

*Курс «Трёхмерное компьютерное зрение»*

**Тема №3**  
**«Многовидовая геометрия»**

**Антон Конушин**



# Параллакс

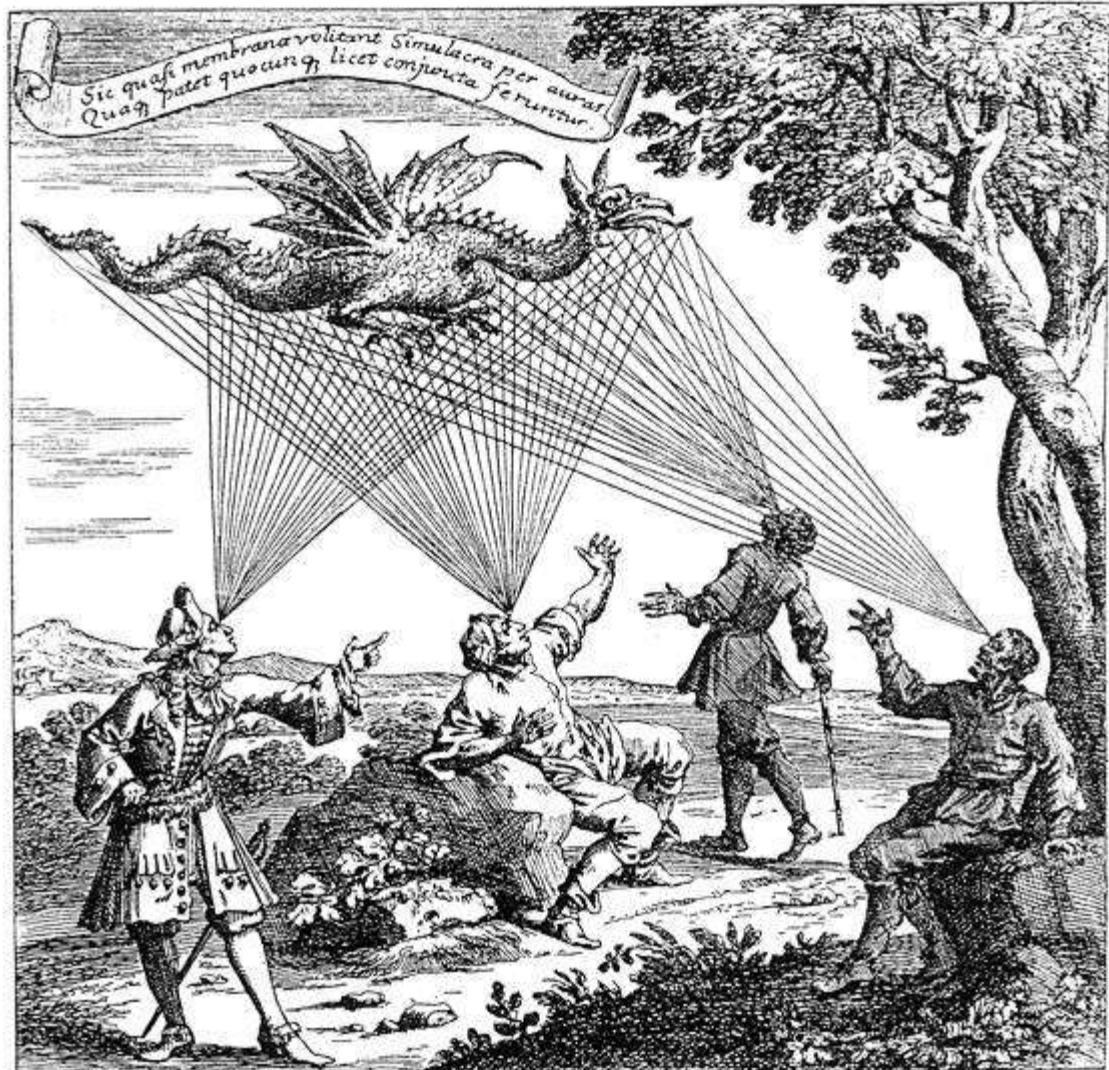
---



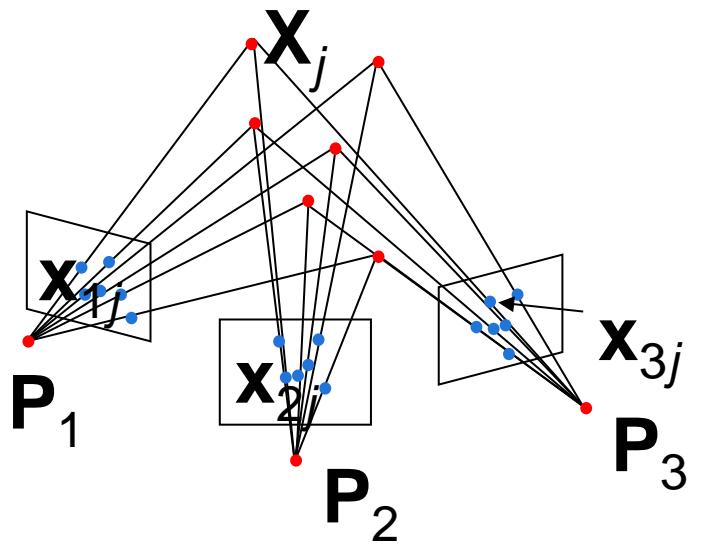
- Параллакс - видимое смещение объекта в зависимости от точки обзора
- Чем объект ближе, тем смещение больше



# Многовидовая геометрия



Драконъ, видимый подъ различными углами зрения  
По гравюре из энди изъ „Oculus artificialis teledioptricus“ Цанна. 1702 года.



Движение проекций 3D точек, вызванное движением камеры, позволяет определить координаты 3D точек (*структуру, structure*) и само движение камеры (*движение, motion*).

Это задача «структурь из движения» (*structure from motion*)

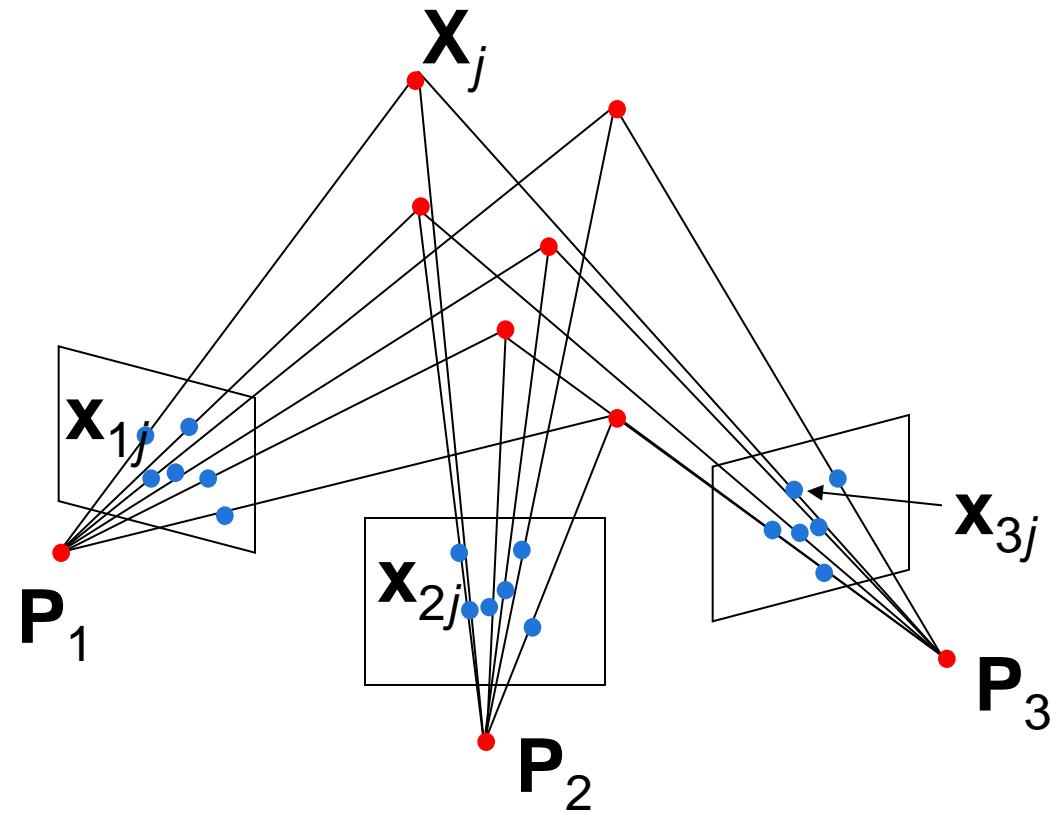
# Формулировка задачи структуры из движения



- Дано:  $m$  изображений  $n$  фиксированных 3D точек

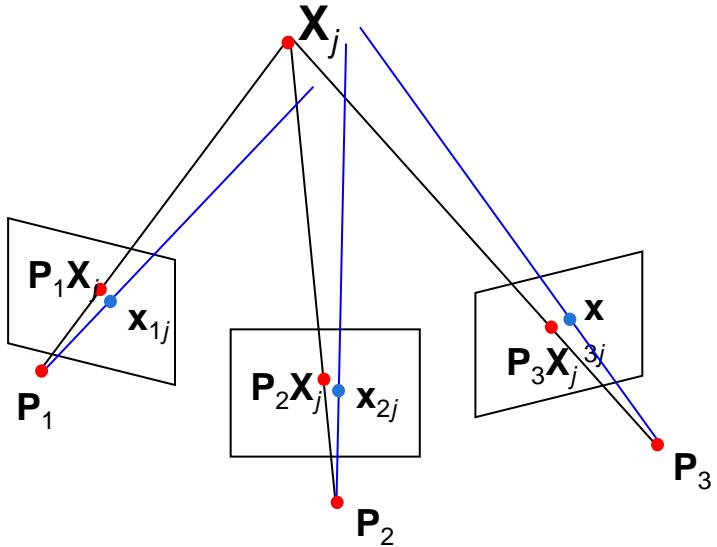
$$\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

- Задача: оценить  $m$  - матриц проекции  $\mathbf{P}_i$  и  $n$  - 3D точек  $\mathbf{X}_j$  из  $mn$  соответствий  $\mathbf{x}_{ij}$





# Целевая функция



Оптимальная целевая функция – сумма ошибок репроекций 3D точек  $\mathbf{X}$  на все изображения:

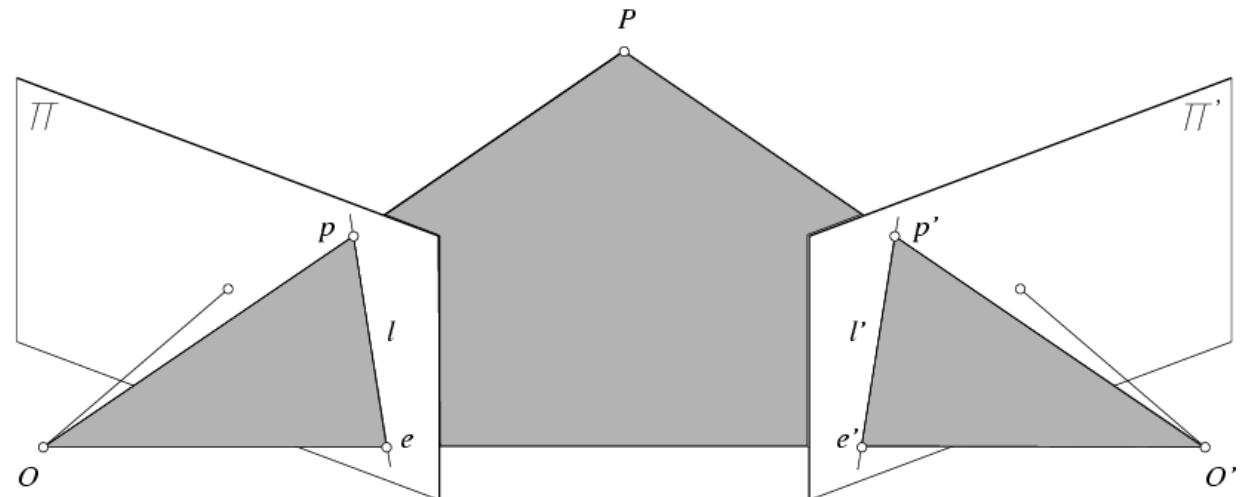
$$E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^m \sum_{j=1}^n D(\mathbf{x}_{ij}, \mathbf{P}_i \mathbf{X}_j)^2$$

- Нелинейная целевая функция с большим количеством параметров
  - $3n$  точек
  - $6m$  параметров для калиброванного случая,  $11m$  для некалиброванного
- Можно решить градиентным спуском, но плохо – масса локальных минимумов
- Поэтому нужно получить хорошее начальное приближение, в этом основная проблема
- Поэтому задачу декомпозириуем на более простые подзадачи

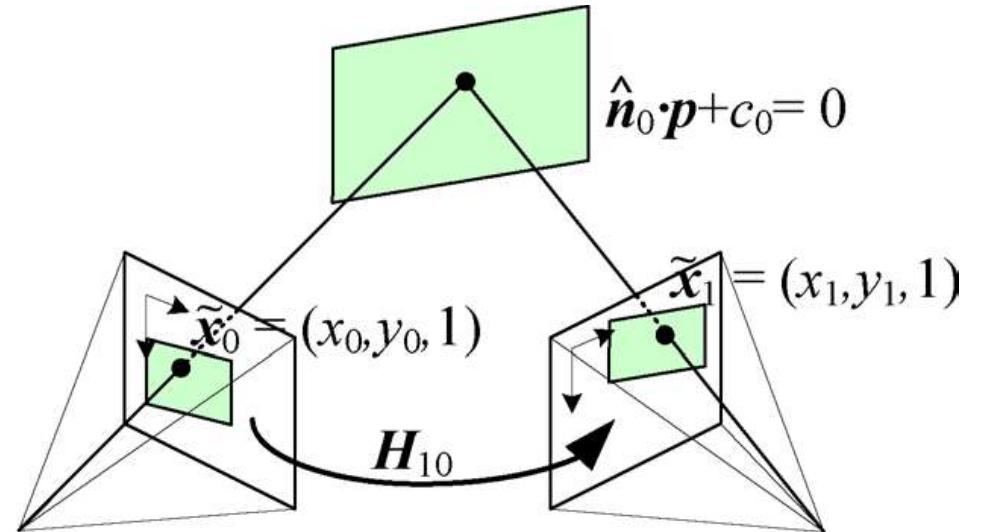


# Геометрия 2x камер

- Многовидовая геометрия позволяет установить взаимосвязи между точками разных изображений, параметрами камеры и 3D точками сцены
- Мы будем рассматривать случай 2x камер



Эпиполярная геометрия

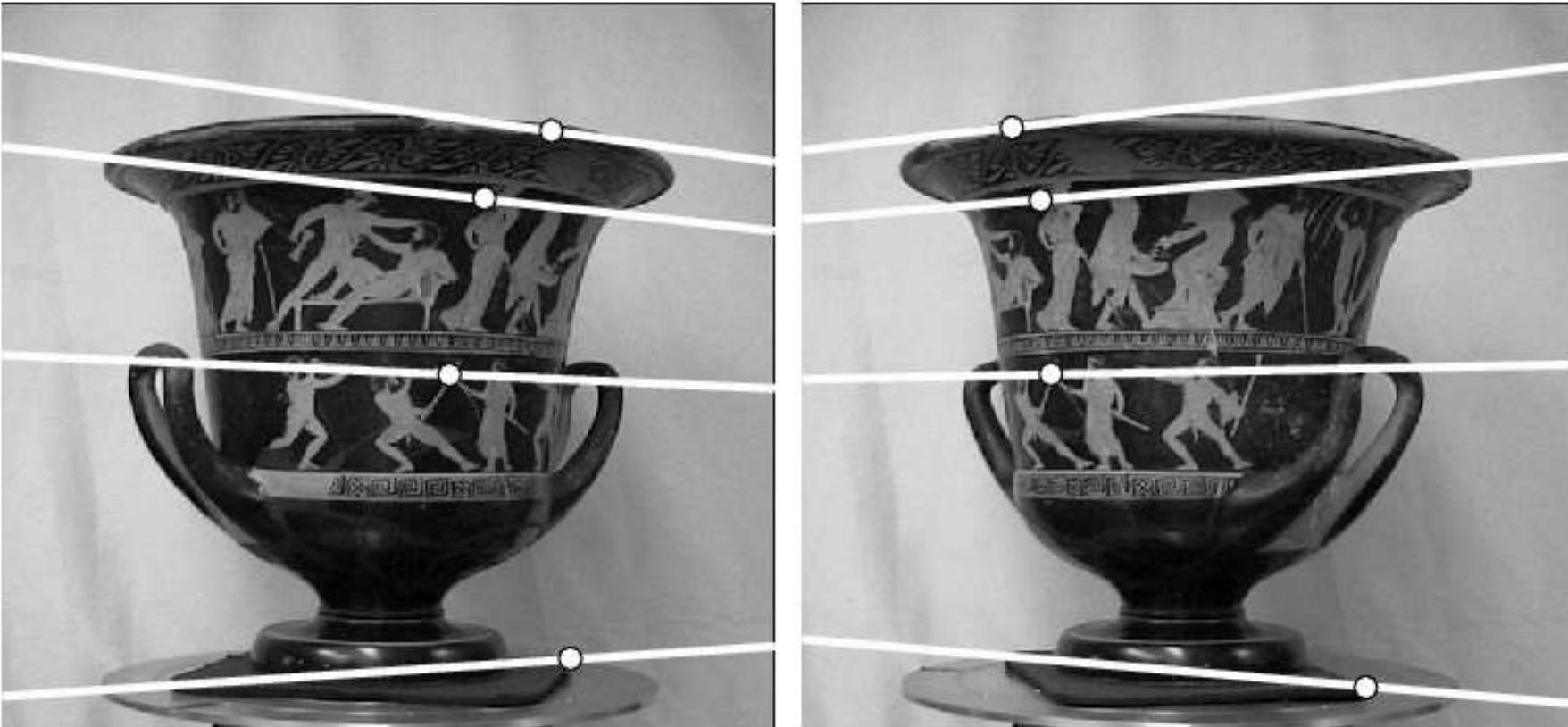


Гомография



# Эпиполярная геометрия

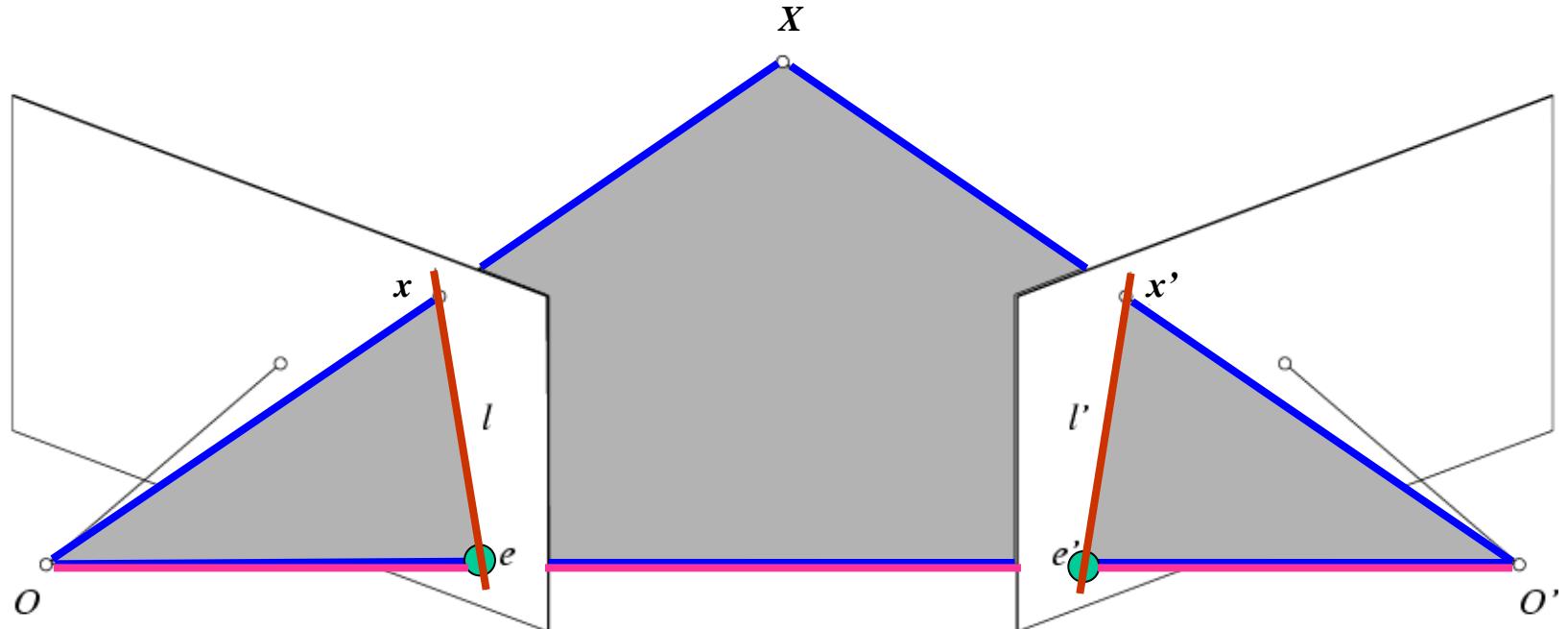
# Система из 2х камер



Рассмотрим случай, когда два снимка одной и той же сцены получены с разных ракурсов (разные центры проекций)



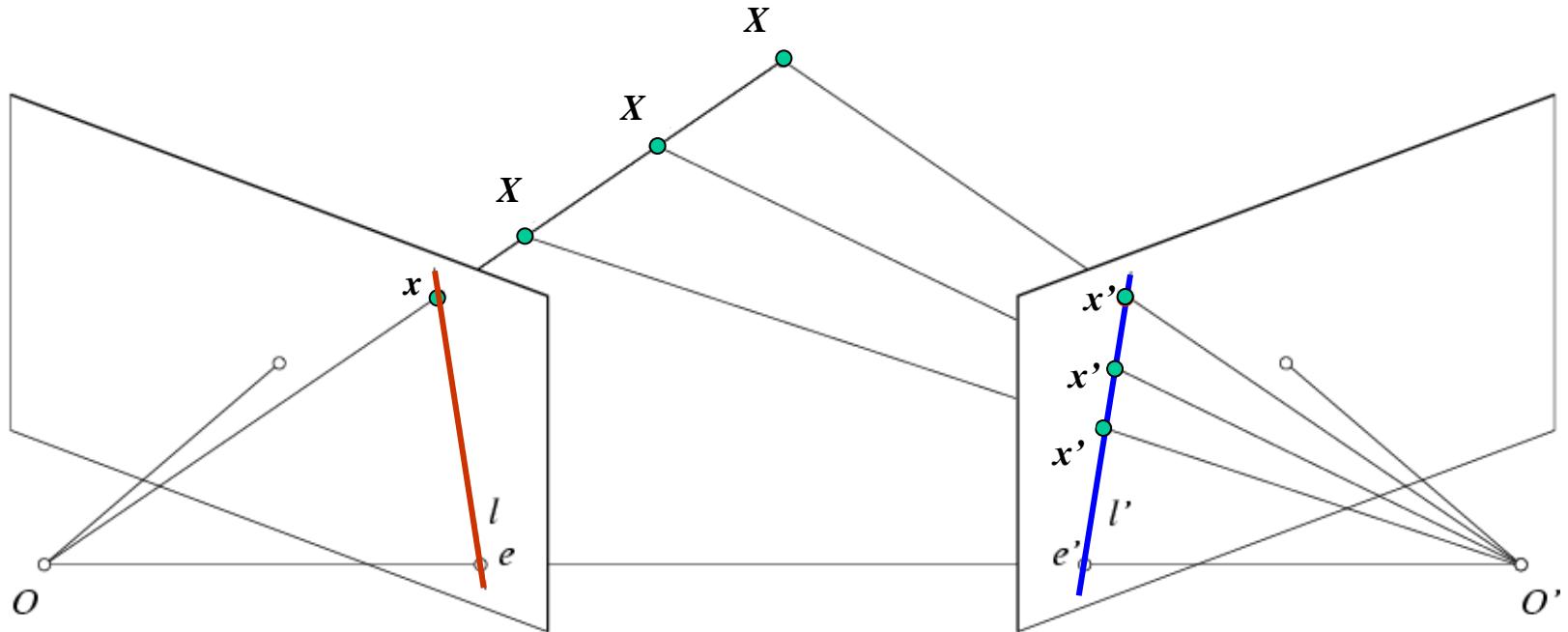
# Эпиполярная геометрия



- **Базовая линия (Baseline)** – линия, соединяющая центры камер
- **Эпиполярная плоскость (Epipolar Plane)** – плоскость, проходящая через базовую линию (пучок плоскостей))
- **Эпиполи/Эпиполяры (Epipoles)**
  - = пересечение базовой линии с картинными плоскостями
  - = проекции центра проекций второй камеры
- **Эпиполярные линии (Epipolar Lines)** – пересечение эпиполярной плоскости с картинной плоскостью (дают соответствующие пары)



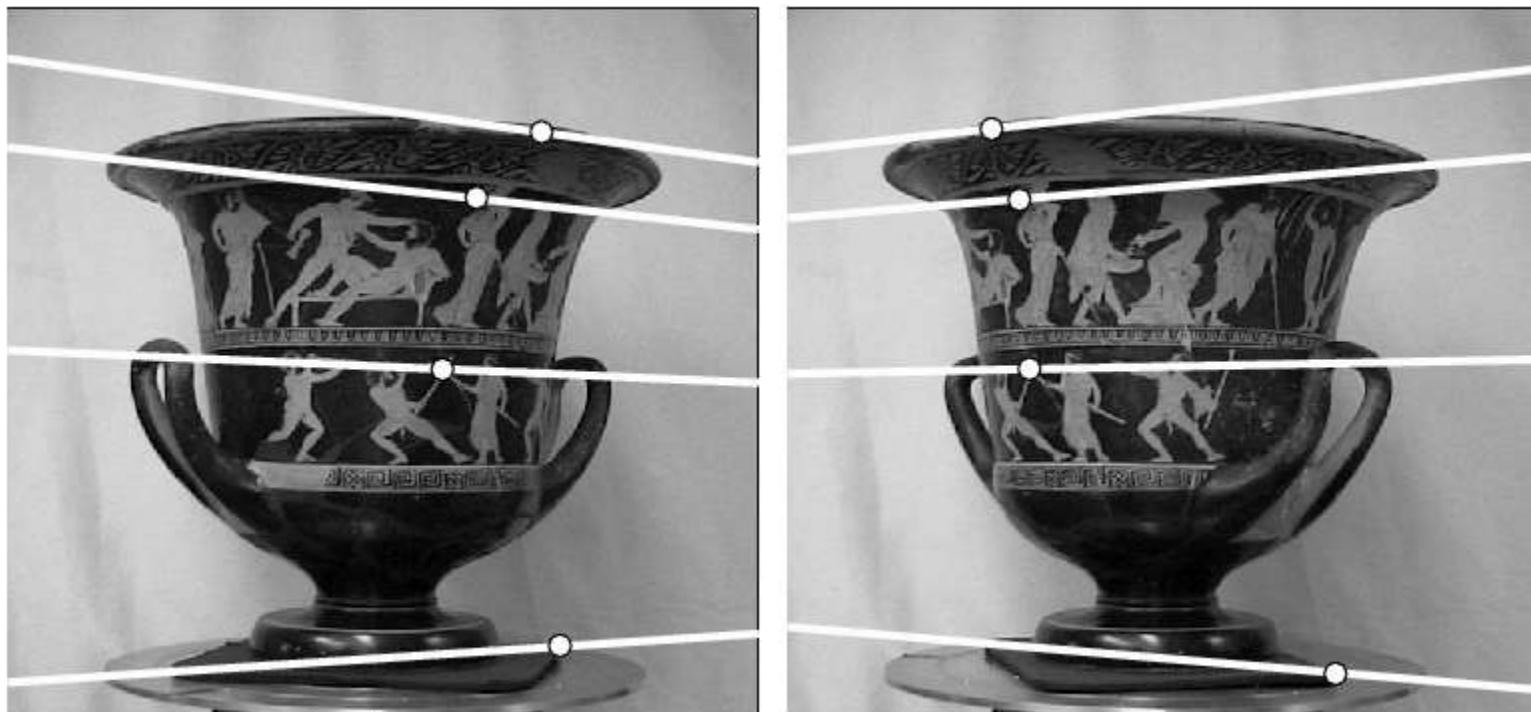
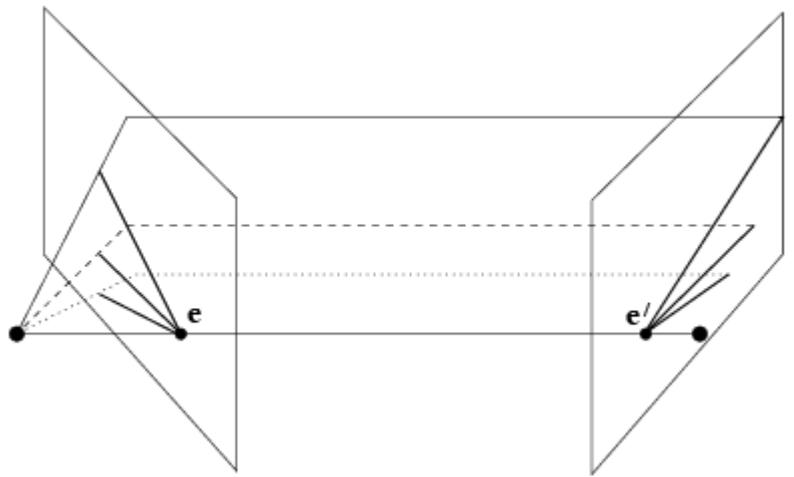
# Эпиполярное ограничение



- Пусть даны изображения  $I$  и  $I'$  с центрами проекции  $O$  и  $O'$ , и проекция  $x$  точки  $X$  (неизвестной) на изображение  $I$ .
- Что мы можем сказать про точку  $x'$  – проекцию  $X$  на  $I'$ ?
  - $x'$  должна лежать на эпиполярной линии  $l'$
- Эпиполярное ограничение:
  - Проекции  $x$  и  $x'$  точки сцены  $X$  на изображения  $I$  и  $I'$  должны лежать на соответствующих эпиполярных линиях  $l$  и  $l'$

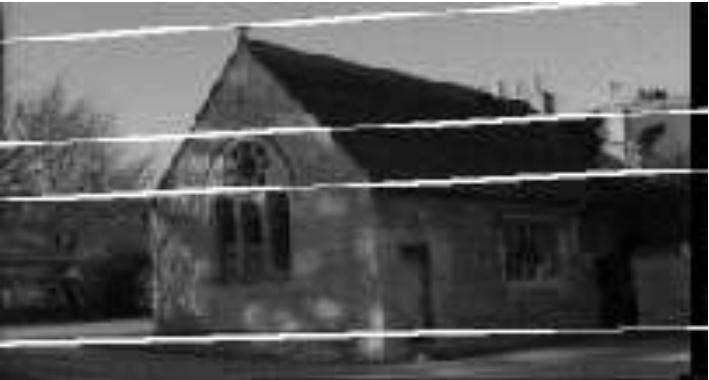
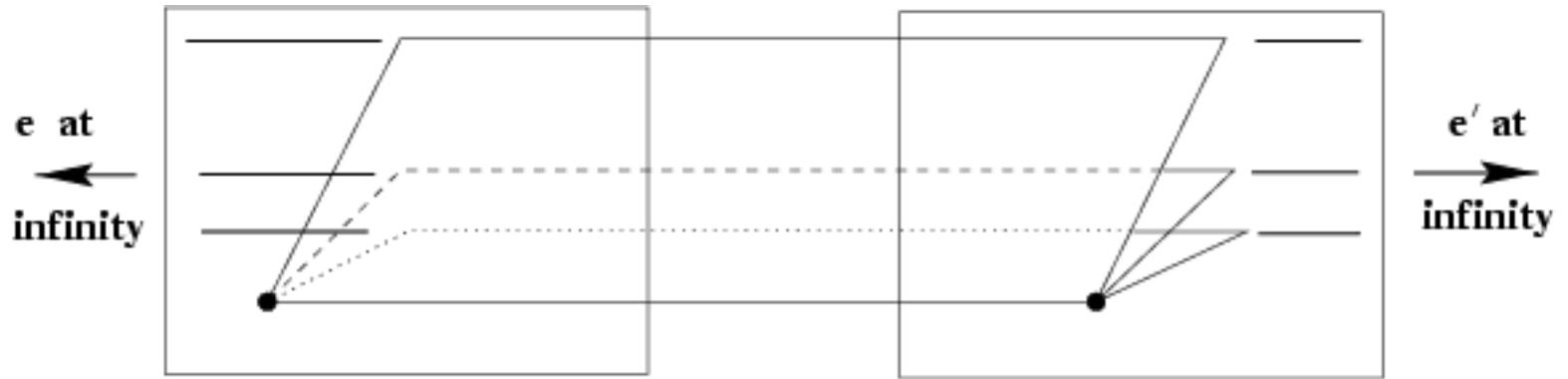


# Пример: сближающиеся камеры





# Пример: стереопара

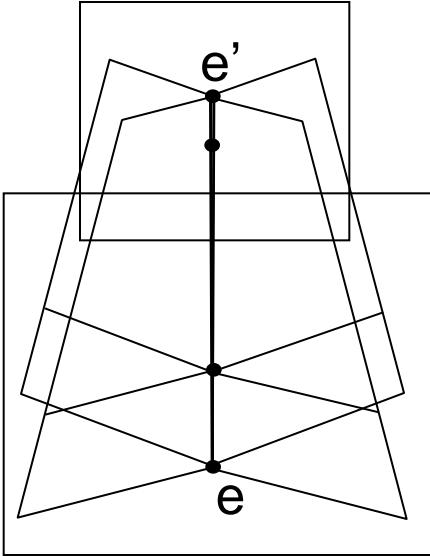
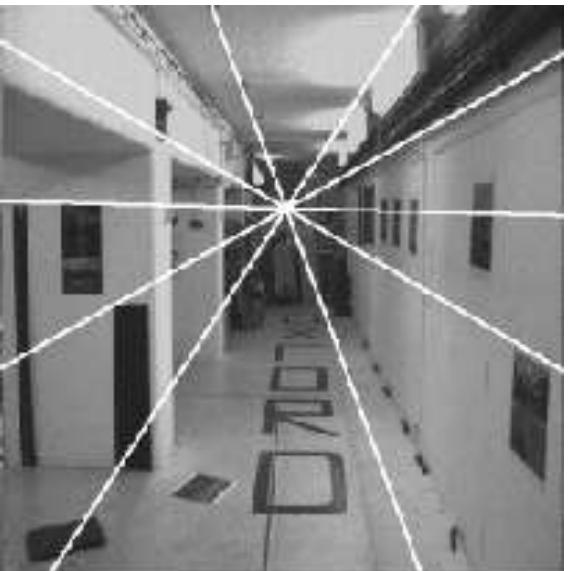
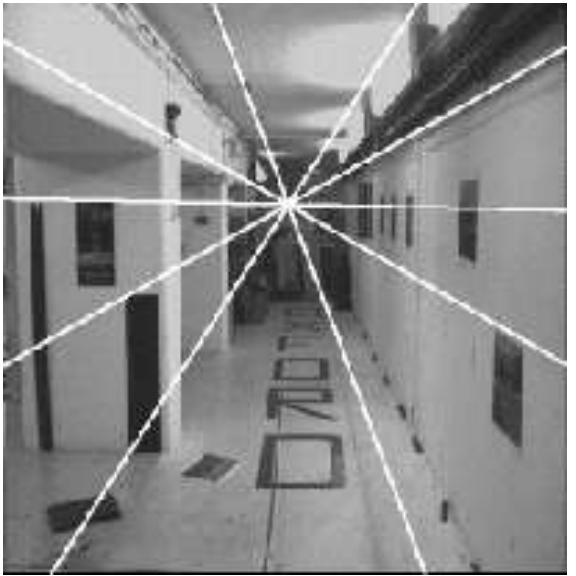


Движение, параллельное картинной плоскости



# Пример: движение вперед

---

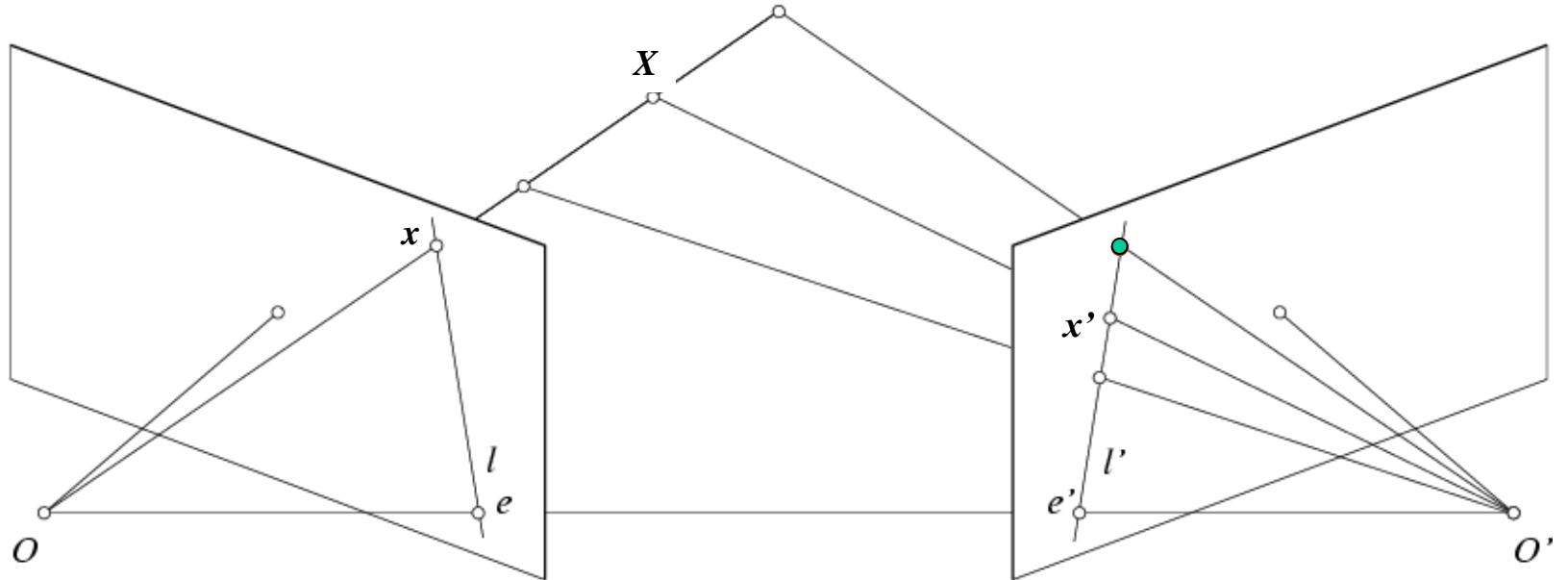


У эпиполей одинаковые координаты в обоих изображениях.  
Точки движутся вдоль линий, исходящих из  $e$  – «фокуса расширения»



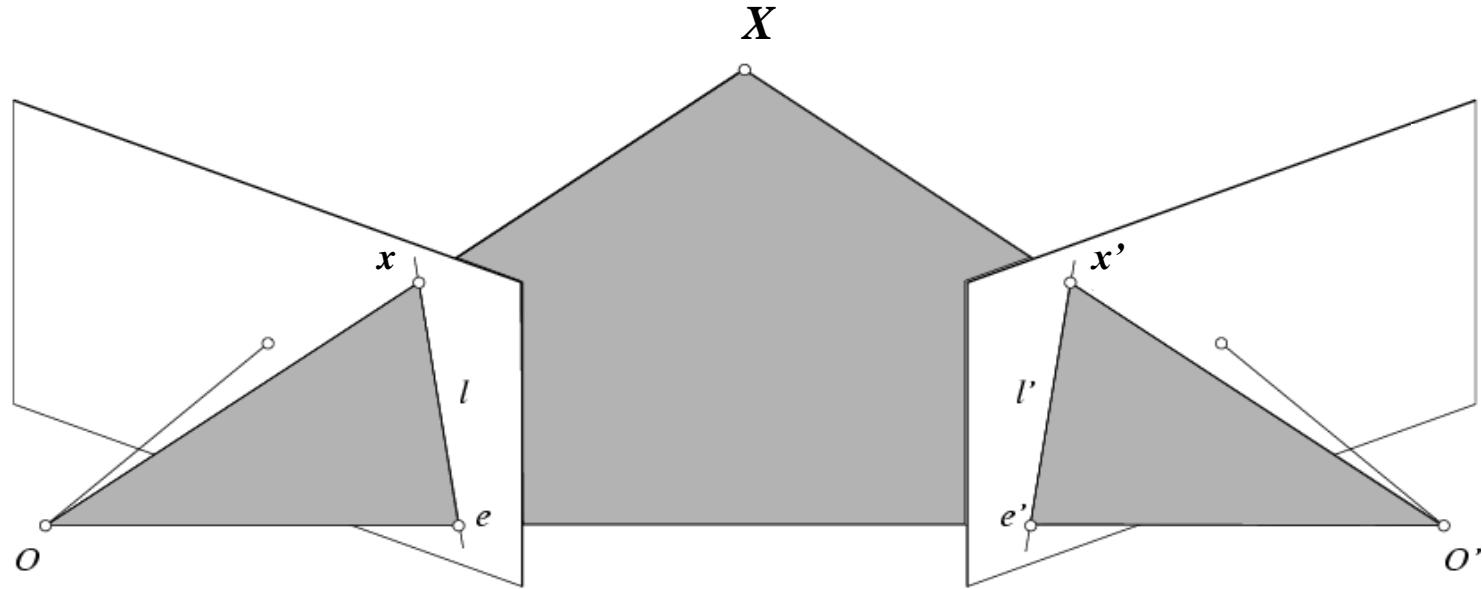
# Фундаментальная и существенная матрицы

# Эпиполярное ограничение



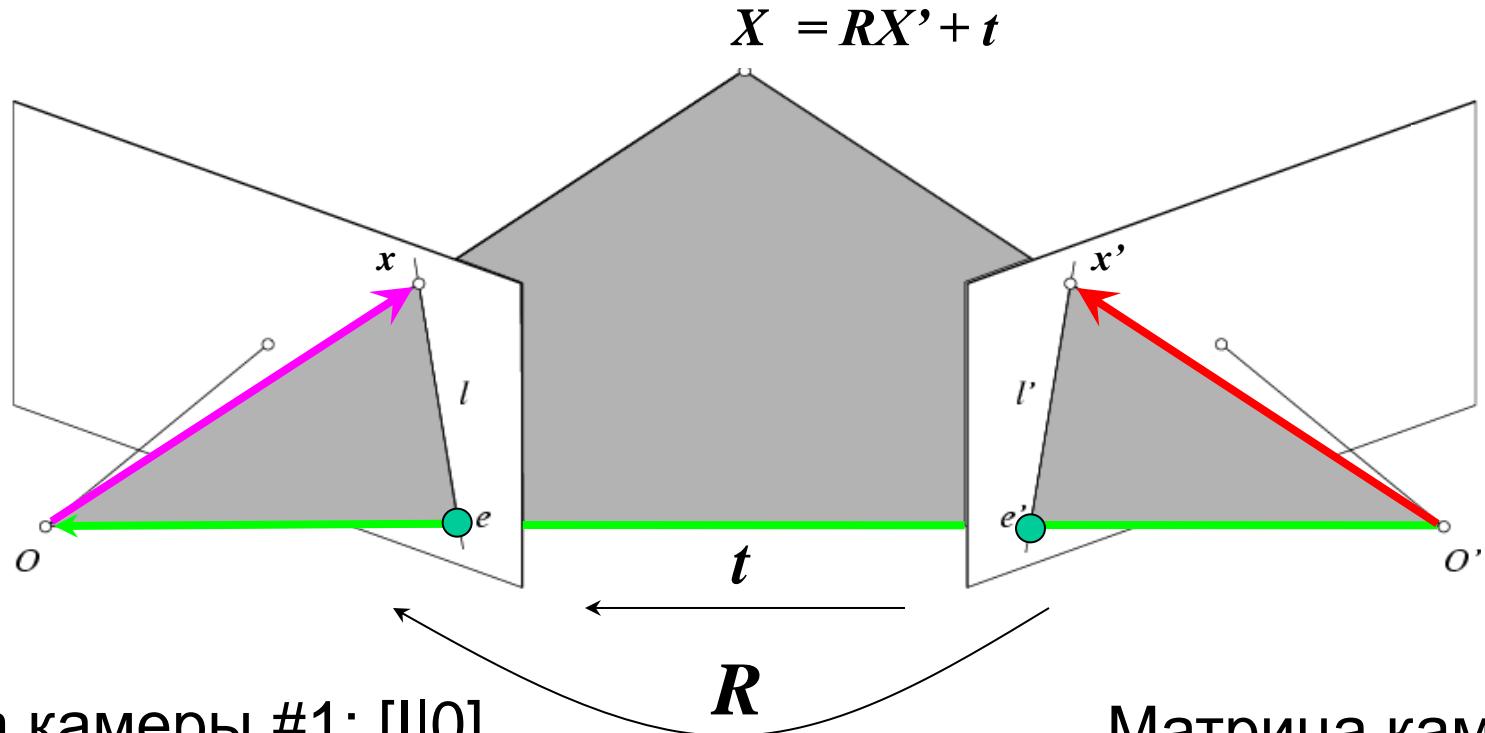
- Интересный вывод:
  - Линия  $l'$  не зависит от  $X$ , если  $x$  и калибровка камер известна
  - Линия  $l$  не зависит от  $X$ , если  $x'$  и калибровка камер известна
  - Между  $x$ ,  $x'$  и калибровкой камер существует зависимость
  - Можем вывести эту зависимость

# Калибранный случай



- Совместим глобальную систему координат с координатной системой первой камеры
- Допустим, известны внутренние калибровки камер  $K, K'$  и внешняя калибровка второй камеры  $R$  и  $T$
- Умножим матрицу проекции каждой камеры на обратную матрицу калибровки  $K^{-1}, K'^{-1}$ 
  - Также отобразим точки  $x, x'$  с изображения на картинную плоскость

# Калибранный случай



Матрица камеры #1:  $[I|0]$

$$X = (u, v, w, 1)^T$$

$$x = (u, v, w)^T$$

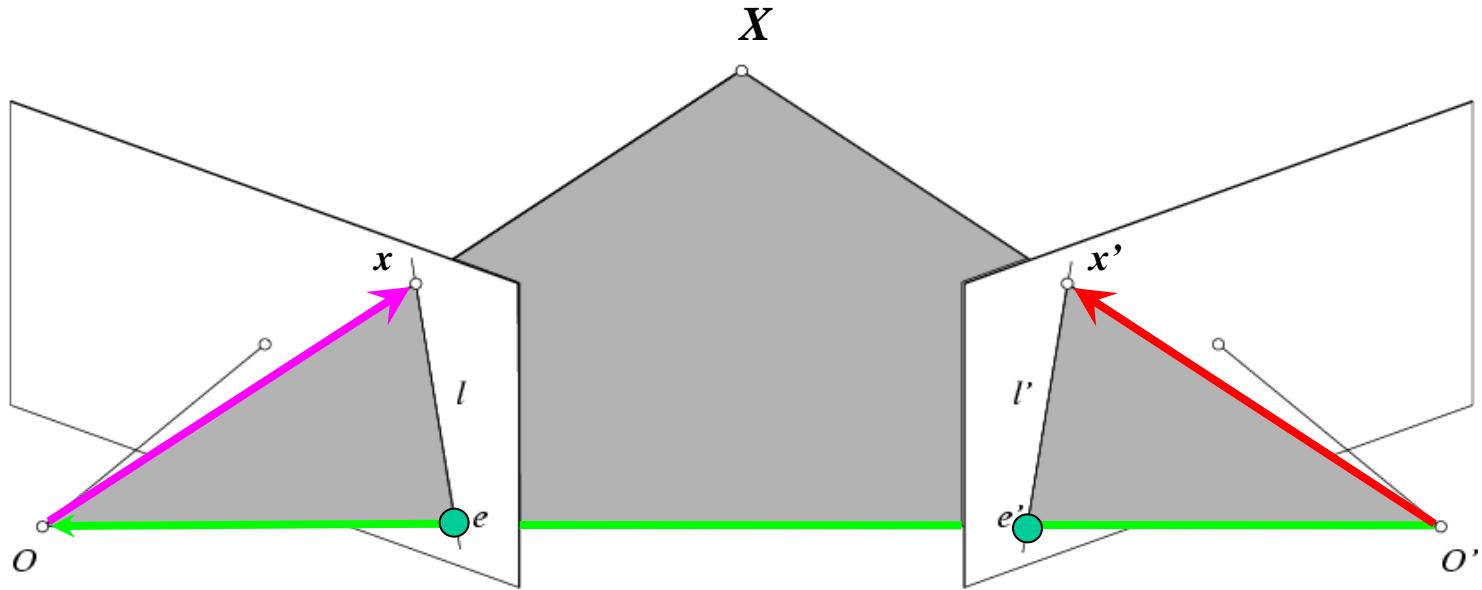
Рассмотрим 3 вектора:  $Ox$ ,  $OO'$ , и  $O'x'$

Векторы  $x$ ,  $t$ , и  $Rx'$  копланарны

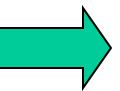
Матрица камеры #2:  $[R^T | -R^T t]$

Вектор  $x'$  во второй системе координат имеет координаты  $Rx'$  в первой

# Калиброванный случай



$$x \cdot [t \times (Rx')] = 0$$



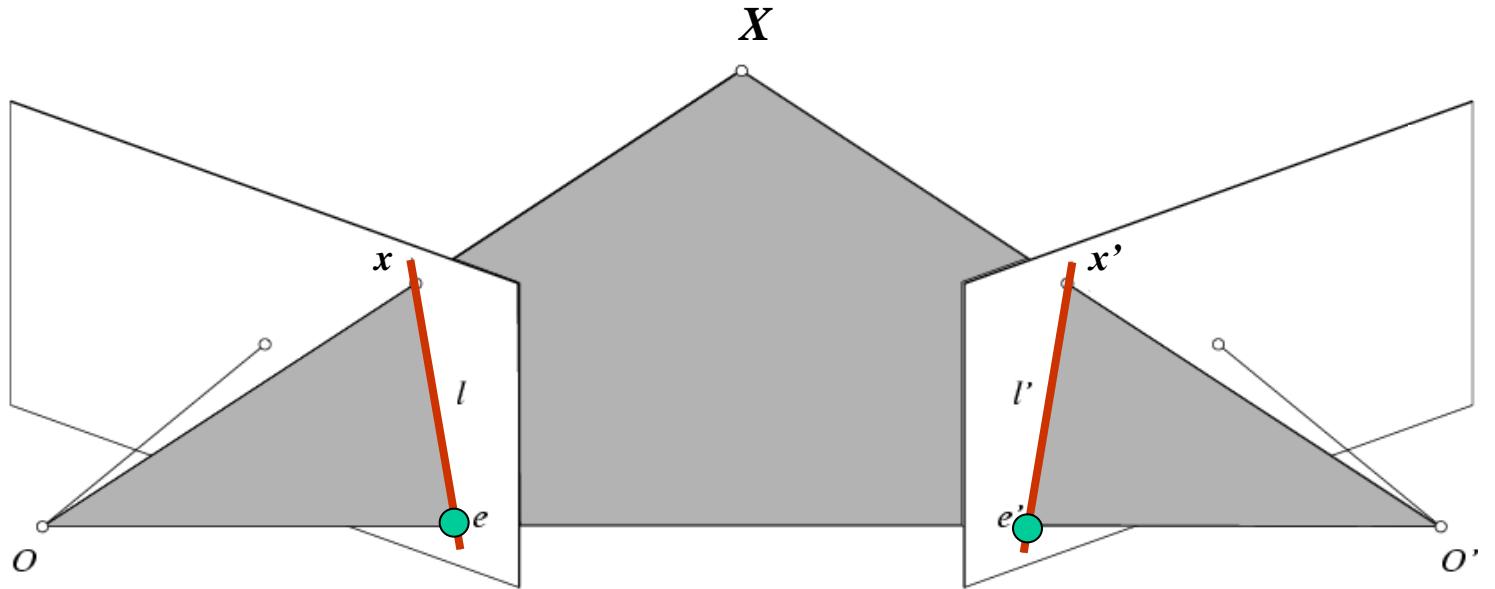
$$x^T E x' = 0 \quad \text{где} \quad E = [t_x]R$$



**Существенная матрица  
(Essential Matrix)**  
(Longuet-Higgins, 1981)

Вектора  $x$ ,  $t$ , и  $Rx'$  копланарны

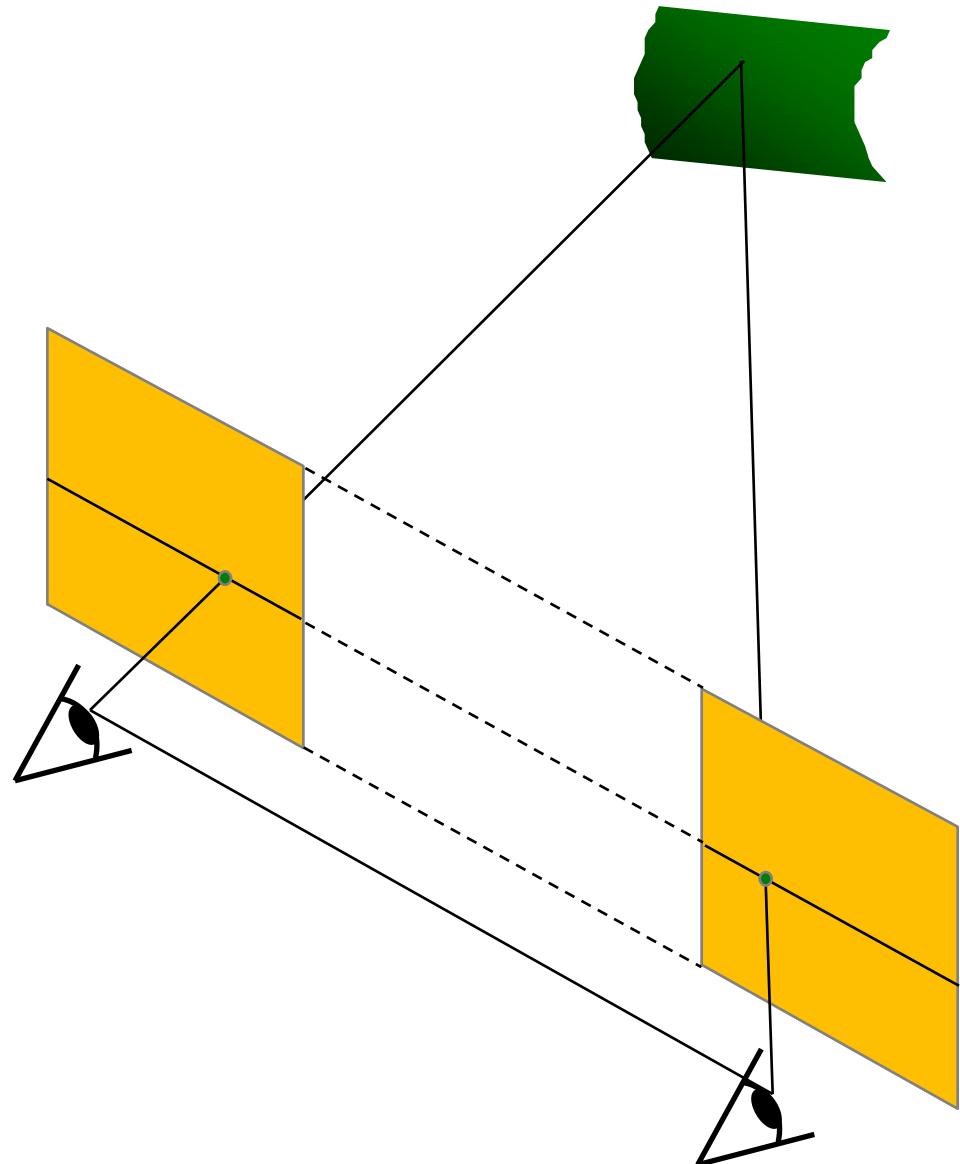
# Свойства существенной матрицы



$$x \cdot [t \times (Rx')] = 0 \quad \rightarrow \quad x^T E x' = 0 \quad \text{где} \quad E = [t_x]R$$

- $E x'$  - эпиполярная линия, соответствующая  $x'$  ( $l = E x'$ )
- $E^T x$  - эпиполярная линия, соответствующая  $x$  ( $l' = E^T x$ )
- $E e' = 0$  и  $E^T e = 0$
- $E$  вырождена (ранг 2)
- $E$  имеет 5 степеней свободы

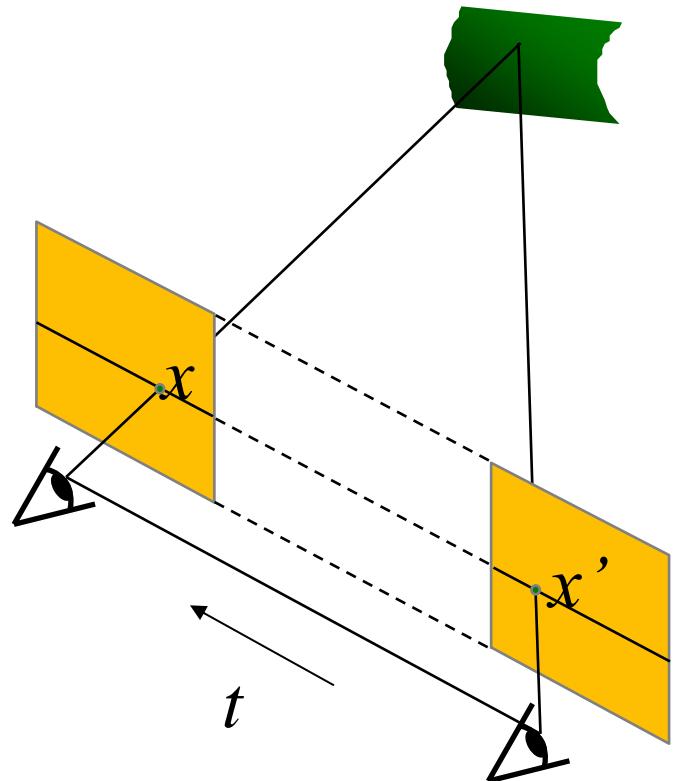
# Простейший случай параллельных камер



- Вторая камера не повёрнута относительно первой
- Смещение второй камеры относительно первой строго по горизонтали
- Внутренняя калибровка одинакова
- Эпиполярные линии расположены вдоль строк изображений



# Простейший случай параллельных камер



Эпиполярное ограничение:

$$x^T E x' = 0, \quad E = t \times R$$

$$R = I \quad t = (T, 0, 0)$$

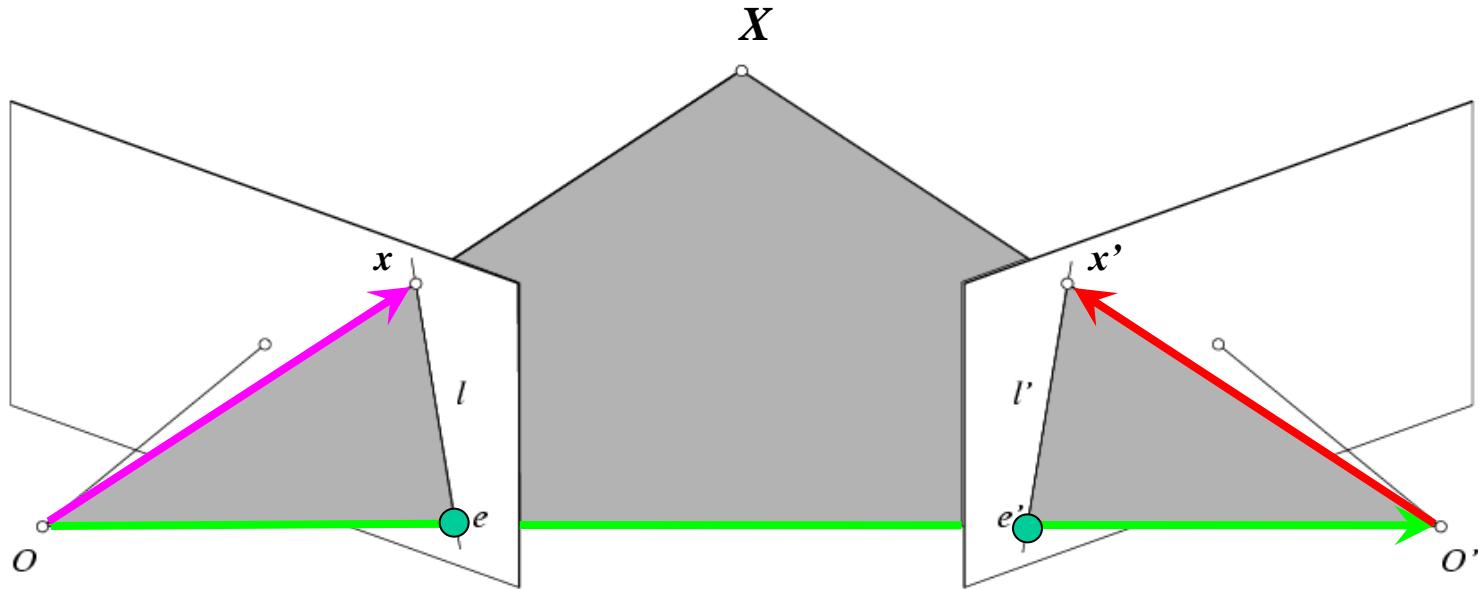
$$E = t \times R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

$$(u \quad v \quad 1) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \quad (u \quad v \quad 1) \begin{pmatrix} 0 \\ -T \\ T v' \end{pmatrix} = 0 \quad T v = T v'$$

v-координаты (строки) у соответствующих точек одинаковы



# Некалибранный случай

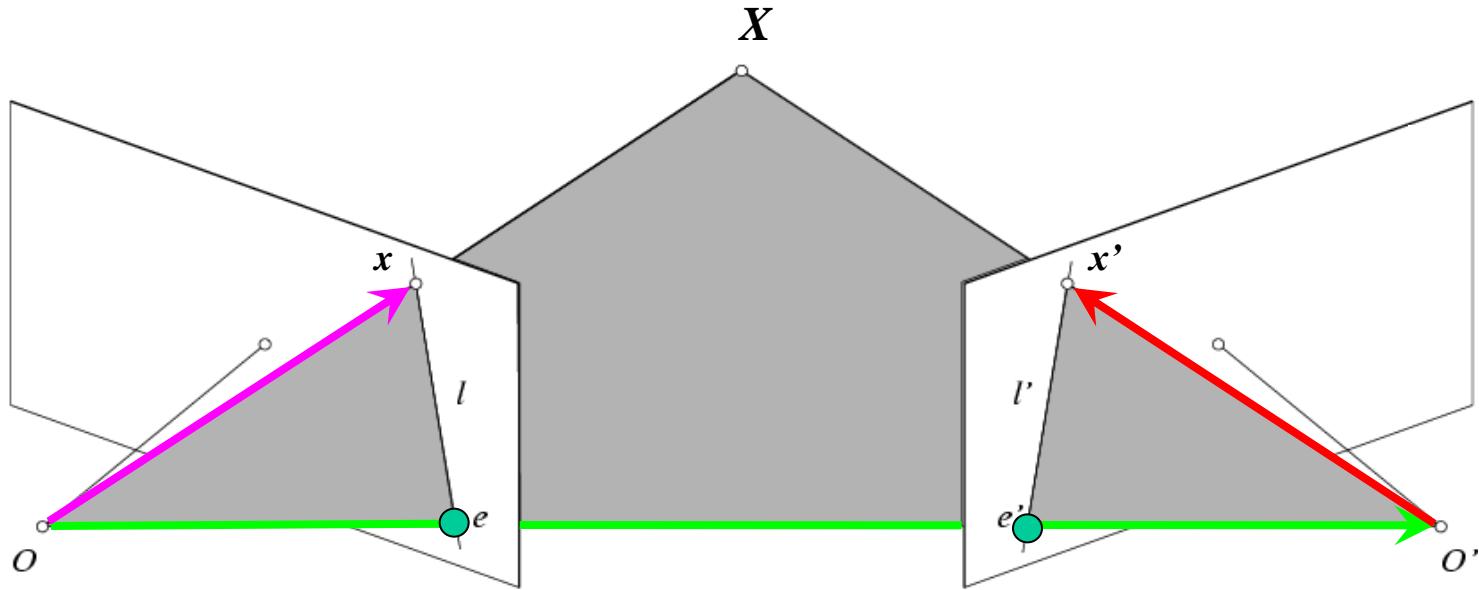


- Пусть матрицы калибровки  $K$  и  $K'$  для обеих камер неизвестны
- Запишем эпиполярное ограничение через *неизвестные* нормализованные координаты:

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

$$x = K \hat{x}, \quad x' = K' \hat{x}'$$

# Некалибранный случай



$$\hat{x}^T E \hat{x}' = 0 \quad \rightarrow \quad x^T F x' = 0 \quad \text{где} \quad F = K^{-T} E K'^{-1}$$

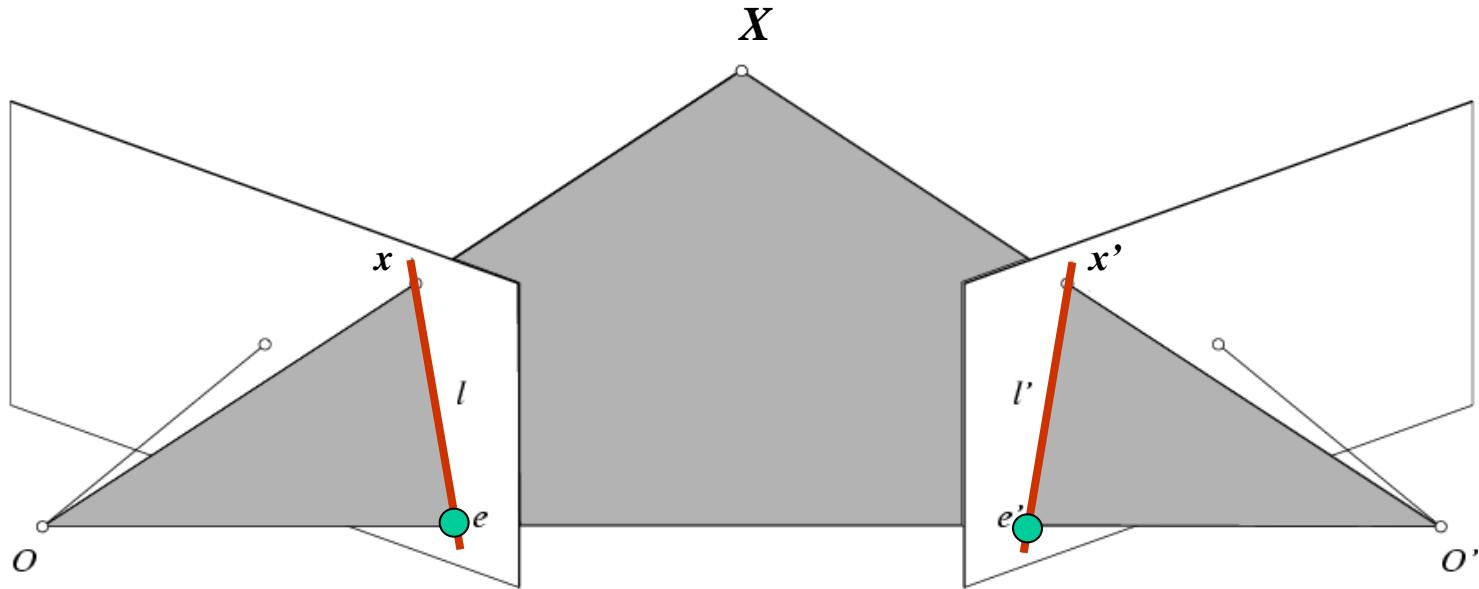
$$x = K \hat{x}$$

$$x' = K' \hat{x}'$$

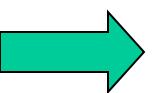


**Фундаментальная матрица  
(Fundamental Matrix)**  
(Faugeras and Luong, 1992)

# Свойства



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

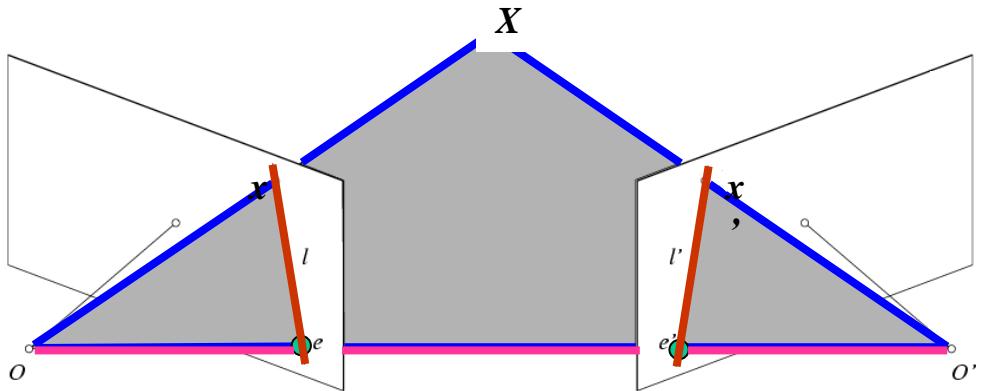


$$x^T F x' = 0 \quad \text{где} \quad F = K^{-T} E K'^{-1}$$

- $Fx$  - эпиполярная линия, соответствующая  $x'$  ( $l = Fx$ )
- $F^T x$  - эпиполярная линия, соответствующая  $x$  ( $l' = F^T x$ )
- $F e' = 0$  и  $F^T e = 0$
- $F$  вырождена (ранг 2)
- $F$  имеет семь степеней свободы



# Резюме эпиполярной геометрии



- Базовая линия (Baseline)
- Эпиполярная плоскость (Epipolar Plane)
- Эпиполи/Эпиполяры (Epipoles)
- Эпиполярные линии (Epipolar Lines)

Пара камер порождает «эпиполярную геометрию», т.е. зависимость между парами соответствующих точек и калибровкой камер

Существенная матрица  
(калибранный случай):

$$x^T E x' = 0 \quad \text{где} \quad E = [t_x] R$$

Фундаментальная матрица  
(некалибранный случай):

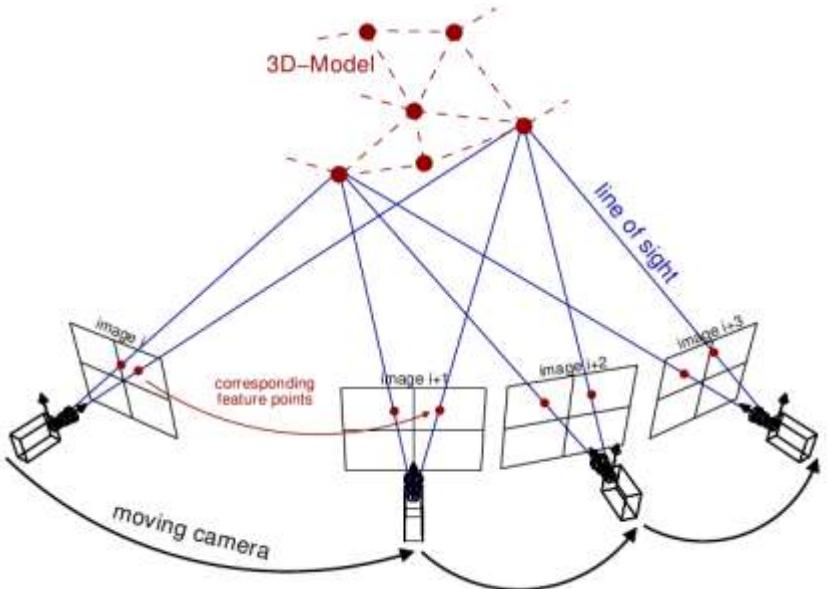
$$x^T F x' = 0 \quad \text{где} \quad F = K^{-T} E K'^{-1}$$

Существенная и фундаментальная матрицы описывают зависимости между калибровкой камер и парами соответствующих точек на изображениях



Зачем считать фундаментальную матрицу?

# Оценка движения камеры



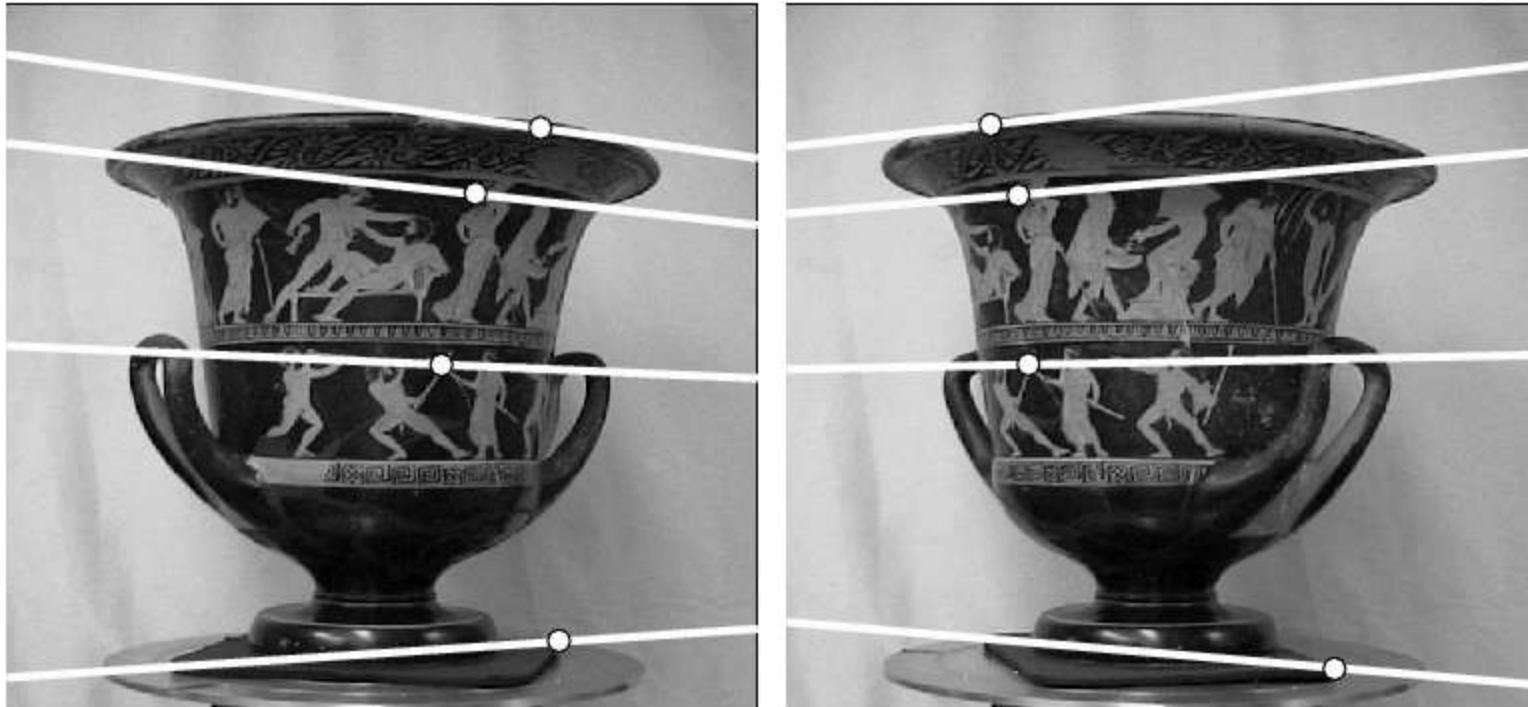
Наблюдение 3Д сцены с нескольких ракурсов (камер)

- Движение камеры – поворот и сдвиг одной камеры относительно другой
- Вспомним, что  $E = [t_x]R$ , т.е. существенная матрица определяется как раз движением камеры
- Вспомним, что  $F = K^{-T}EK'^{-1}$
- Значит, если мы знаем  $F$  и  $K, K'$ , то можем вычислить  $E$
- $K$  можно получить из EXIF в jpg файлах
- $E$  можем факторизовать и получить  $R$  и  $T$  (с точностью до масштаба)



# Проверка эпиполярного ограничения

Что нам даёт известная  $F$ ?

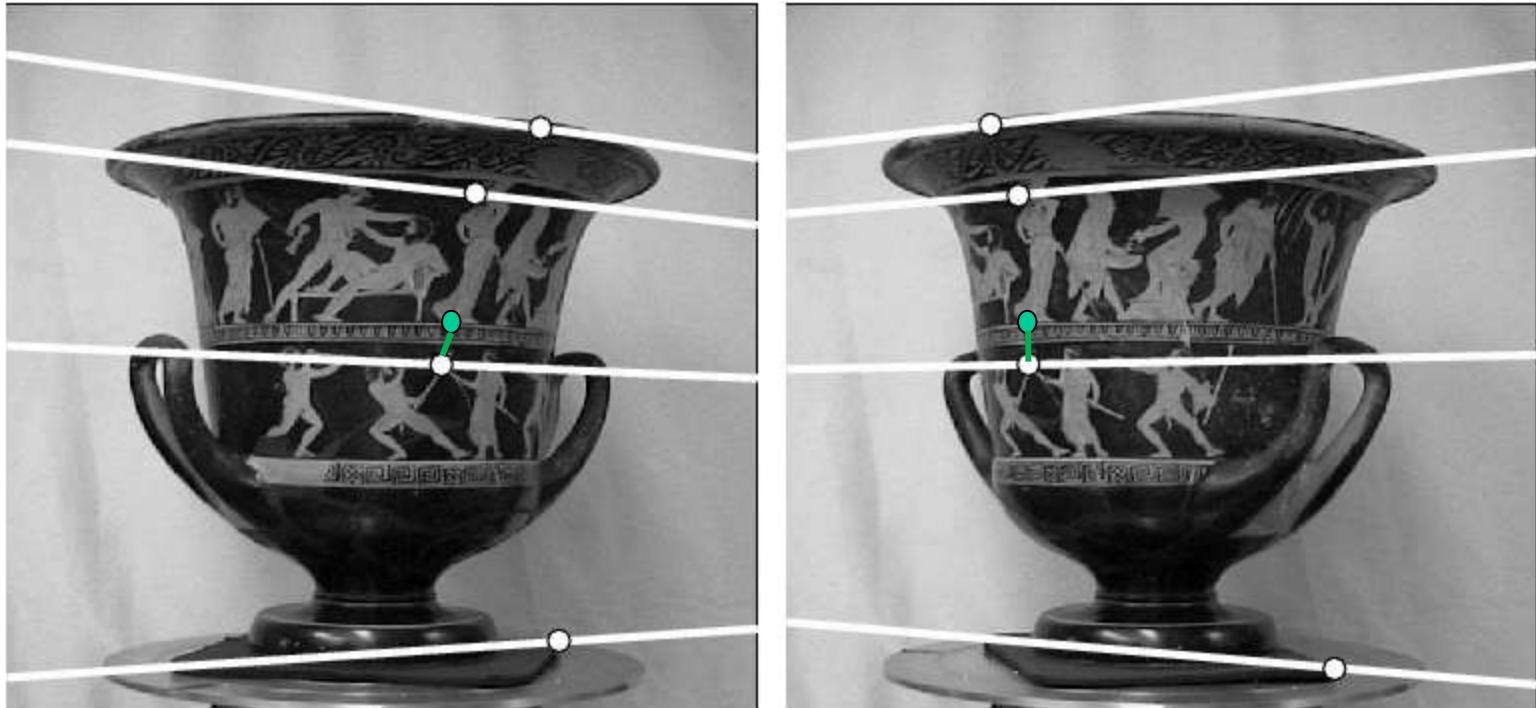


- Пусть  $(x, x')$  – пара соответствующих точек на 2х изображениях
- Тогда должно выполняться  $x^T F x' = 0$
- Если не соблюдается, значит  $(x, x')$  – ложное соответствие!



# Проверка эпиполярного ограничения

Что нам даёт известная  $F$ ?



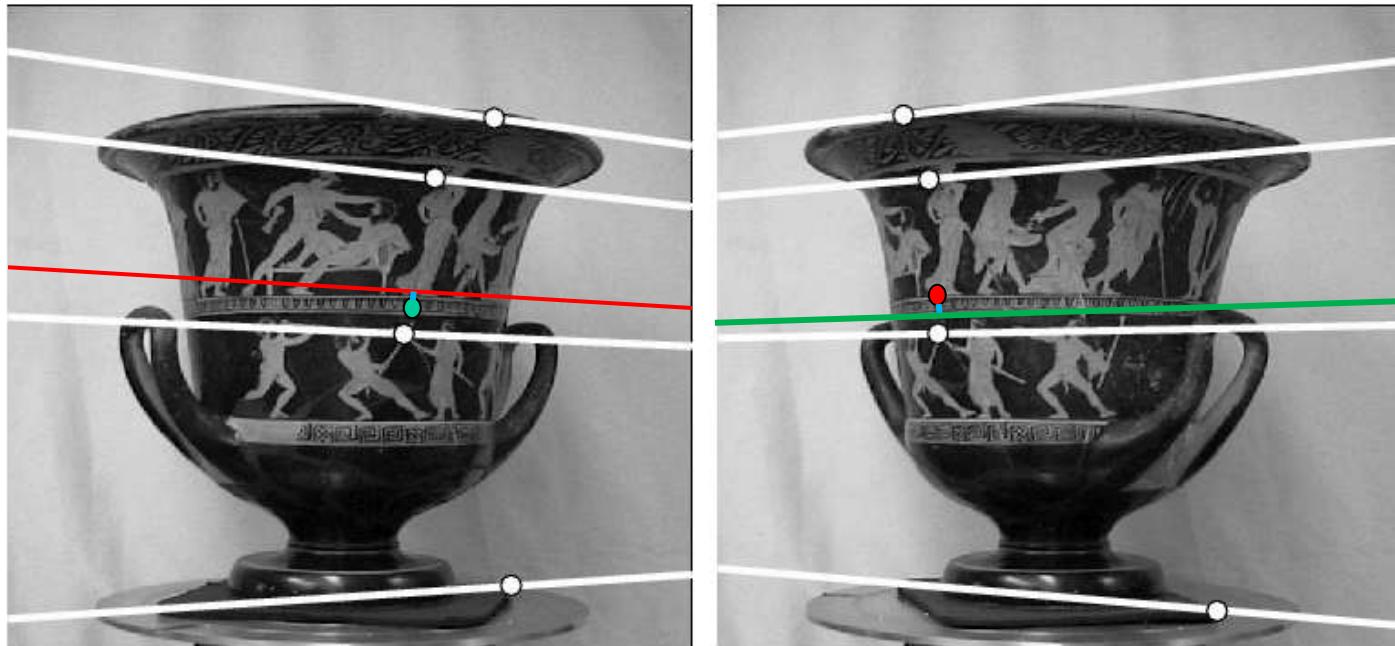
- $\|x^T F x'\|$  - алгебраическая ошибка, не имеющая физического смысла
- «Оптимальная ошибка» - сумма квадратов расстояний от измеренных точек, до «идеальных»

$$\sum_{i=1}^N [d^2(x'_i, \hat{x}'_i) + d^2(x_i, \hat{x}_i)], \text{ где } (\hat{x}'_i)^T F \hat{x}_i = 0$$



# Проверка эпиполярного ограничения

Что нам даёт известная  $F$ ?



- Вычисление оптимальной ошибки сложное, требует нелинейной оптимизации из-за необходимости найти «идеальную» пару для каждой измеренной
- Поэтому чаще считают ошибку переноса (transfer error)

$$\sum_{i=1}^N \left[ d^2(x'_i, Fx_i) + d^2(x_i, F^T \hat{x}_i) \right]$$



# Оценка фундаментальной матрицы



# 8-и точечный алгоритм

- Фундаментальная матрица накладывает ограничения на пары соответствующих точек  $x^T F x' = 0$
- Попробуем её вычислить напрямую из соответствий, без калибровки камеры
- Даны соответствия  $(x, x')$ :  $x = (u, v, 1)^T$ ,  $x' = (u', v', 1)^T$

$$(u, v, 1) \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \quad \rightarrow \quad (uu', uv', u, vu', vv', v, u', v', 1) \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

- 1 ссоответствие даёт 1 уравнение на  $F$
- Если точек много, можем свести к обычной задаче однородных наименьших квадратов



# 8-и точечный алгоритм

$$(uu', uv', u, vu', vv', v, u', v', 1) \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0 \quad \begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Минимизируем:

$$\sum_{i=1}^N (x_i^T F x_i')^2$$

При условии

$$|F|^2 = 1$$

- Можно решать как однородную систему уравнений, так и неоднородную

# Недостатки 8и точечного метода



$$\begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

# Недостатки 8и точечного метода



250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48

$$\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

- Плохая численная обусловленность
- Можно поправить масштабированием данных
- После масштабирования среднее точек должно быть в 0, средняя длина  $\sqrt{2}$



# Приведение к рангу 2

---

- Фундаментальная матрица  $F$  имеет ранг 2
- В общем случае, результат DLT-оценки  $F$  имеет ранг 3 (из-за влияния шума)
- Необходимо привести к рангу 2
- Найдем  $F'$ , минимизирующую норму Фробениуса:

$$\|F - F'\|, \text{ где } \text{rang}(F') = 2$$

- Воспользуемся SVD-разложением:
- $F = UDV^T$ , или  $F = U\text{diag}(r,s,t)V^T$ , где  $r \geq s \geq t$
- $F' = U\text{diag}(r,s,0)V^T$

# Оценка на практике

---



- Множество соответствий ( $N>8$ ), в которых могут быть ошибки
- Нужно использовать робастные методы
- Например, RANSAC с ошибкой переноса:

$$\sum_{i=1}^N \left[ d^2(x'_i, Fx_i) + d^2(x_i, F^T \hat{x}_i) \right]$$

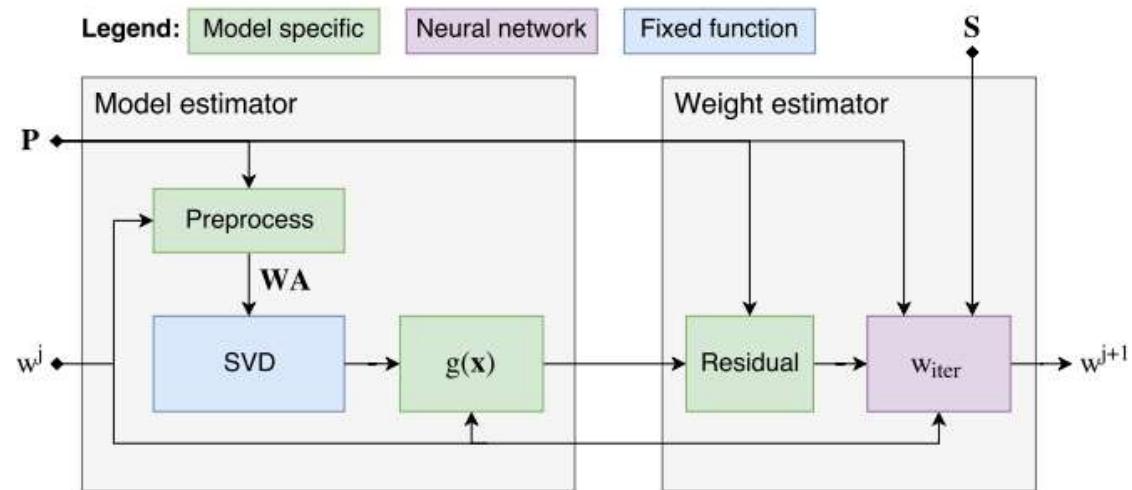
- Затем уточняют по всем найденным корректным соответствиям
  - В начале линейным методом
  - Потом нелинейной оптимизацией с ошибкой переноса или оптимальной ошибкой

# Нейросети для оценки фундаментальной матрицы



- Напрямую оценивать  $F$  по изображениям без особых точек не получается
- Ищут подходы, которые используют нейросети в рамках существующего подхода
- Например, можно представить решение задачи в виде обобщённых взвешенных МНК:

$$\mathbf{x}^{j+1} = \arg \min_{\mathbf{x}: \|\mathbf{x}\|=1} \sum_{i=1}^N w(\mathbf{p}_i, \mathbf{x}^j) \|(\mathbf{A}(\mathbf{P}))_i \cdot \mathbf{x}\|^2$$

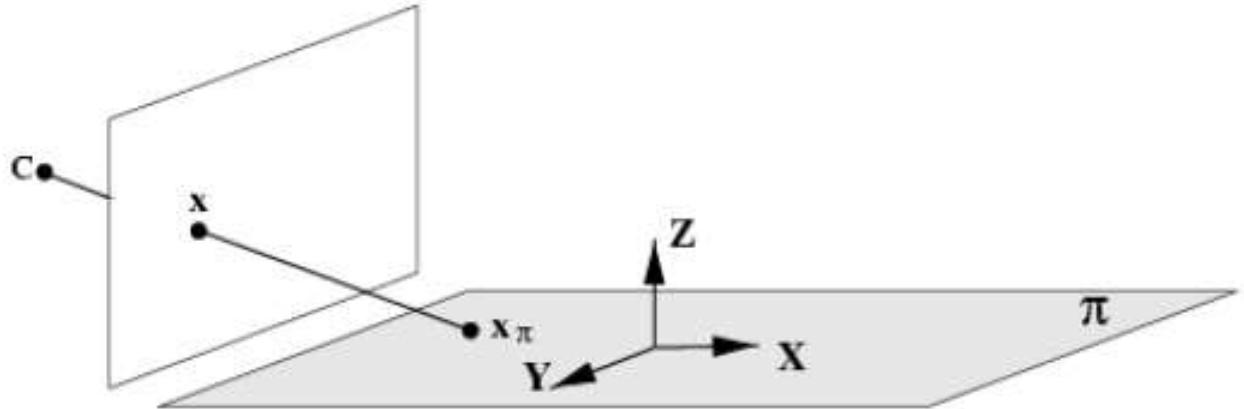


Нейросеть можно использовать  
для предсказания весов



# Гомография

# Вырожденный случай плоской сцены



Модель преобразования  
плоскости при перспективном  
преобразовании называется  
гомография (homography)

Частный случай:

- Пусть точка X лежит на плоскости
- Выберем специальную мировую систему координат

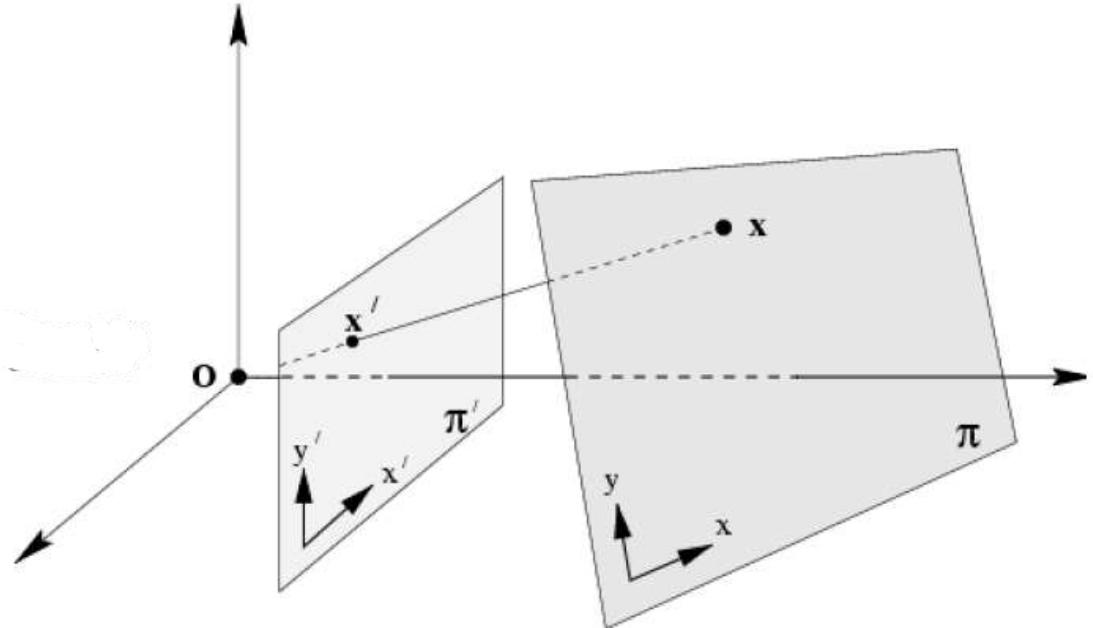
$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \cong \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{12} & p_{33} & p_{34} \end{bmatrix} * \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{12} & p_{34} \end{bmatrix} * \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

# Общий случай гомографии



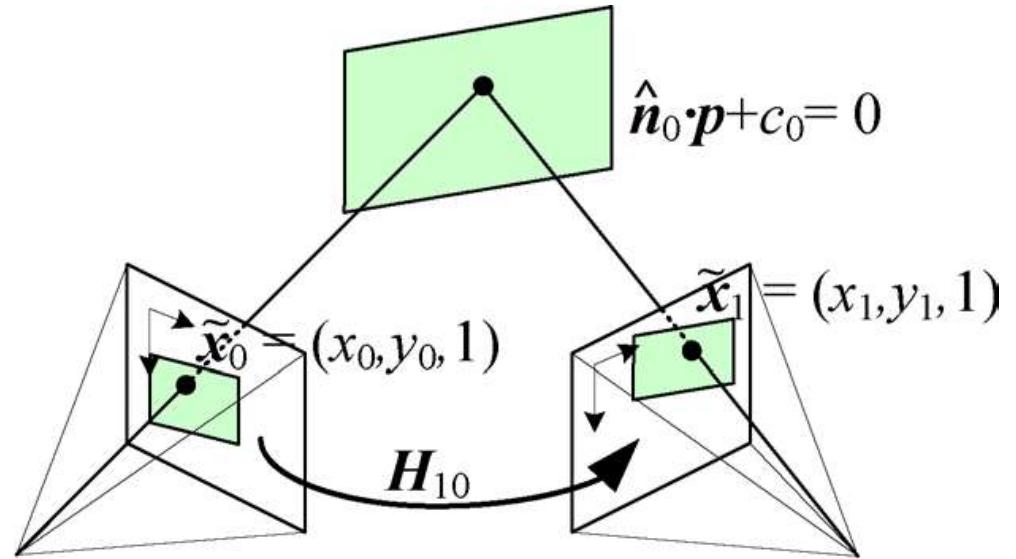
- В общем случае гомография также описывается матрицей 3x3

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{12} & h_{33} \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

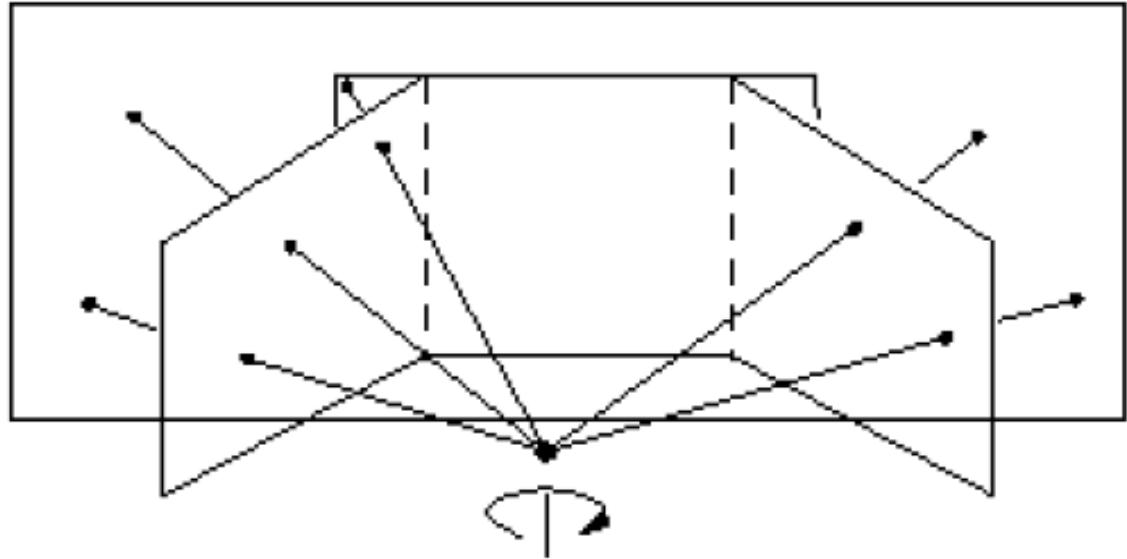


- 8 степеней свободы

# Применение гомографии



Плоская сцена



Камера на штативе

Вывод: если преобразование одного изображения в другое хорошо описывается гомографией, то мы либо наблюдаем плоскую сцену, либо центры проекций камер близки



# Оценка гомографии

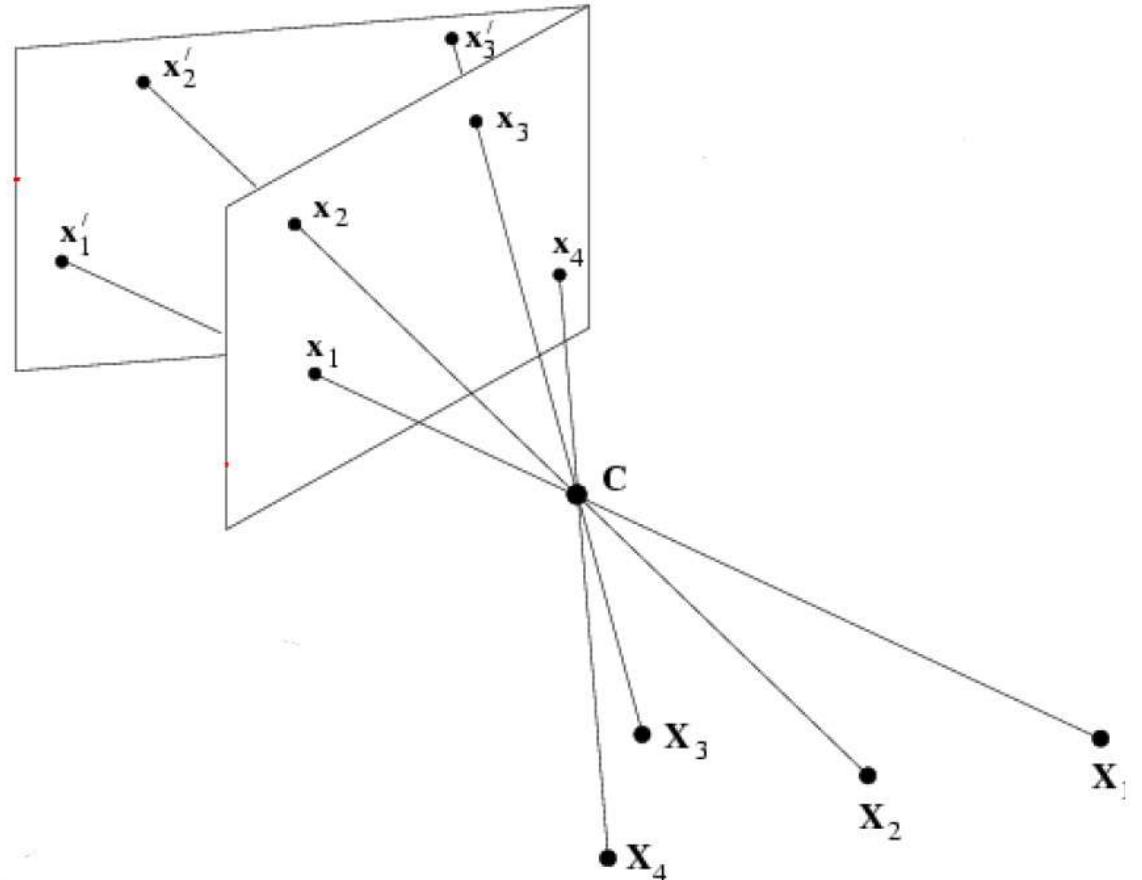
# Начнём с линейного подхода (DLT)



- Гомография – линейное преобразование в однородных координатах

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{12} & h_{33} \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- 8 степеней свободы
- Возьмём соответствия и составим систему уравнений



# Линейный метод



$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \Leftrightarrow \begin{cases} uh_1 + vh_2 + h_3 = u' \\ uh_4 + vh_5 + h_6 = v' \\ uh_7 + vh_8 + h_9 = 1 \end{cases} \Leftrightarrow \begin{bmatrix} 0 & 0 & 0 & -u & -v & -1 & v'u & v'v & v' \\ u & v & 1 & 0 & 0 & 0 & -u'u & -u'v & -u' \\ -v'u & -v'v & -v' & u'u & u'v & u' & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow A\mathbf{h} = \mathbf{0}$$

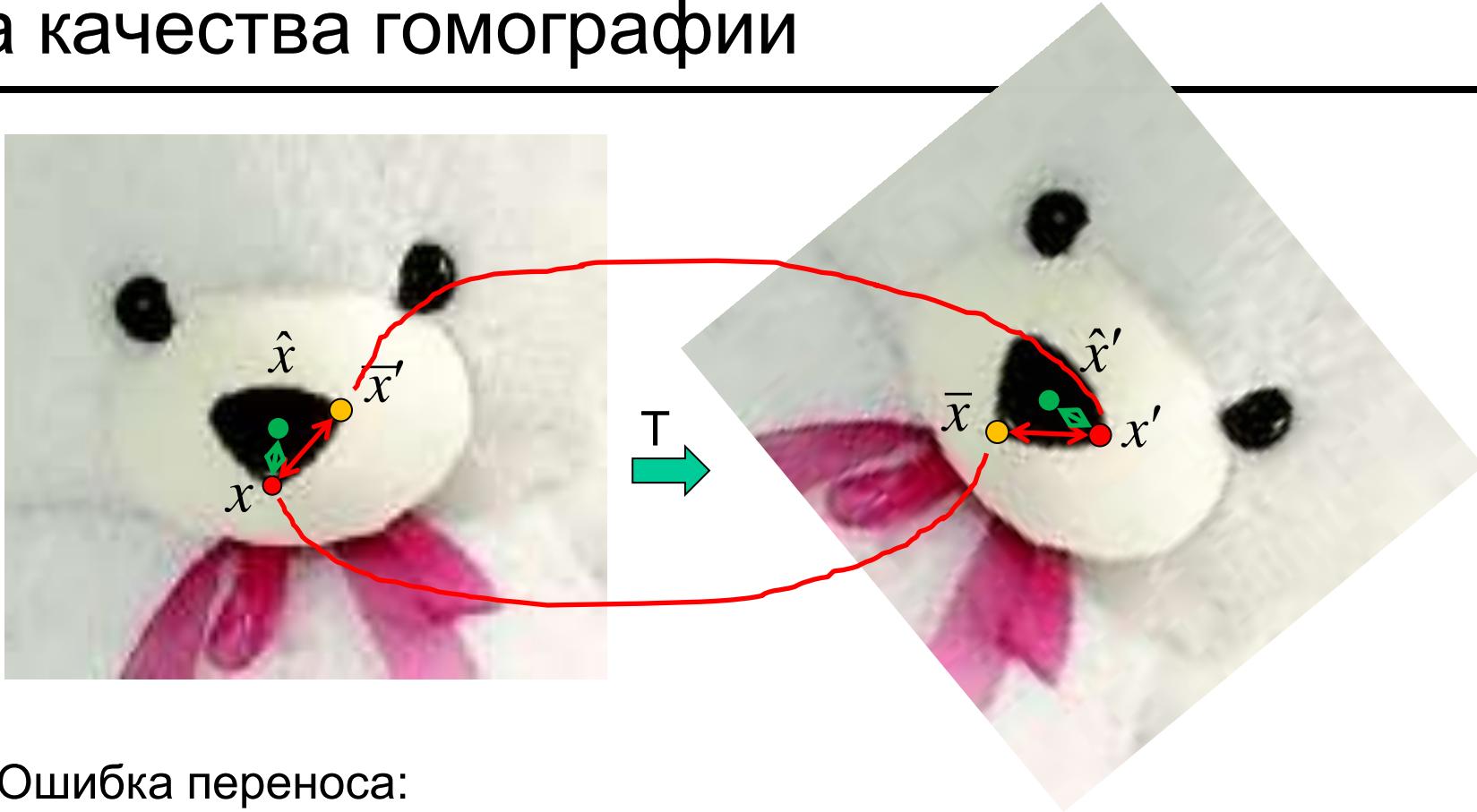


# Линейный метод

$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \Leftrightarrow \begin{cases} uh_1 + vh_2 + h_3 = u' \\ uh_4 + vh_5 + h_6 = v' \\ uh_7 + vh_8 + h_9 = 1 \end{cases} \Leftrightarrow \begin{bmatrix} 0 & 0 & 0 & -u & -v & -1 & v'u & v'v & v' \\ u & v & 1 & 0 & 0 & 0 & -u'u & -u'v & -u' \\ -v'u & -v'v & -v' & u'u & u'v & u' & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Leftrightarrow A\mathbf{h} = \mathbf{0}$$

- Зе уравнение – линейная комбинация 1 и 2го
- Поэтому 1 пара соответствующих точек даёт 2 уравнения на  $\mathbf{h}$
- Из 4x пар точек получаем систему вида  $A\mathbf{h} = 0$  при  $\|\mathbf{h}\| = 1$
- Можем решить через SVD-разложение
- $A = USV^T$
- $\mathbf{h}$  – последний столбец  $V$

# Оценка качества гомографии



Ошибка переноса:

$$\sum_i d^2(x'_i, \bar{x}_i) + d^2(x_i, \bar{x}'_i) \quad \bar{x} = Hx \quad \bar{x}' = H^{-1}x'$$

Геометрическая ошибка:

$$\sum_i d^2(x'_i, \hat{x}'_i) + d^2(x_i, \hat{x}_i) \quad \hat{x}'_i = T\hat{x}_i$$

# Вычисление гомографии на практике

---



- Множество соответствий ( $N > 4$ ), в которых могут быть ошибки
- Нужно использовать робастные методы
- Например, RANSAC с ошибкой переноса:

$$\sum_i d^2(x'_i, Hx_i) + d^2(H^{-1}x'_i, x_i)$$

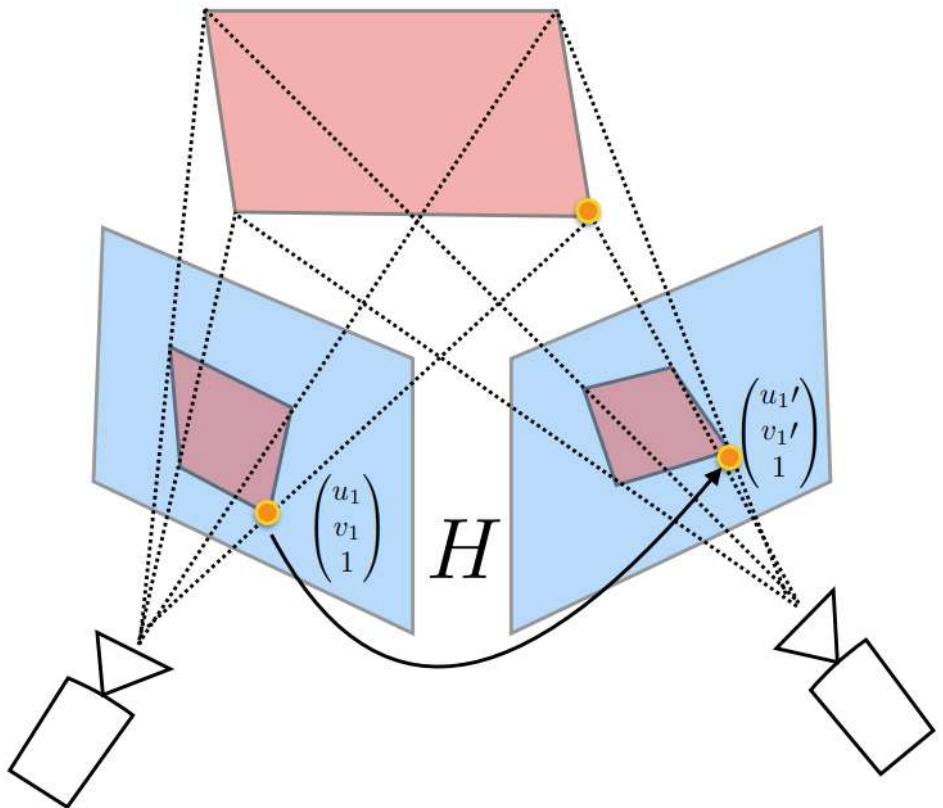
- Затем уточняют по всем найденным корректным соответствиям
  - В начале линейным методом
  - Потом нелинейной оптимизацией с ошибкой переноса или оптимальной ошибкой



# Оценка гомографии через нейросети



# Параметризация гомографии



$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \sim \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

$$H_{4point} = \begin{pmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{pmatrix} \quad \Delta u_1 = u'_1 - u_1$$

↑  
1-to-1 mapping

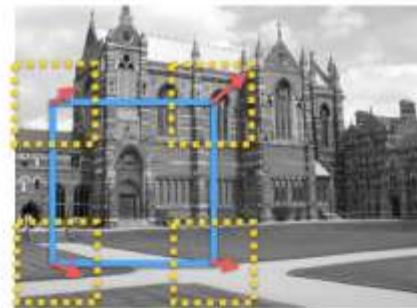
$$H_{matrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix}$$



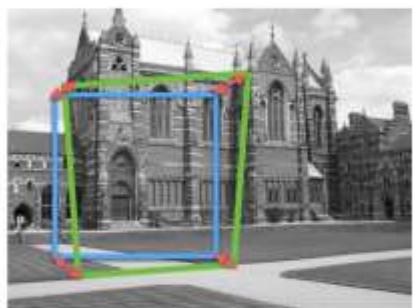
# Нейросетевая модель



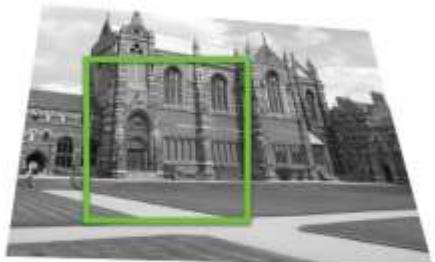
Step 1: Randomly crop at position  $\mathbf{p}$ . This is Patch A.



Step 2: Randomly perturb four corners of Patch A.



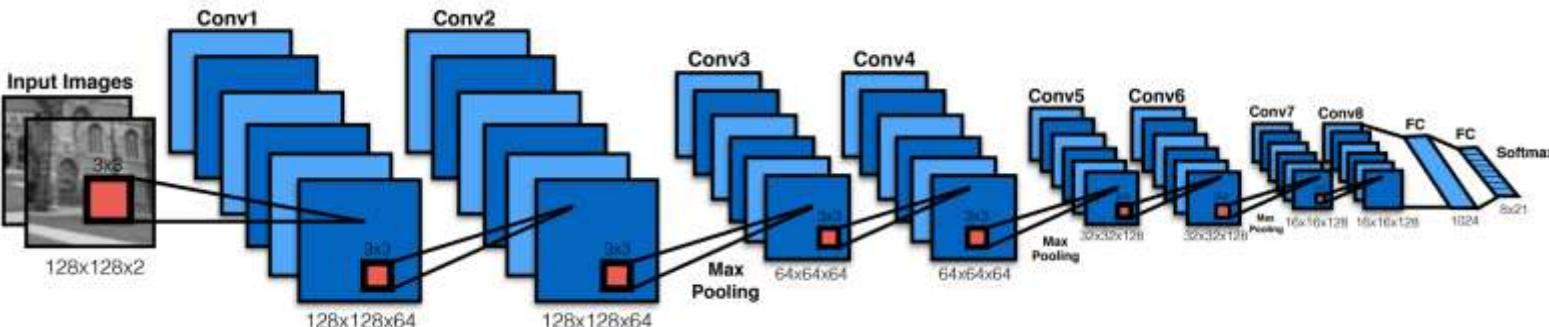
Step 3: Compute  $\mathbf{H}^{AB}$  given these correspondences.



Step 4: Apply  $(\mathbf{H}^{AB})^{-1} = \mathbf{H}^{BA}$  to the image, and crop again at position  $\mathbf{p}$ , this is Patch B.

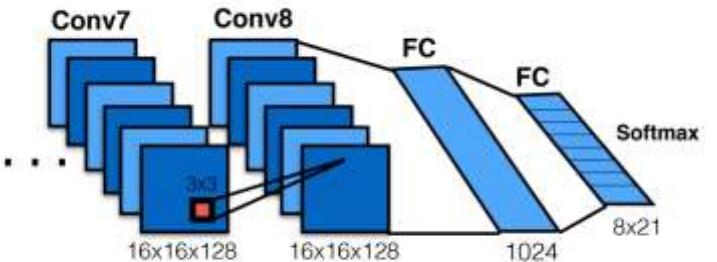


Step 5: Stack Patch A and Patch B channel-wise and feed into the network. Set  $\mathbf{H}^{AB}$  as the target vector.



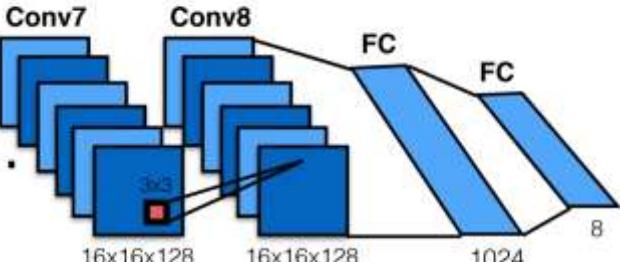
## Нейросеть

### Classification HomographyNet



$$\text{Loss: Cross-Entropy} = - \sum_x p(x) \log q(x)$$

### Regression HomographyNet



$$\text{Loss: Euclidean (L2)} = \frac{1}{2} \|p(x) - q(x)\|^2$$

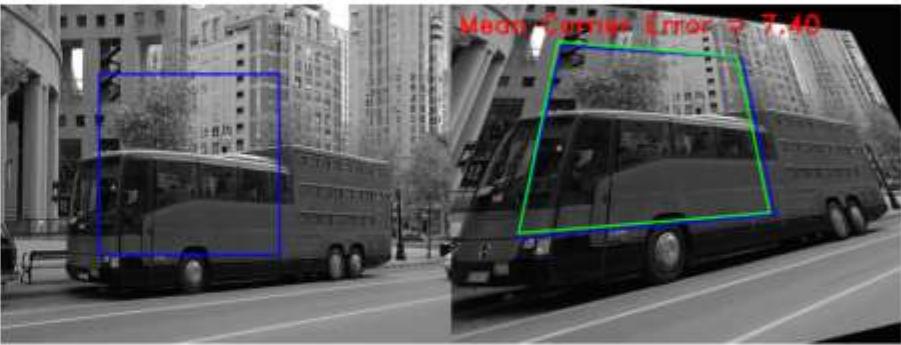
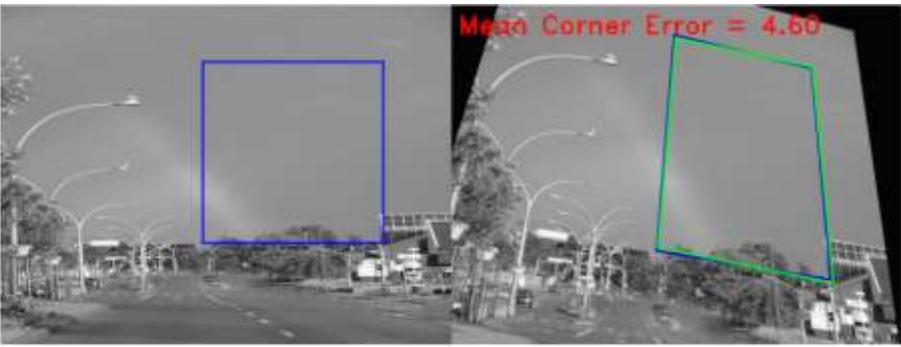
## Варианты предсказания результатов

<https://github.com/JirongZhang/DeepHomography>

# Сравнение моделей

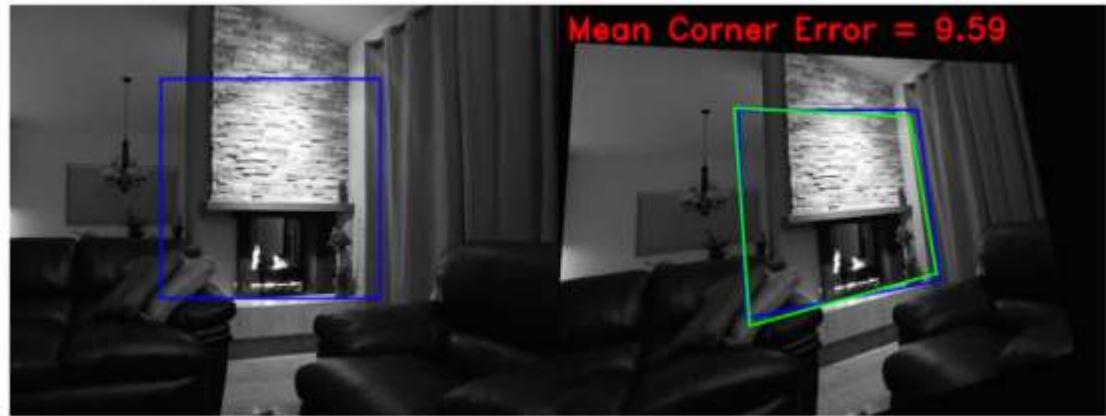
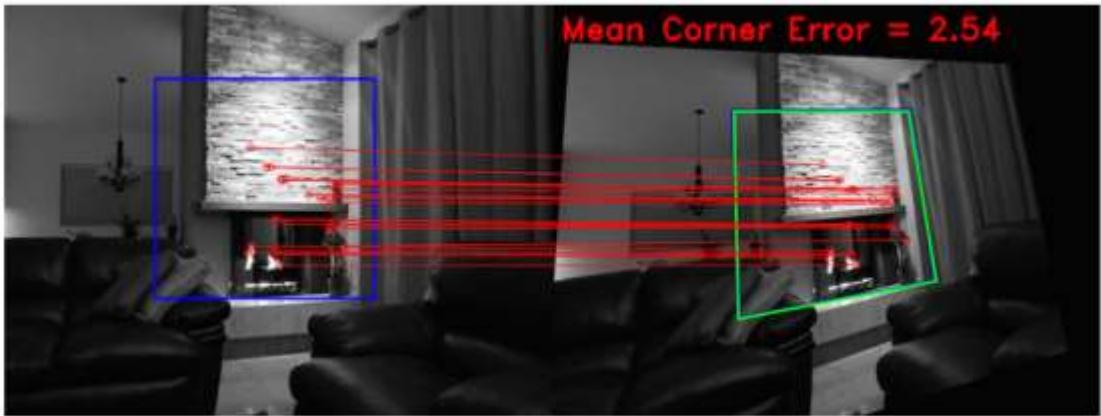
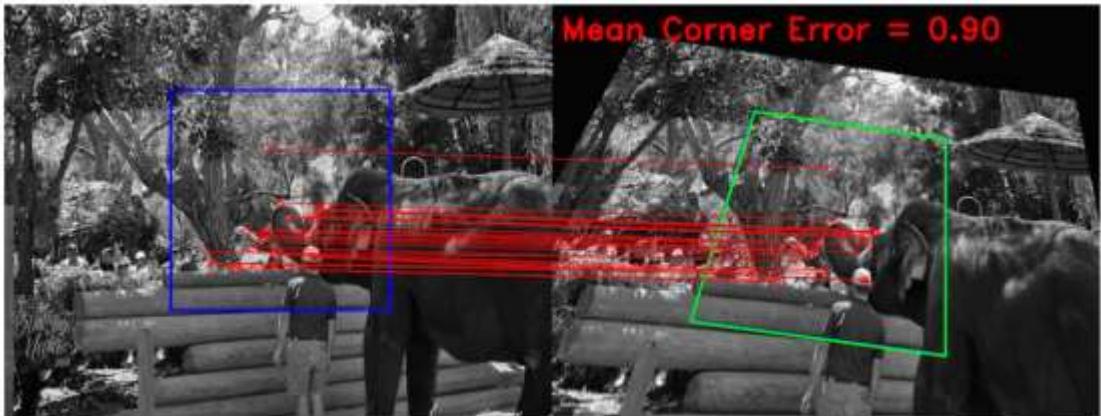


Точки + RANSAC



Нейросетевая модель

# Сравнение моделей



Точки + RANSAC

Нейросетевая модель

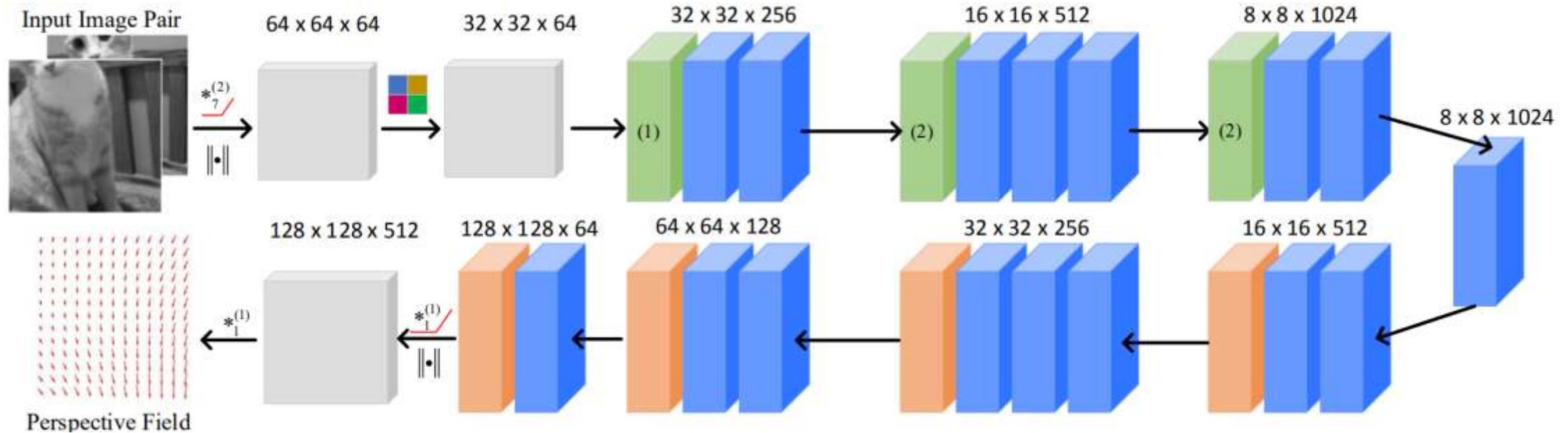


# Перспективные поля



- Оценка всего 4x точек чувствительна к ошибкам в каждой из них.
- Давайте оценивать вектора сдвигов для каждого пикселя (Perspective Fields)  
$$\{\mathbf{p}_i, \mathbf{q}_i\}_{i=1}^{HW} = \{\mathbf{p}_i, \mathbf{p}_i - \Delta\mathbf{p}_i\}_{i=1}^{HW}$$
- Для каждого пикселя это даст предсказание соответствующей точки
- По множеству точек с помощью RANSAC мы можем оценить гомографию

# Оценка PF



- Для оценки PF нужно использовать попиксельную модель, Encoder-Decoder
- На некоторых бенчмарках это SotA( 18.02.2022)

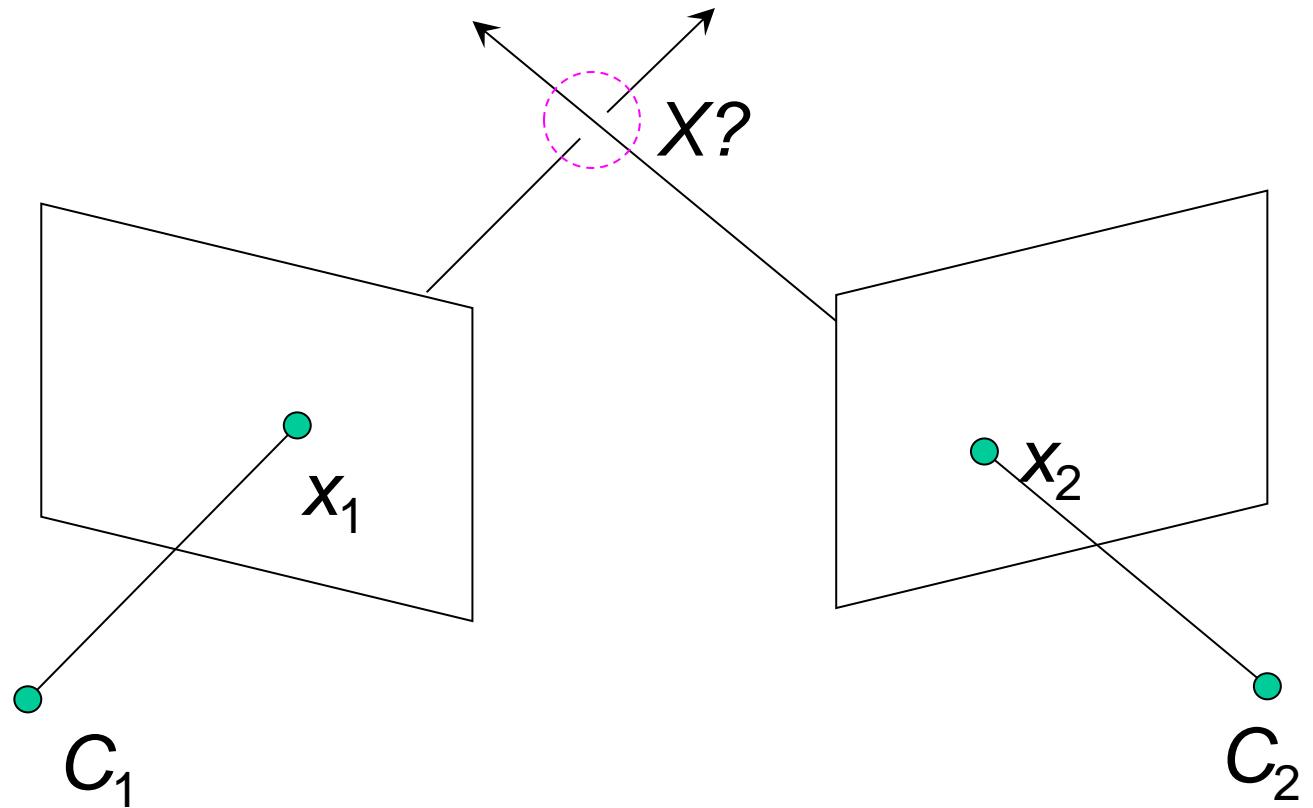


# Триангуляция

# Триангуляция («структура» сцены)



Даны проекции  $x_1$ ,  $x_2$  точки  $\mathbf{X}=(X,Y,Z)$  на 2 или более изображения (с известными матрицами калибровки), найти координаты точки





# DLT метод

---

- $P$  и  $(x,y)$  известны, нужно найти  $(X,Y,Z)$
- Уравнение перспективной проекции:

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

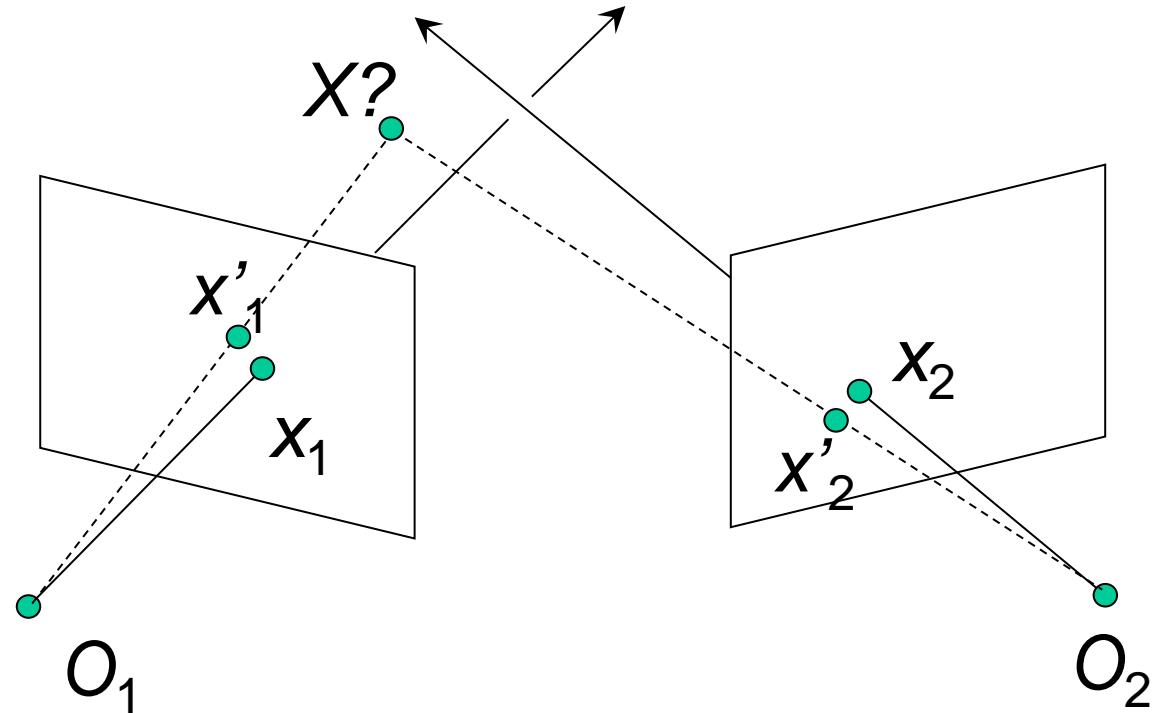
- Для 1 камеры получаем 2 уравнения на  $(X,Y,Z)$
- Т.е. нужны минимум 2 камеры для оценки 3Д координат точки линейным методом
- Дальше всё как обычно, сведение к МНК



# Метод «золотого стандарта»

Найти точку  $X$ , минимизирующую ошибку репроекции:

$$d^2(x_1, P_1 X) + d^2(x_2, P_2 X)$$

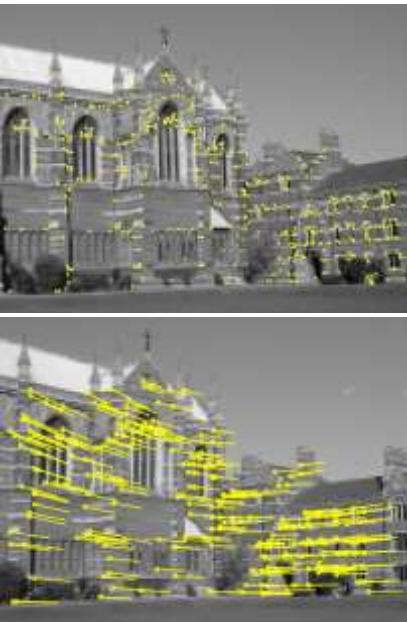
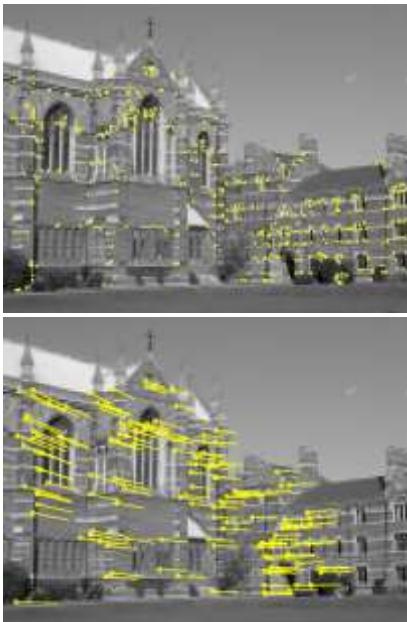


- Результат DLT используем как начальное приближение
- Требует применения нелинейной оптимизации

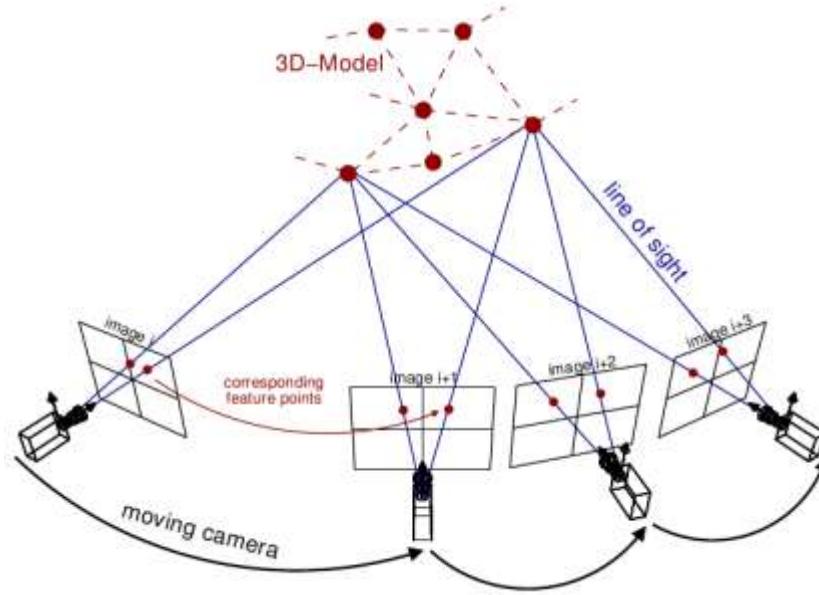


# Задача структуры и движения

# Разреженная 3D реконструкция



Point features and matching



Camera localization and structure estimation

- Опираемся на сопоставление ключевых точек между изображениями
- 3D реконструкция в форме облака точек, соответствующих найденным ключевым точкам на изображениях
- Решаем совместно определение 3Д точек, соответствующих ключевым, и положение камер в пространстве относительно них
- 3D реконструкция – обратная задача к задаче компьютерной графики

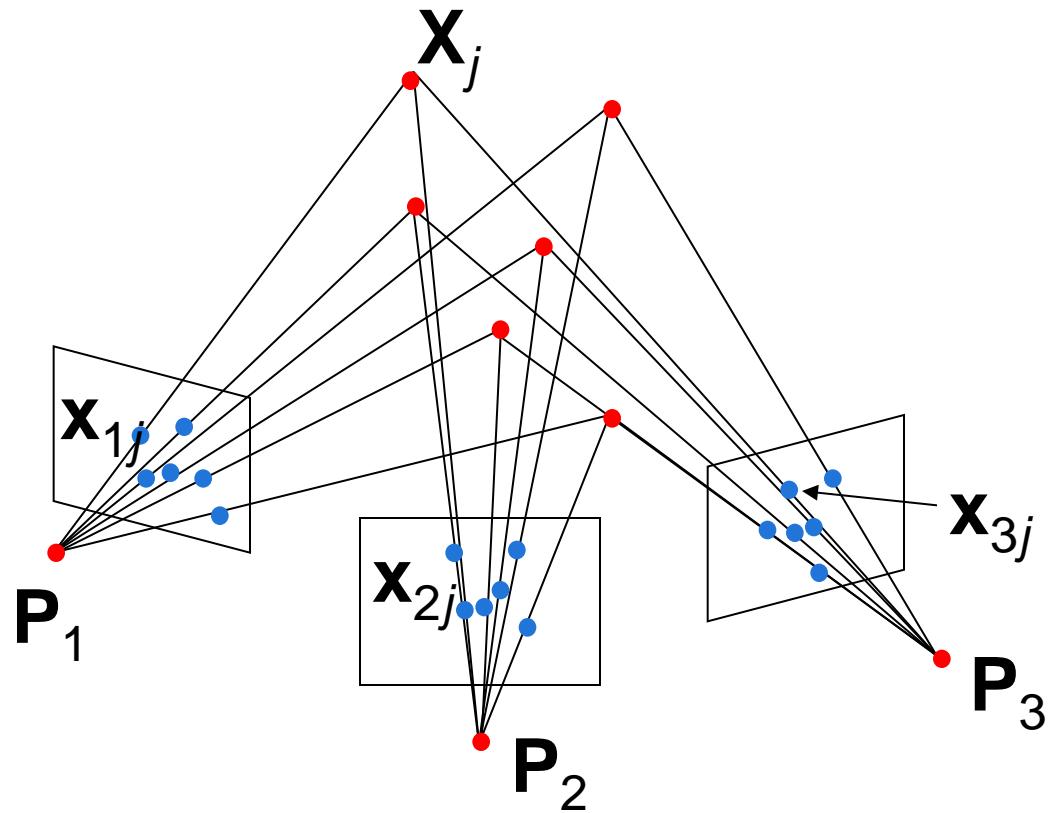


# Формулировка задачи

- Дано:  $m$  изображений  $n$  фиксированных 3D точек

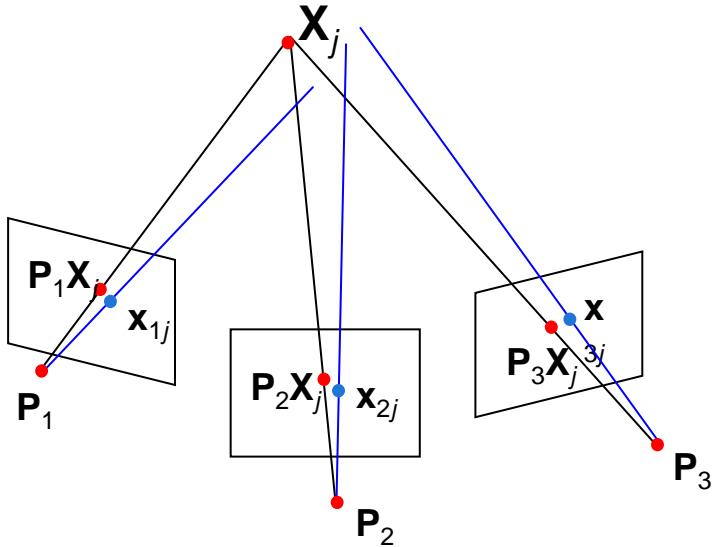
$$\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

- Задача: оценить  $m$  - матриц проекции  $\mathbf{P}_i$  и  $n$  - 3D точек  $\mathbf{X}_j$  из  $mn$  соответствий  $\mathbf{x}_{ij}$





# Целевая функция



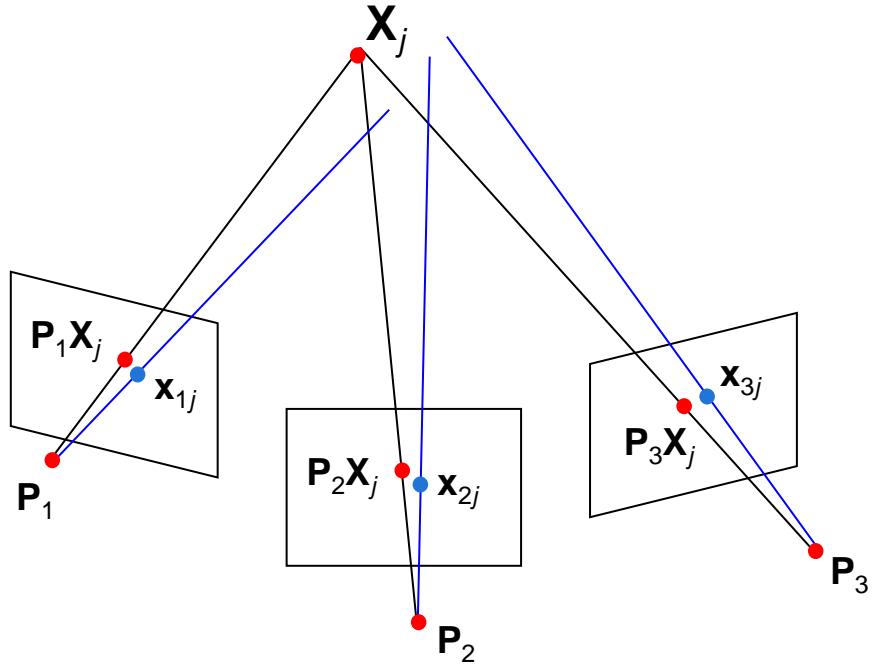
Оптимальная целевая функция – сумма ошибок репроекций 3D точек  $X$  на все изображения:

$$E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^m \sum_{j=1}^n D(\mathbf{x}_{ij}, \mathbf{P}_i \mathbf{X}_j)^2$$

- Нелинейная целевая функция с большим количеством параметров
  - $3n$  точек
  - $6m$  параметров для калиброванного случая,  $11m$  для некалиброванного
- Решается градиентным спуском, но плохо – масса локальных минимумов
- Поэтому нужно получить хорошее начальное приближение, в этом основная проблема



# Калибранный случай



Оптимальная целевая функция – сумма ошибок репроекций 3D точек  $X$  на все изображения:

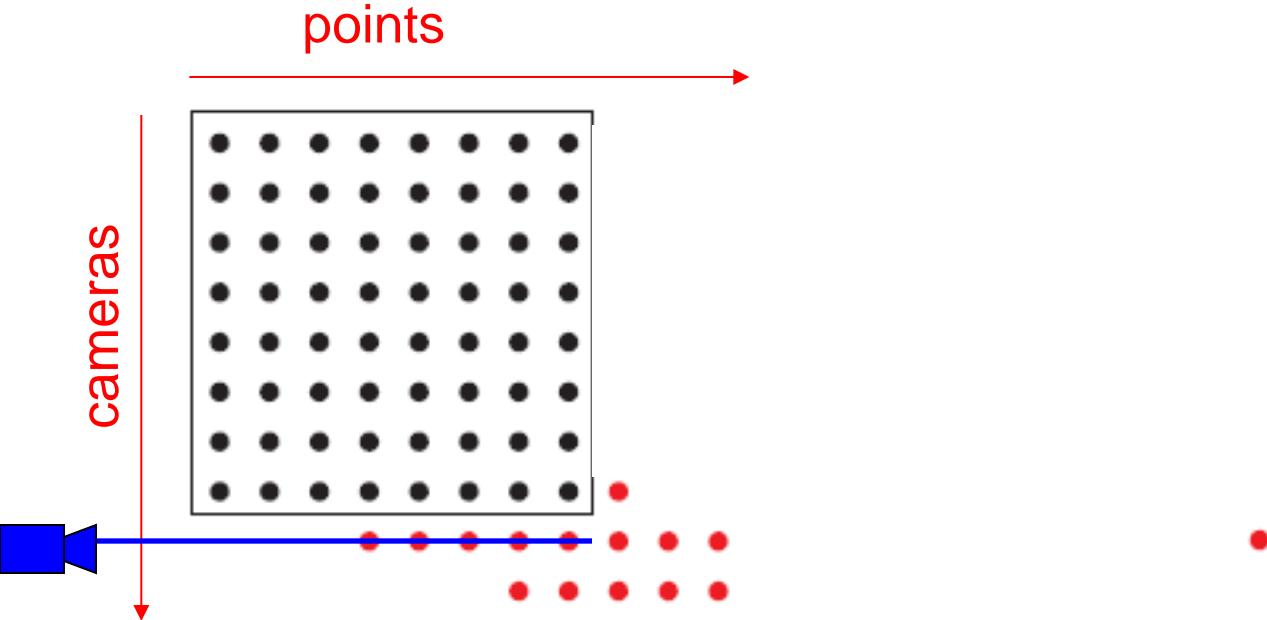
$$E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^m \sum_{j=1}^n D(\mathbf{x}_{ij}, \mathbf{P}_i \mathbf{X}_j)^2$$

- Будем рассматривать случай, когда известны внутренние калибровки всех камер  $K_i$
- Тогда мы можем вычислять существенную матрицу между двумя видами по соответствиям, т.е. оценивать напрямую внешнюю калибровку камеры
- Составим алгоритм для получения начального приближения и уточним результат градиентным спуском

# Последовательный подход



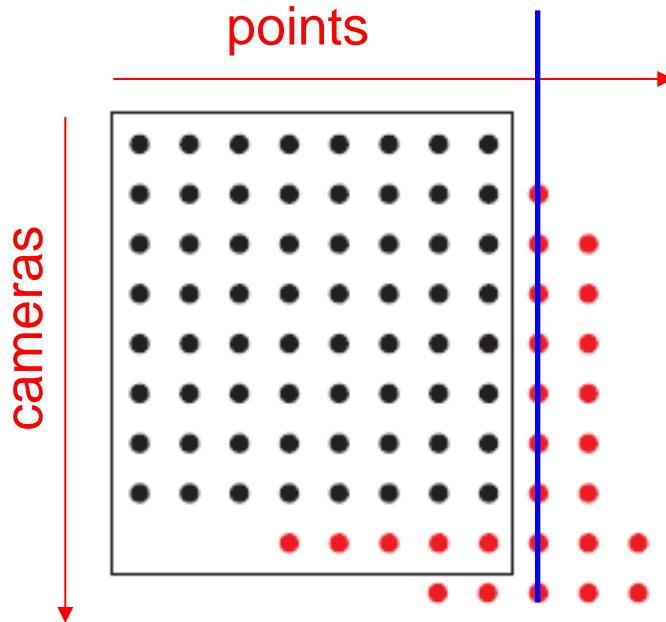
- Выберем какие-то 2 камеры
- Инициализируем движение через существенную матрицу двух камер
- Инициализируем структуру
- Для каждого следующего вида:
  - Оцениваем матрицу проекции новой камеры по всем известным 3D точкам, видимым на этом изображении – калибровка





# Последовательный подход

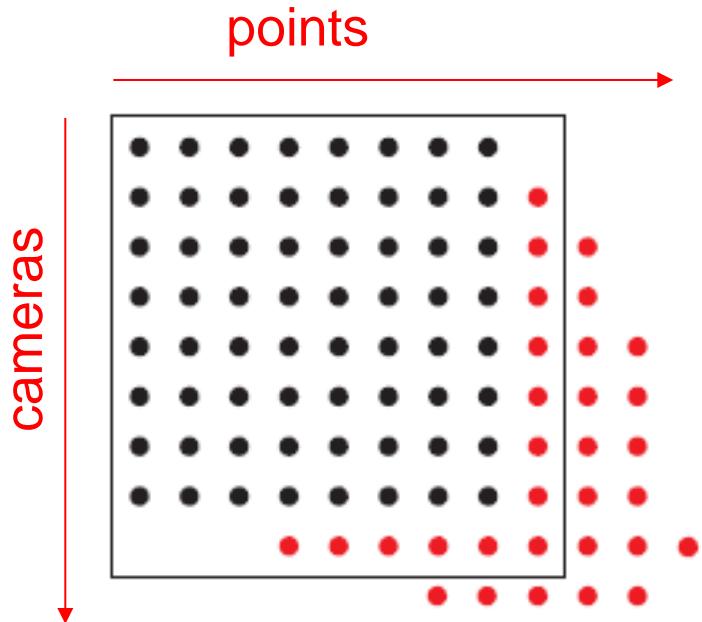
- Выберем какие-то 2 камеры
- Инициализируем движение через существенную матрицу двух камер
- Инициализируем структуру
- Для каждого следующего вида:
  - Оцениваем матрицу проекции новой камеры по всем известным 3D точкам, видимым на этом изображении – калибровка
  - Уточняем и дополняем структуру, вычисляем новые 3D точки, уточняем существующие точки, видимые на камере - триангуляция





# Последовательный подход

- Выберем какие-то 2 камеры
- Инициализируем движение через существенную матрицу двух камер
- Инициализируем структуру
- Для каждого следующего вида:
  - Оцениваем матрицу проекции новой камеры по всем известным 3D точкам, видимым на этом изображении – калибровка
  - Уточняем и дополняем структуру, вычисляем новые 3D точки, уточняем существующие точки, видимые на камере - триангуляция
  - Уточняем структуру и в движение методом связок

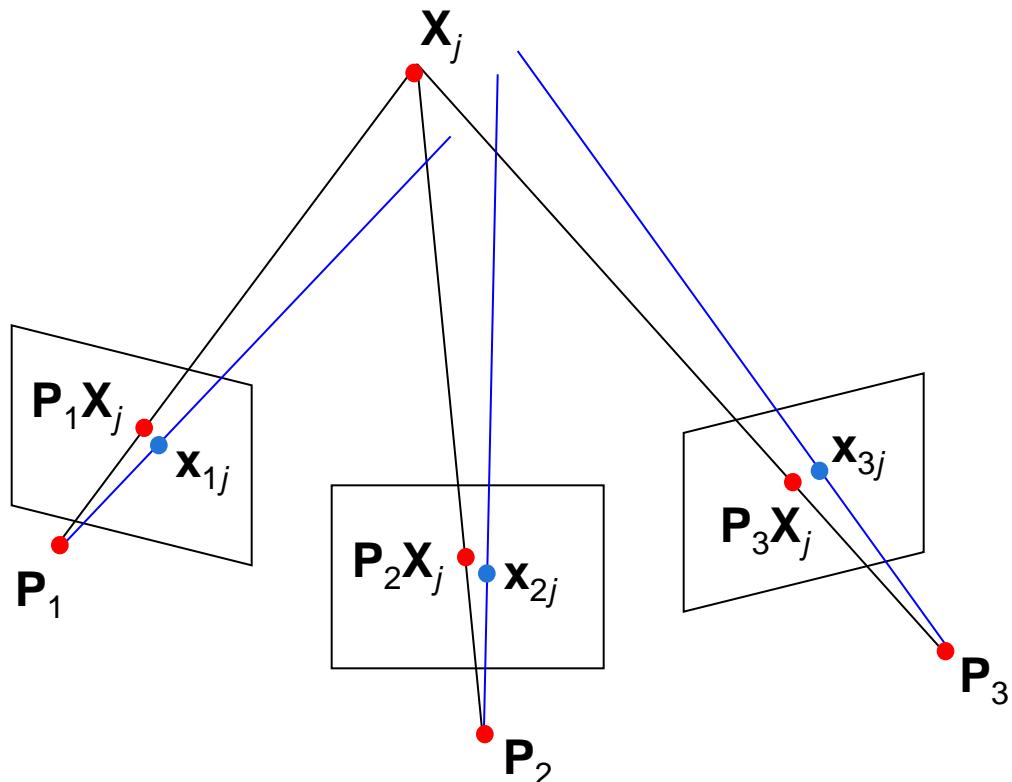


# Метод связок (Bundle adjustment)



- Нелинейный метод для уточнения структуры и движения
- Минимизируем сумму ошибок проекций всех точек на все камеры:

$$E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^m \sum_{j=1}^n D\left(\mathbf{x}_{ij}, \mathbf{P}_i \mathbf{X}_j\right)^2$$





# Неоднозначность решения

---

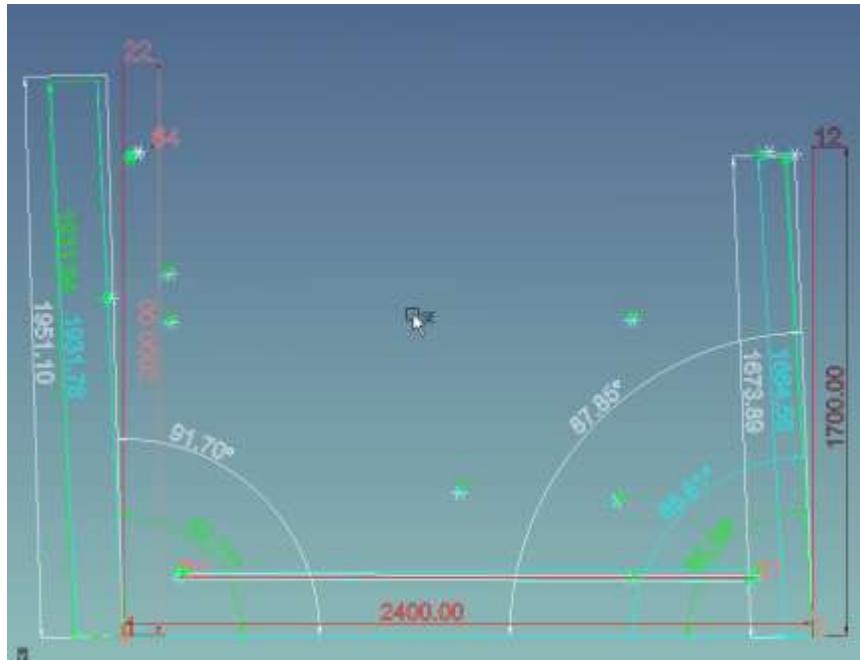
- Если мы умножим всю сцену на некоторый коэффициент  $k$  и в то же время умножим матрицы камер на  $1/k$ , проекции точек сцены на изображения не изменятся:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \left(\frac{1}{k} \mathbf{P}\right)(k\mathbf{X})$$

Вывод: оценить абсолютный масштаб (размеры) сцены только по изображениям невозможно!

Нужна дополнительная информация

# Фотограмметрия



Чтобы провести измерения по изображениям (задача фотограмметрии), необходимо знать расстояние между минимум 2мя точками для нормализации сцены

# Резюме

---



- Многовидовая геометрия позволяет установить взаимосвязи между точками разных изображений, параметрами камер и 3D точками сцены
- При наблюдении сцены с 2x ракурсов возникает эпиполярная геометрия
- Взаимосвязи эпиполярной геометрии можно описать фундаментальной и существенной матрицей
- Существенная матрица зависит от движения камеры, её оценка позволяет оценить движение между ракурсами
- Гомография описывает вырожденные случаи движения камеры
- Для гомографии предложили нейросетевые модели, позволяющие предсказывать параметры гомографии по 2м кадрам
- Триангуляция позволяет оценить структуру сцены
- Общая задача структуры и движения может решаться итеративным методом, через последовательное решение задач оценки движения и структуры
- Совместное уточнение структуры и движения называется методом связок (Bundle Adjustment)