борис
simiyutin.boris@yandex.ru