

Школа Анализа
Данных Яндекса

Курс «Анализ изображений и видео, ч.2»

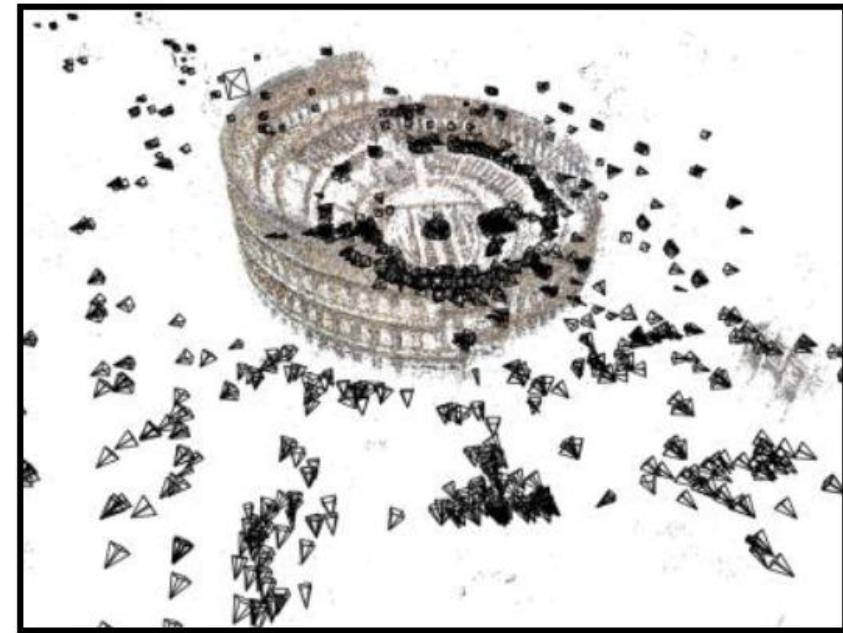
Лекции №8-9
«Структура из движения (SfM)»

Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа
ВМК МГУ

7 и 14 апреля 2017 года

Большая задача

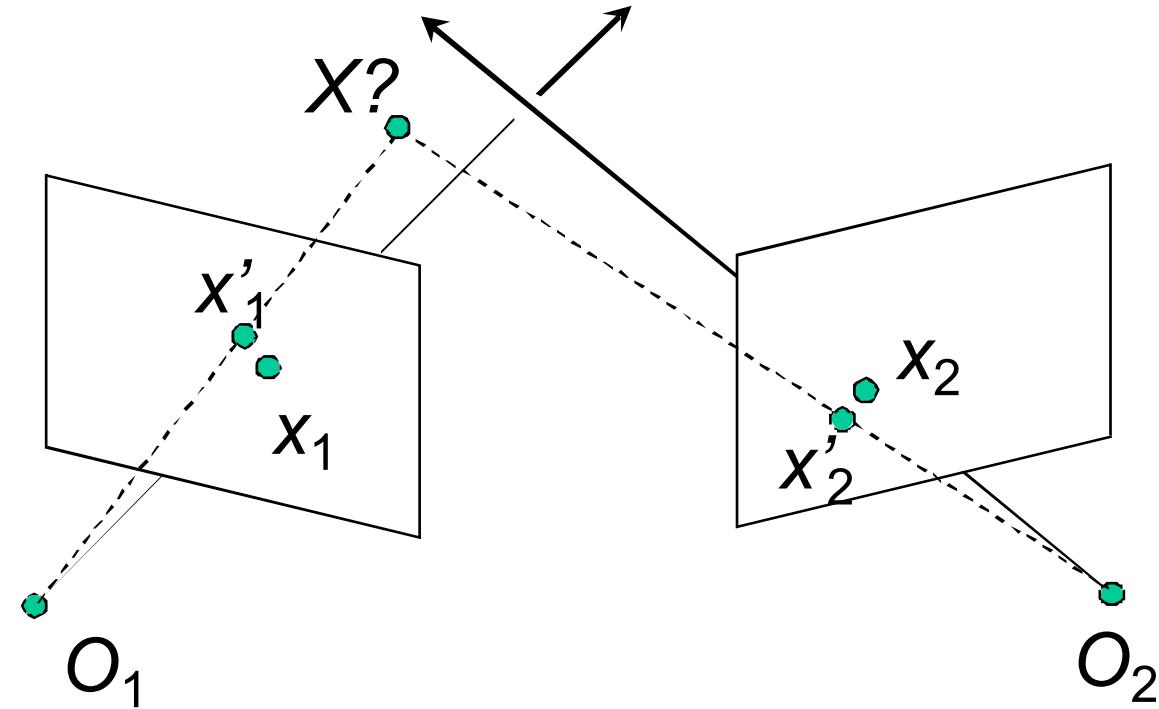


Коллекция изображений
из интернета

Трёхмерная модель

В 24 часа...

Что мы можем сделать?



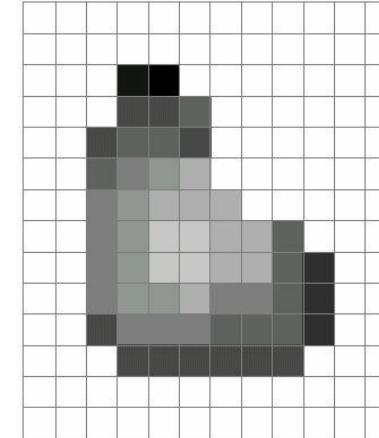
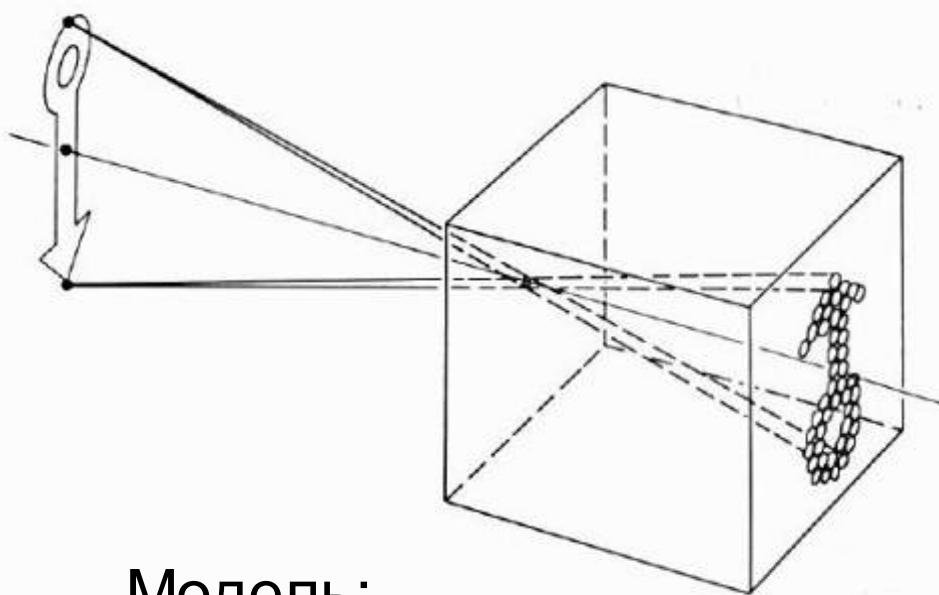
Стереозрение или триангуляция

План лекций



- Перспективная камера
 - Модель перспективной камеры
 - Отличия от реальной камеры
- Многовидовая геометрия
 - Гомография
 - Эпиполярная геометрия
- «Структура из движения» и подзадачи
- Применение – строим Рим за 24 часа

Модель камеры-обскуры

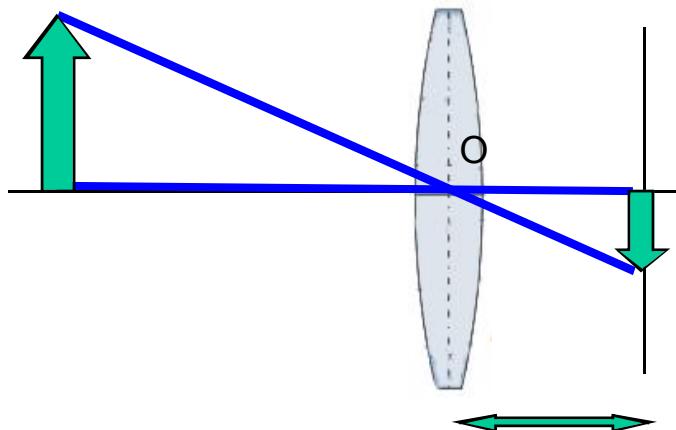


+ дискретное изображение
на метрице

Модель:

- В препадре отверстие размеров в одну точку
- Все лучи проходят через одну точку
- Эта точка называется **Центром Проекции (ЦП)**
- Реальный размер отверстия – **Апертура**
- Изображение формируется на **Картинной плоскости**
- **Фокусным расстоянием f** называется расстояние от ЦП до Картинной плоскости

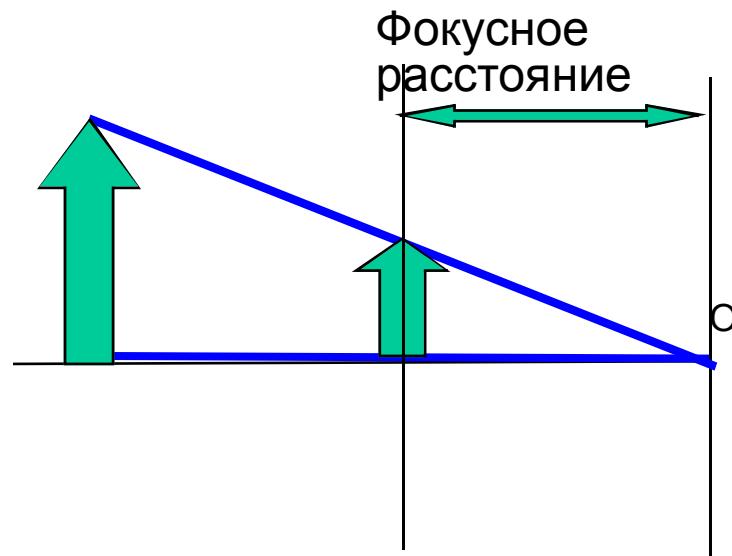
Построение модели



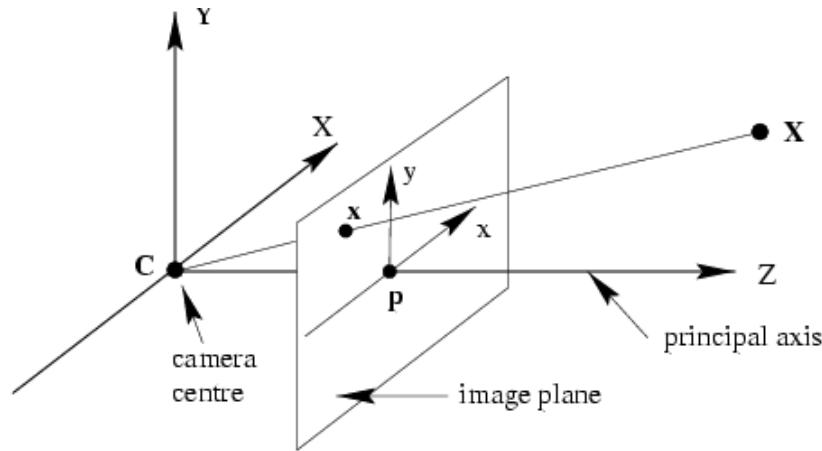
- Камера-обскура
- Изображение позади фокуса
- Изображение перевернутое

Фокусное расстояние

- Модель перспективной проекции
- Перенесем объект на противоположенную сторону
- То же самое фокусное расстояние!
- Изображение нормальное, не перевернутое



Модель камеры-обскуры

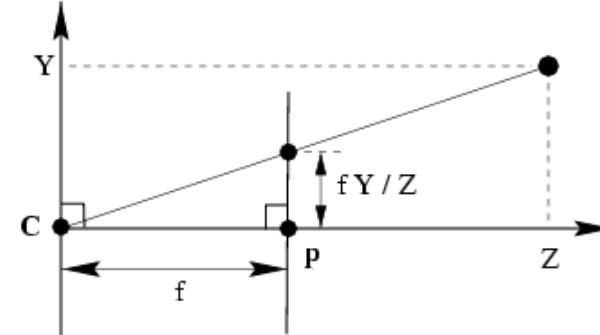


$C = (0,0,0)$ – центр камеры (проекции)

$X = (X, Y, Z)$ – точка в 3х мерном пространстве

$x = (x, y)$ – проекция X на изображении

C, x, X лежат на одной прямой



Тогда простейшее уравнение перспективной проекции:

$$x = f \frac{X}{Z} \quad y = f \frac{Y}{Z}$$

Перспективная проекция – нелинейное преобразование!

Однородные координаты



- Для удобства хотим записывать перспективную проекцию и другие преобразования в матричном виде
- Для этого нужно перейти от обычных координат в однородные:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Однородные координаты
точки изображения

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Однородные координаты
точки сцены

Перевод из однородных в обычные:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

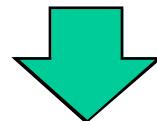
Однородные координаты активно используются в проективной геометрии, но нам они только ограниченно понадобятся

Аффинные преобразования



В однородных координатах удобно записывать разные преобразования:

$$X \rightarrow AX + T_x$$



$$\begin{bmatrix} aX + T_x \\ bY + T_y \\ cZ + T_z \\ 1 \end{bmatrix} \underset{\approx}{=} \begin{bmatrix} a & 0 & 0 & T_x \\ 0 & b & 0 & T_y \\ 0 & 0 & c & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Р в матричной форме



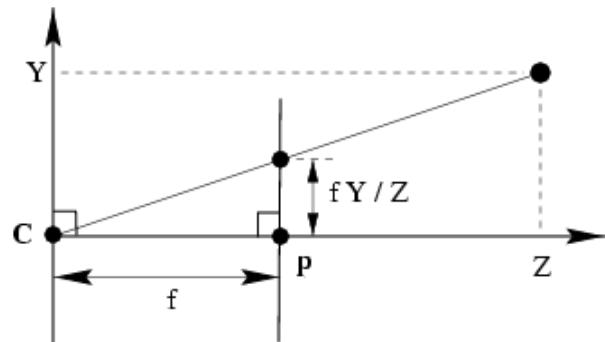
Запишем теперь перспективную проекцию в матричном виде:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} \quad x = \frac{X}{Z} \quad y = \frac{Y}{Z}$$

$$P = [I | 0]$$

$$\lambda x = P X$$

Внутренняя калибровка (f)



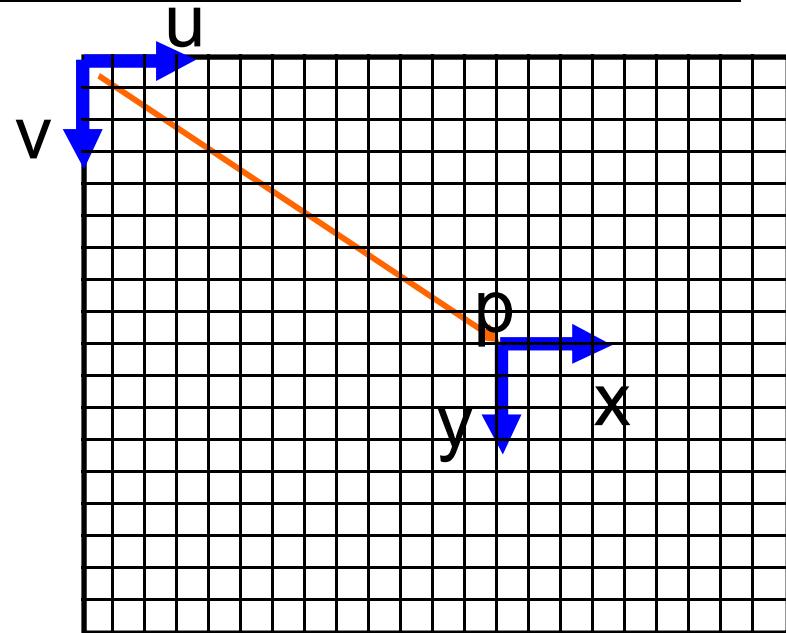
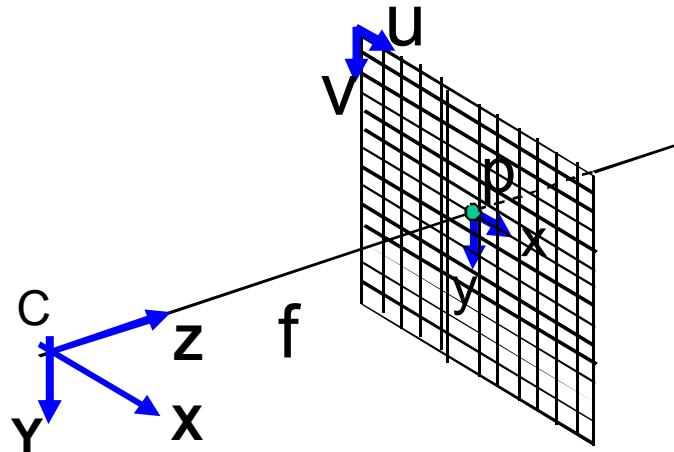
$$x = f \frac{X}{Z} \quad y = f \frac{Y}{Z}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \rightarrow P = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$\underbrace{\hspace{10em}}$ $\underbrace{\hspace{10em}}$

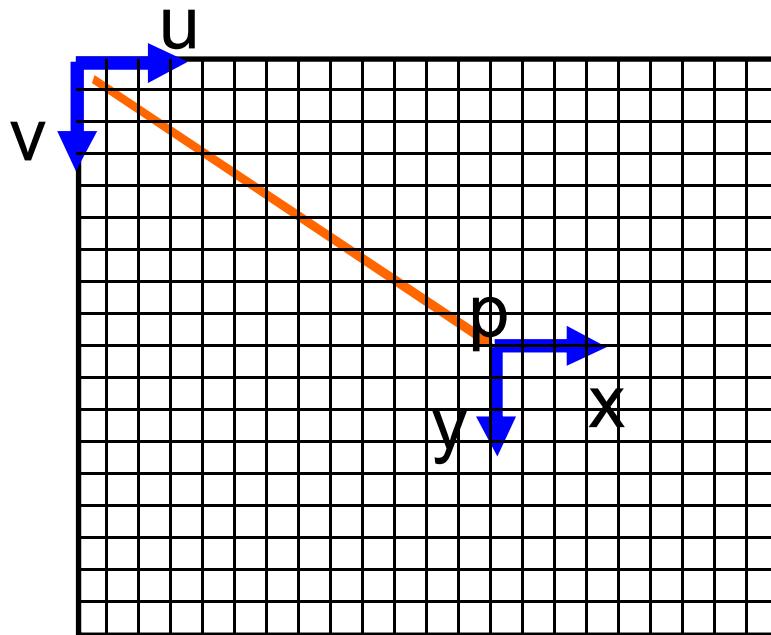
\mathbf{K} $[\mathbf{I} | \mathbf{0}]$

Внутренняя калибровка



- p – основание перпендикуляра из C на картинную плоскость (принципиальная точка)
- $p = (0,0, f)$ в мировых координатах, центр системы координат в картинной плоскости
- В картинной области расположена матрица (изображение)
- Нужно преобразовать из мировых координат в пиксельные

Внутренняя калибровка



- Для перевода в координаты изображения в пикселях нужно сделать масштабирование и сдвиг

$$(x, y) \rightarrow (sx + c_x, sy + c_y) \cong (x / \text{pix} + c_x, y / \text{pix} + c_y)$$

Где pix – размер пикселя, (c_x, c_y) – принципиальная точка в координатах изображения

Внутренняя калибровка



$$(x, y) \rightarrow (sx + c_x, sy + c_y) \cong (x / \text{pix} + c_x, y / \text{pix} + c_y)$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \cong \begin{bmatrix} 1 / \text{pix} & 0 & c_x \\ 0 & 1 / \text{pix} & c_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x / \text{pix} + c_x \\ y / \text{pix} + c_y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 / \text{pix} & 0 & c_x \\ 0 & 1 / \text{pix} & c_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cong \begin{bmatrix} f / \text{pix} & 0 & c_x \\ 0 & f / \text{pix} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

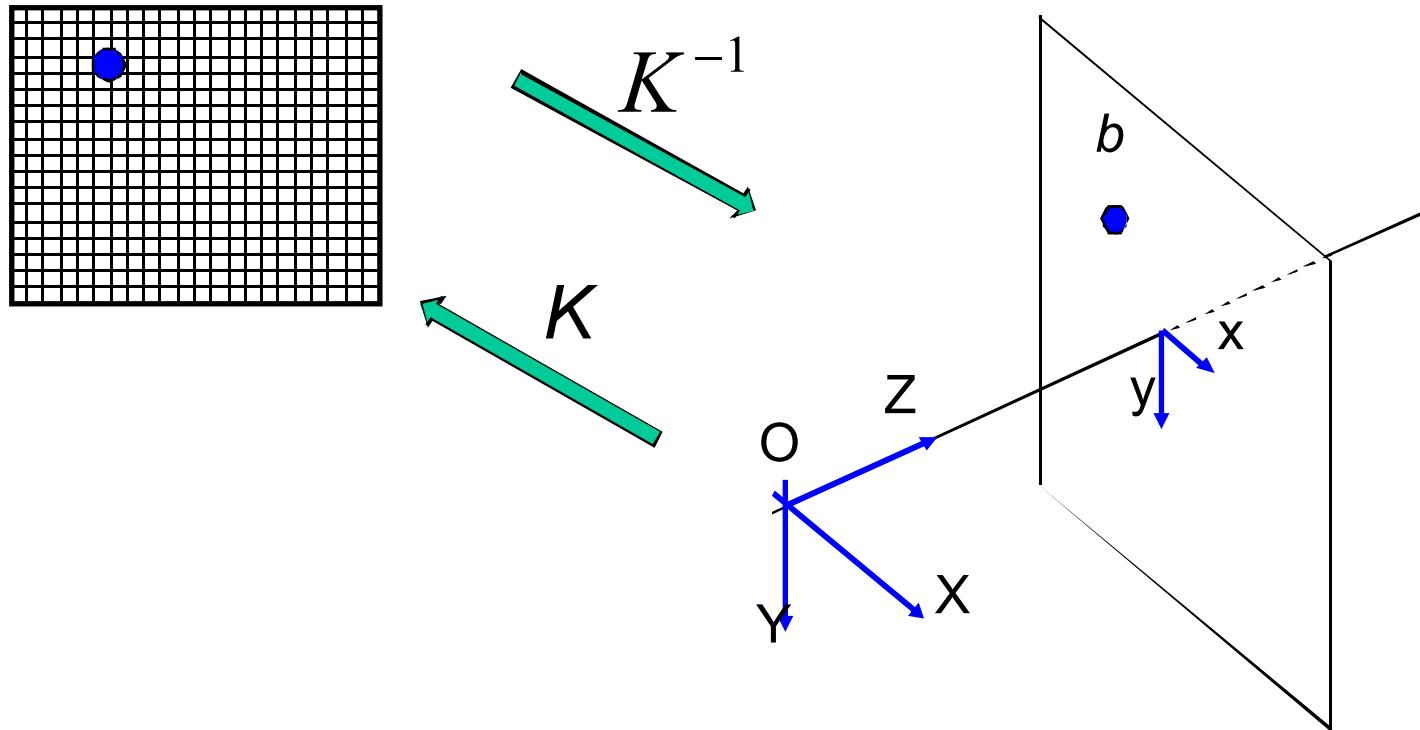
Внутренняя калибровка



Или матрица калибровки камеры (camera calibration matrix), internal calibration

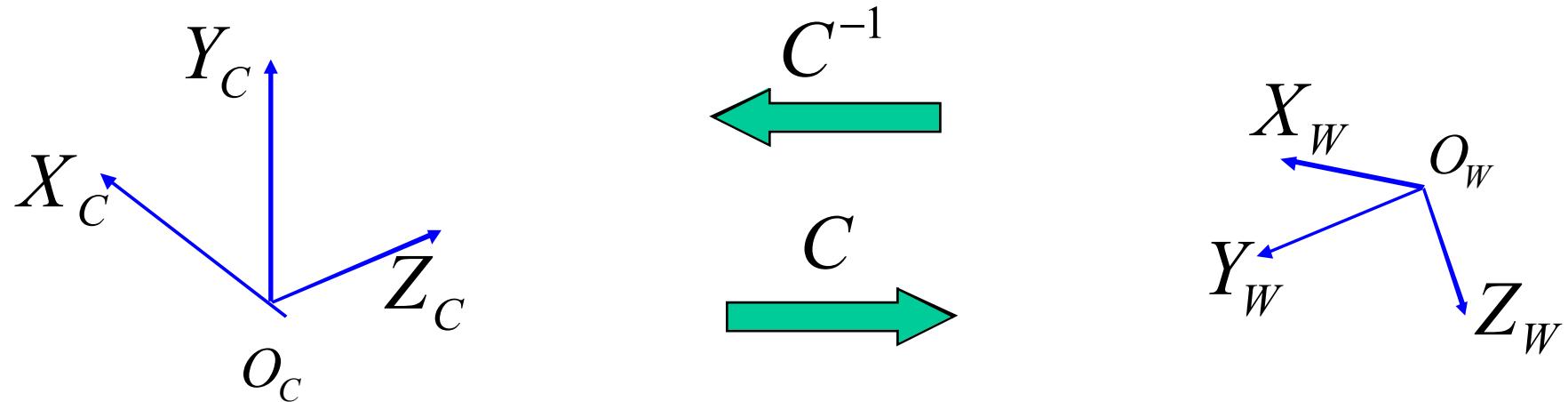
$$K = \begin{bmatrix} a_x & s & c_x \\ 0 & a_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f / \text{pix} & s & c_x \\ 0 & f / \text{pix} & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

- a – масштабирующие коэффициенты
- s – наклон, угол между осями x и y изображения
- c – положение принципиальной точки изображения (в пикселях)
- f, pix – фокусное расстояние, размер пикселя



- Внутренняя калибровка K отображает точки с картинной плоскости (в координатах камер) на изображение (в пиксели)
- Обратная матрица K^{-1} отображает точки изображения (в пикселях) на картинную плоскость (в координаты камеры, с $f=1$)
 - Координаты в пространстве с центром в C будут $(x, y, 1, 1)$
 - Отображение точек на изображении в лучи

Из мировой в координаты камеры



$$C = \begin{bmatrix} R & T \\ [0,0,0] & 1 \end{bmatrix}$$

- Положение и ориентация камеры в мировых координатах задается евклидовым преобразованием С
- Обратное преобразование
 - Из мировых координат в координаты камеры
 - Обратное к С преобразование! – $\text{Inverse}(C)$

Внешняя калибровка



- Матрица преобразования из мировой системы координат в систему координат камеры называется матрицей *внешней калибровки*

$$C^{-1} = \begin{pmatrix} R & T \\ [0,0,0] & 1 \end{pmatrix}^{-1} = \begin{bmatrix} R^T & -R^T T \\ [0,0,0] & 1 \end{bmatrix}$$

- Внешняя калибровка определяется положением и ориентацией камеры в пространстве

Конечная проективная камера



$$P = K \cdot [I \mid 0] \cdot C^{-1}$$

Матрица проекции камеры

Внутренняя калибровка

Центральная проекция

Внешняя калибровка

- Матрица P размером 3×4 , которую можно факторизовать вышеуказанным образом, где K – верхнетреугольная, называется **конечной проективной камерой** (*finite projective camera*)
 - У неё 11 степеней свободы (5 внутренних параметров, 3 параметра поворота и 3 параметра положения центра проекции)

Обобщенная проективная камера



- Произвольная однородная матрица ранга 3 называется *обобщенной проективной камерой* (*general projective camera*)

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \cong \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{12} & p_{33} & p_{34} \end{bmatrix} * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \lambda x = PX$$

- У P в общем случае также 11 степеней свободы (неизвестных параметров)
- Если ранг < 3 , тогда всё проецируется в линию или точку, а не плоскость, т.е. изображения не получается

План лекций



- Перспективная камера
 - Модель перспективной камеры
 - Некоторые свойства реальной камеры
- Многовидовая геометрия
 - Гомография
 - Эпиполярная геометрия
- «Структура из движения» и подзадачи
- Применение – строим Рим за 24 часа

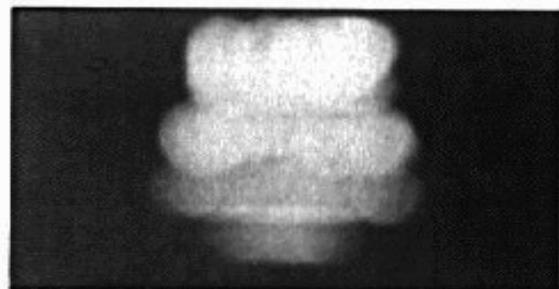
В домашних условиях



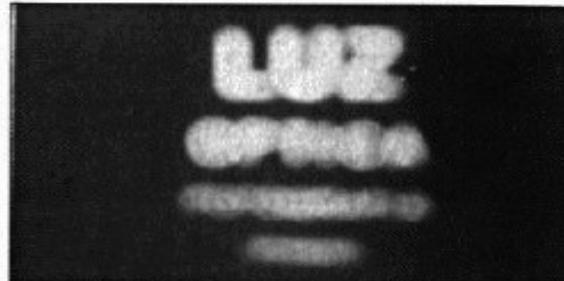
Почему
картина
такая
размытая?

<http://www.debevec.org/Pinhole/>

Уменьшаем апертуру



2 mm



1 mm



0.6mm

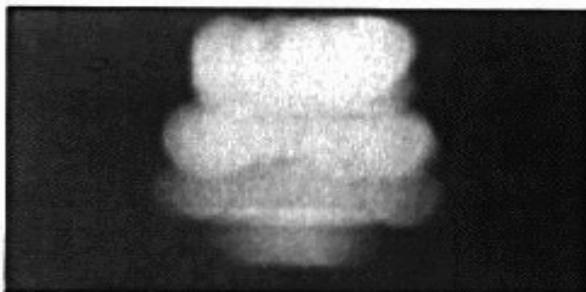


0.35 mm

Почему бы не сделать апертуру как можно меньше?

- Меньше апертура – меньше света проходит
- При малых апертурах начинаются дифракционные эффекты

Уменьшаем диафрагму



2 mm



1 mm



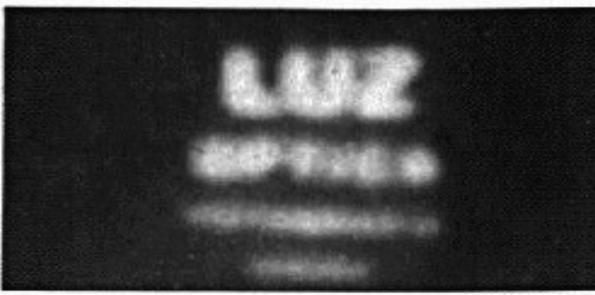
0.6mm



0.35 mm

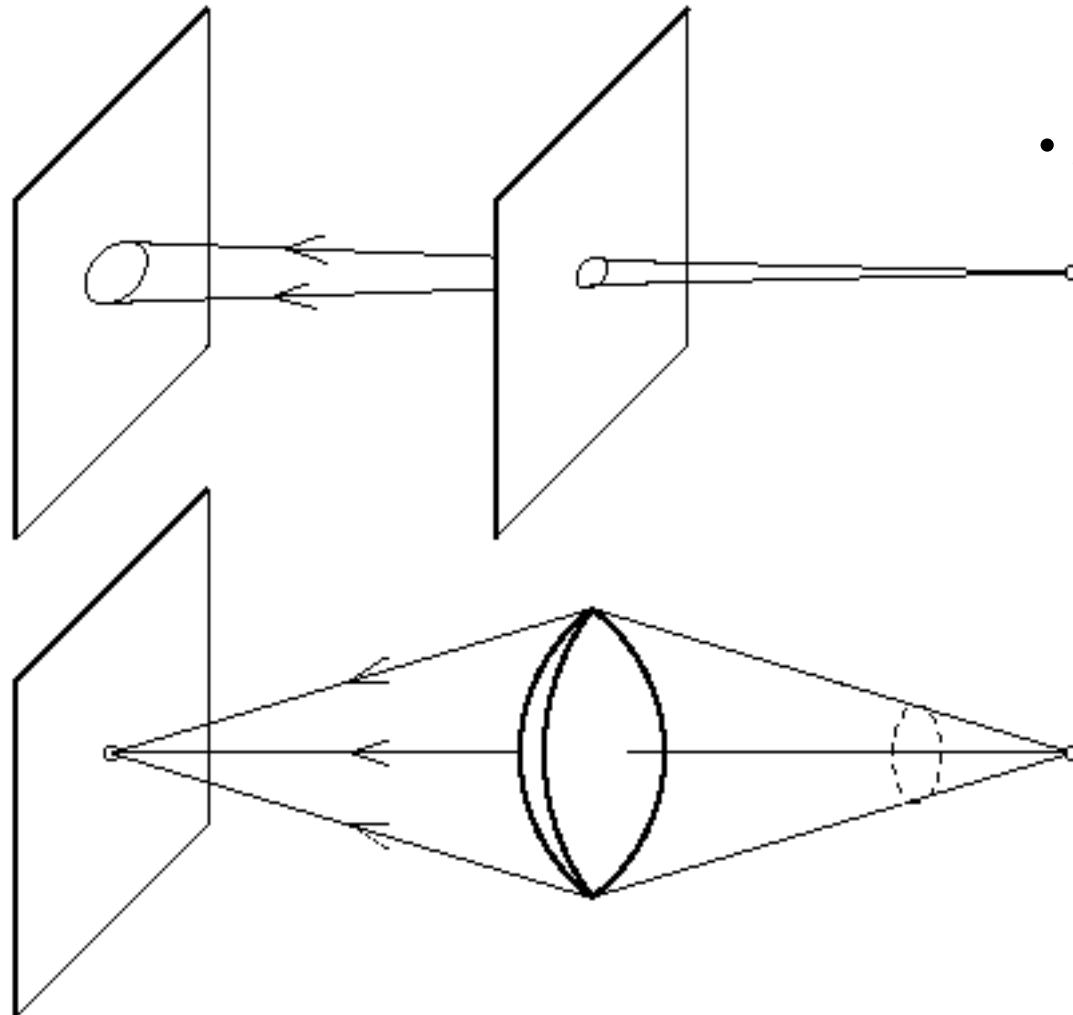


0.15 mm



0.07 mm

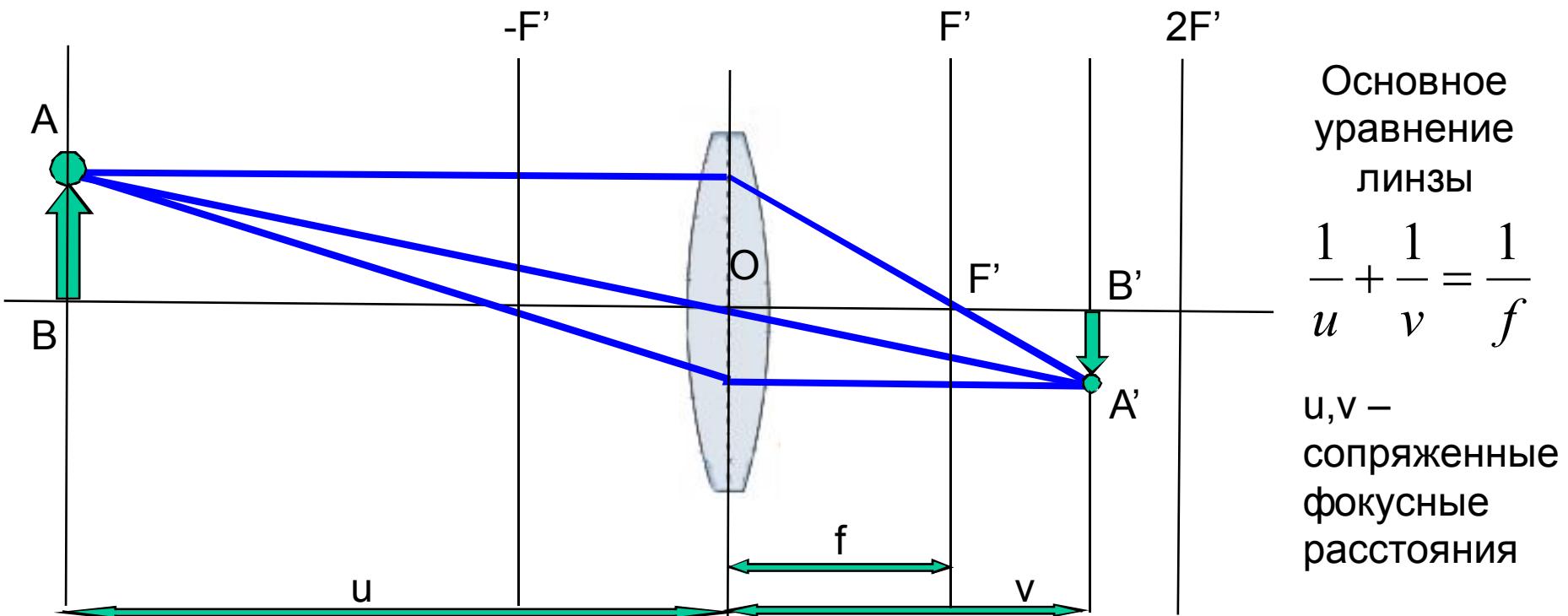
Линза!



- Линза:
диск из прозрачного однородного материала, ограниченный двумя полированными сферическими, или плоской и сферической, поверхностями

Линза позволяет увеличить поток света от каждой точки

Линза и уравнение линзы



- Лучи от одной точки объекта, преломляясь линзой, фокусируются в одной точке позади линзы
- Луч, проходящий через центр линзы не преломляется
- Система точно как камера-обскура, но собирает больше света
 - О – центр проекции (центр линзы)

Фотоаппарат

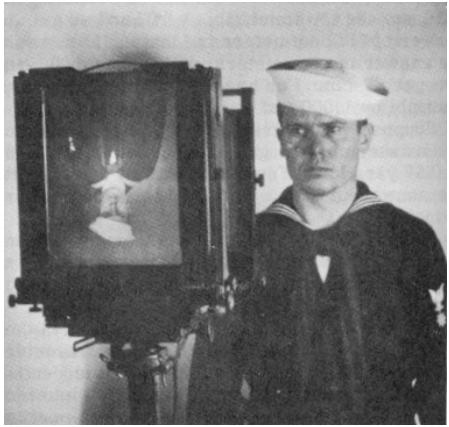
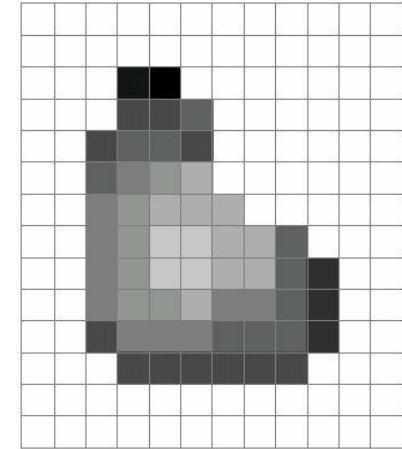
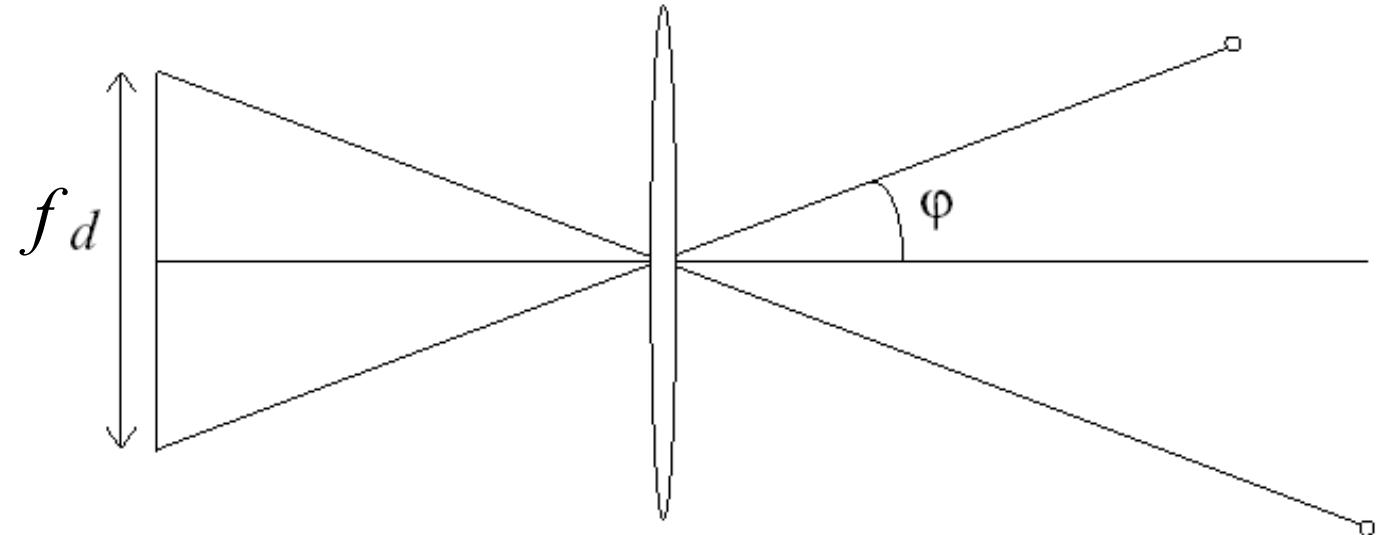
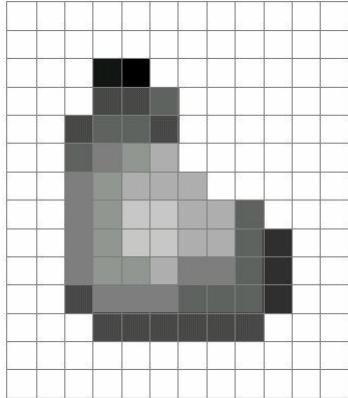


Figure from US Navy Manual of Basic Optics and Optical Instruments, prepared by Bureau of Naval Personnel. Reprinted by Dover Publications, Inc., 1969.



- Все фотоаппараты устроены на описанном принципе
- Вместо линзы более сложная оптическая система (объектив)
- Изображение формируется на плёнке или матрице

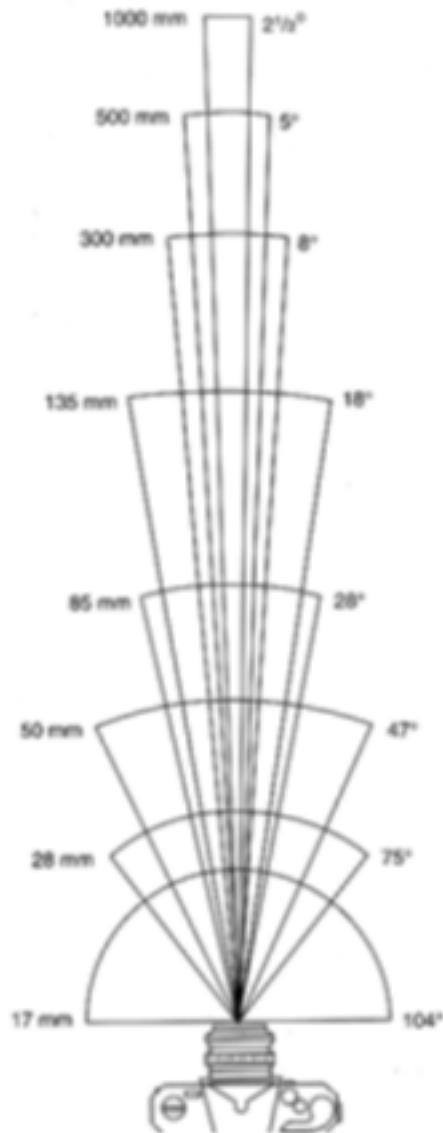
Угол обзора (Field of View)



$$\varphi = \tan^{-1}\left(\frac{d}{2f}\right)$$

- Размер матрицы ограничен – ограничен угол обзора
- Больше фокусное расстояние – меньше угол обзора
- Меньше фокусное расстояние – больше угол обзора

Трансфокация (Zoom)



17mm



28mm



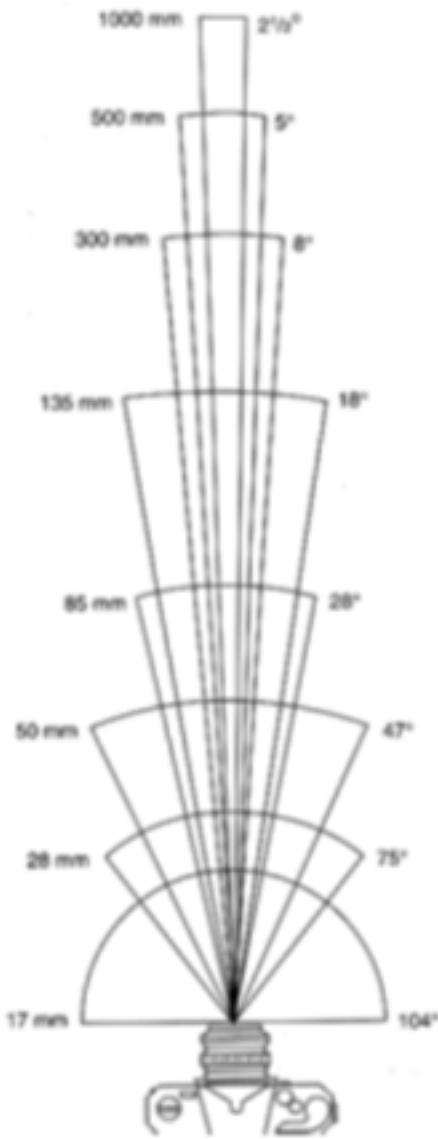
50mm



85mm

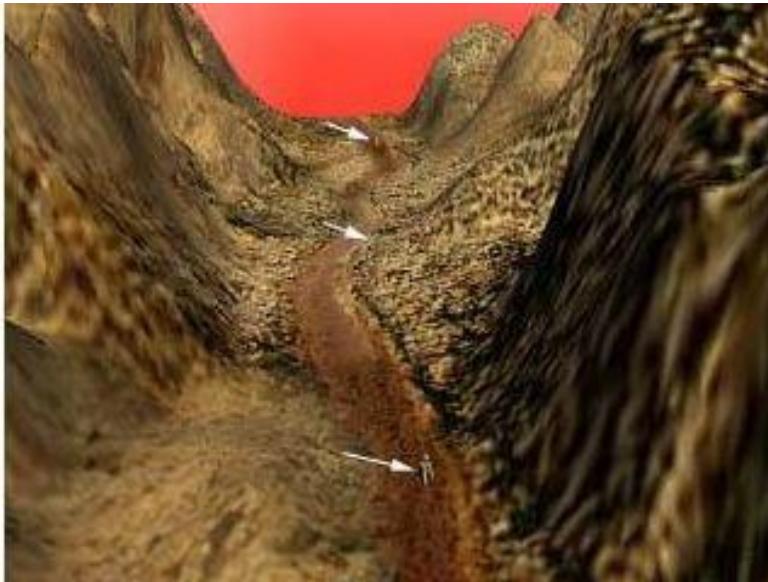
From London and Upton

Трансфокация (Zoom)



From London and Upton

Забавные эффекты



<http://www.hash.com/users/jsherwood/tutes/focal/Zoomin.mov>



Figure 5.1

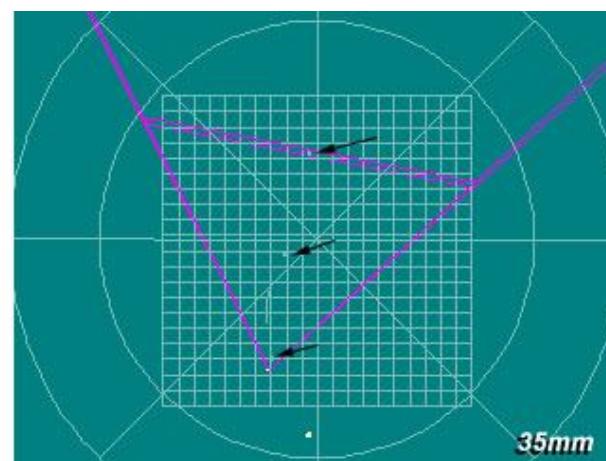


Figure 5.2

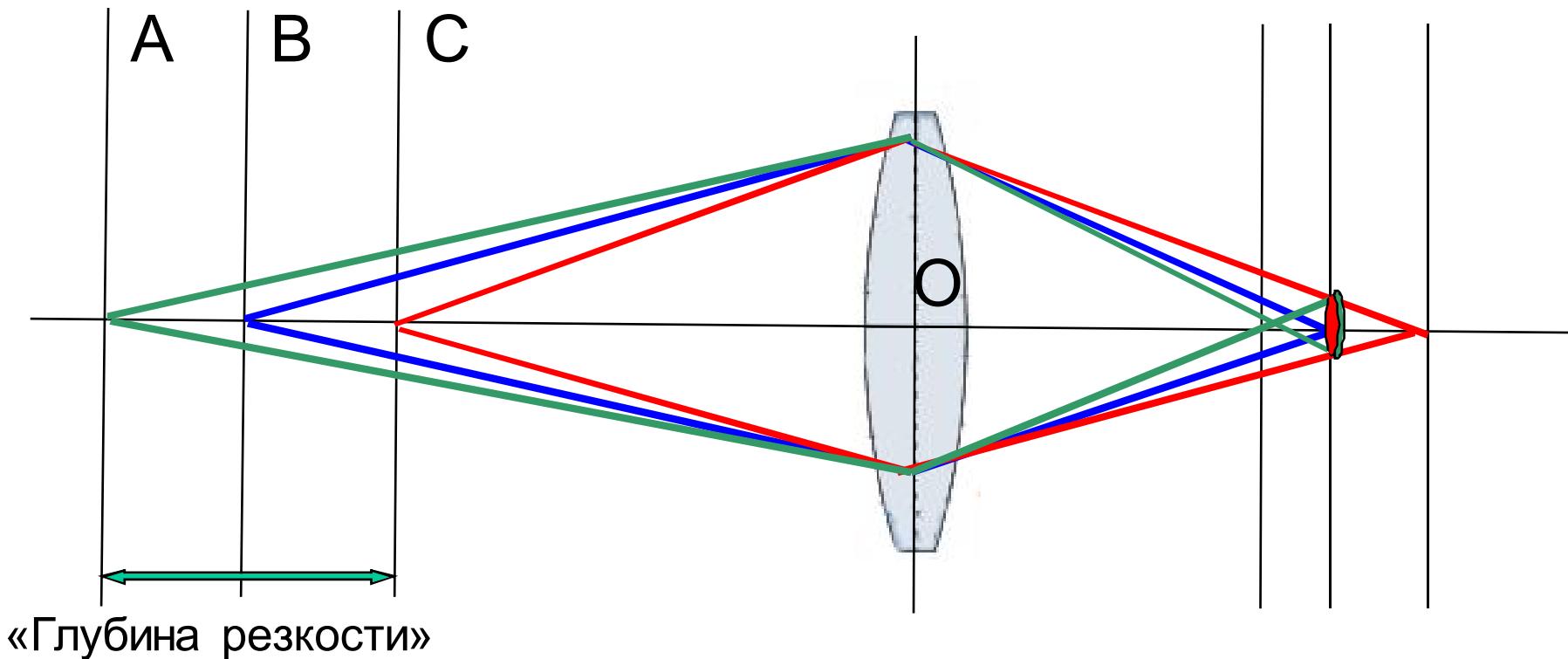
Эффект трансфокации



Эффект трансфокации



Глубина резкости (Depth of field)



«Глубина резкости»

- «Пятно рассеяния» маленькое – объект «в фокусе»
- Интервал в котором объекты попадают «в фокус» называется «глубиной резкости»

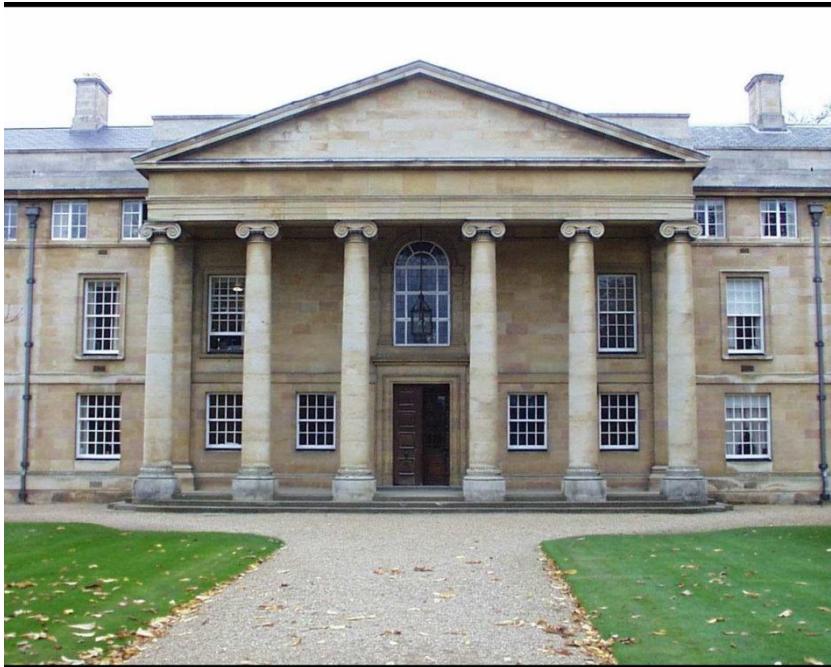


Глубина резкости



Глубину резкости можно использовать для оценки расстояния от объекта до камеры (Shape from focus)

Радиальная дисторсия



- Прямые линии по краям изображения превращаются в кривые
- Точки смещаются вдоль радиуса от принципиальной точки
- Модель дисторсии Tsai:

$$\hat{x} = x + L(r)x \quad r^2 = x^2 + y^2$$

$$\hat{y} = y + L(r)y \quad L(r) \approx k_1 r^2 + k_2 r^4$$

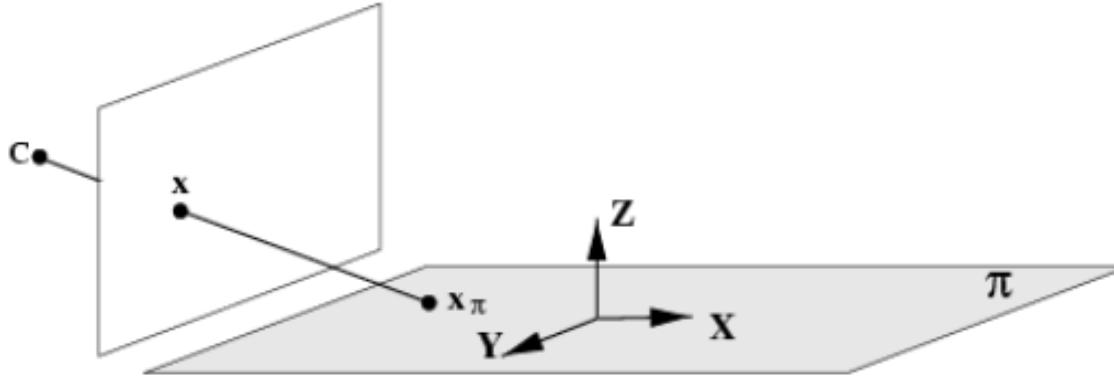
План лекций



- Перспективная камера
 - Модель перспективной камеры
 - Отличия от реальной камеры
- Многовидовая геометрия
 - Гомография
 - Эпиполярная геометрия
- «Структура из движения» и подзадачи
- Применение – строим Рим за 24 часа



Гомография



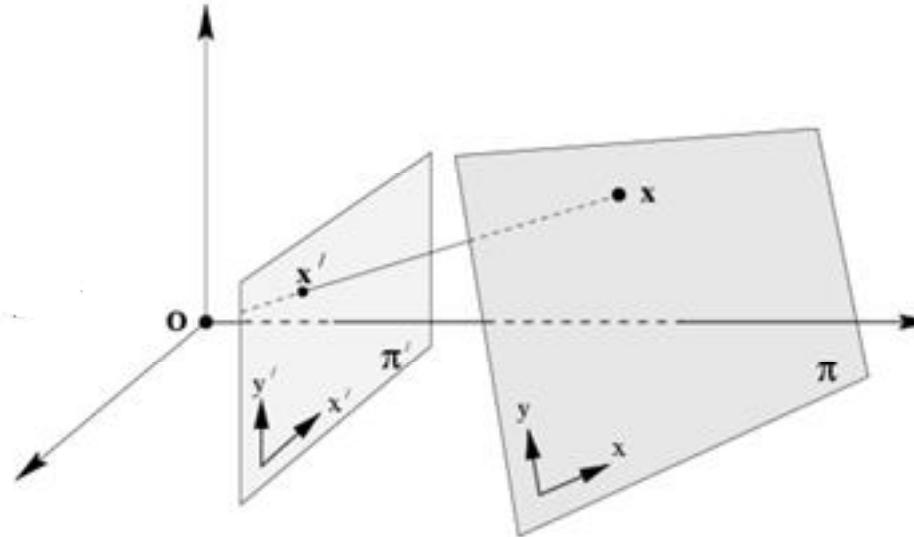
- Перспективное преобразование плоскости (гомография)
- Пусть точка X лежит на плоскости
- Выберем специальную мировую систему координат

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

Гомография

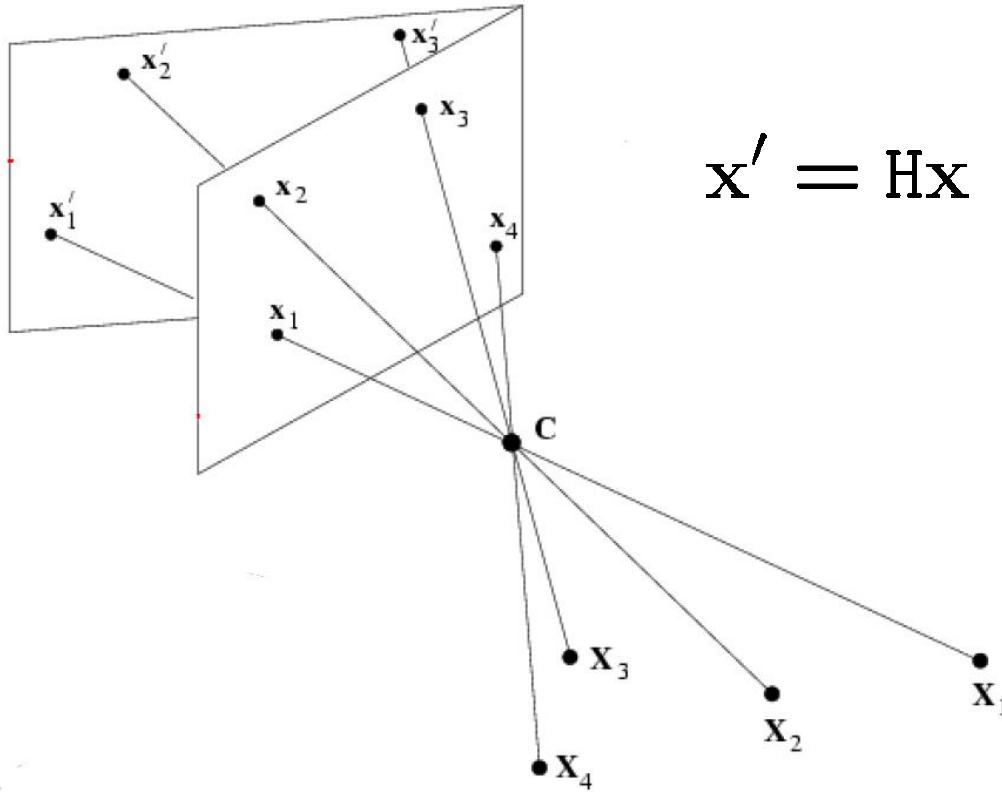


- Для произвольной плоскости аналогично (домножим на матрицу евклидова преобразования)
- 8 степеней свободы



$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} \simeq \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{12} & h_{33} \end{bmatrix}^* \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

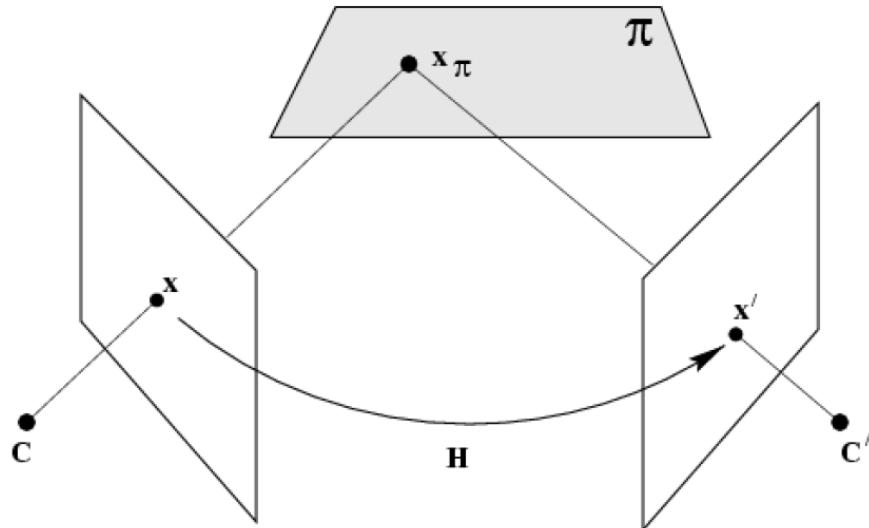
Стационарная камера



Если центр камеры не изменился, то одно изображение в другое преобразуется гомографией



Плоская сцена



$$x = H_{1\pi}x_\pi \quad x' = H_{2\pi}x_\pi$$

$$x' = H_{2\pi}x_\pi$$

$$= H_{2\pi}H_{1\pi}^{-1}x = Hx$$

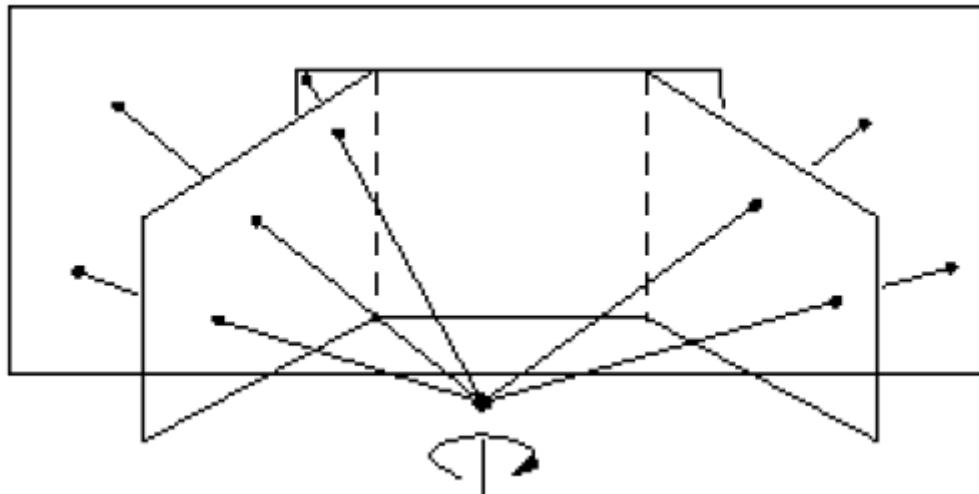
- Если наблюдаемая сцена – плоская, тогда преобразование точек между изображениями можно тоже описать гомографией
- Это результат используется в дальнейшем

Применение гомографии



- Выбираем 4 точки, которые являются вершинами прямоугольника с известными размерами
- Вычисляем гомографию (4е точки -> прямоугольник)
- Разворачиваем изображение
- Можем измерять расстояния на плоскости!

Построение мозаики (панорамы)



Проектируем все изображения на общую плоскость

План лекций



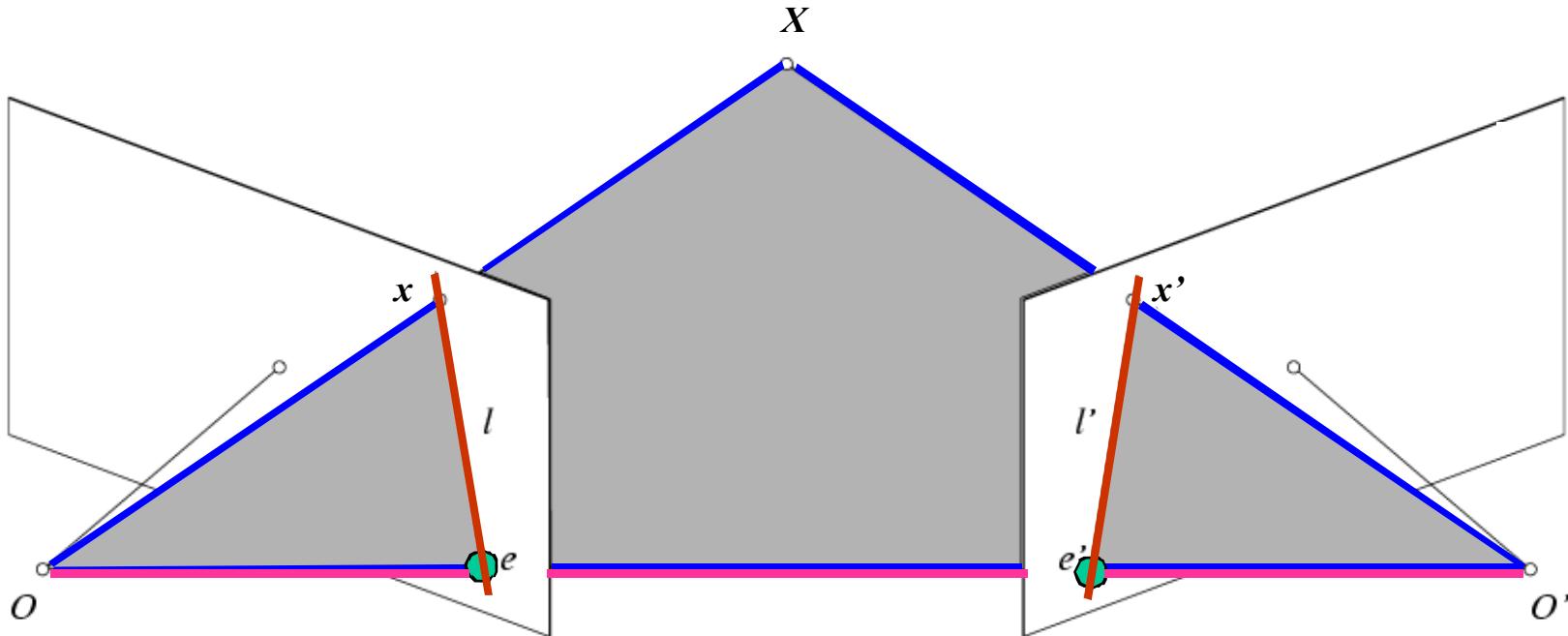
- Перспективная камера
 - Модель перспективной камеры
 - Отличия от реальной камеры
- Многовидовая геометрия
 - Гомография
 - Эпиполярная геометрия
- «Структура из движения» и подзадачи
- Применение – строим Рим за 24 часа

Геометрия 2x камер



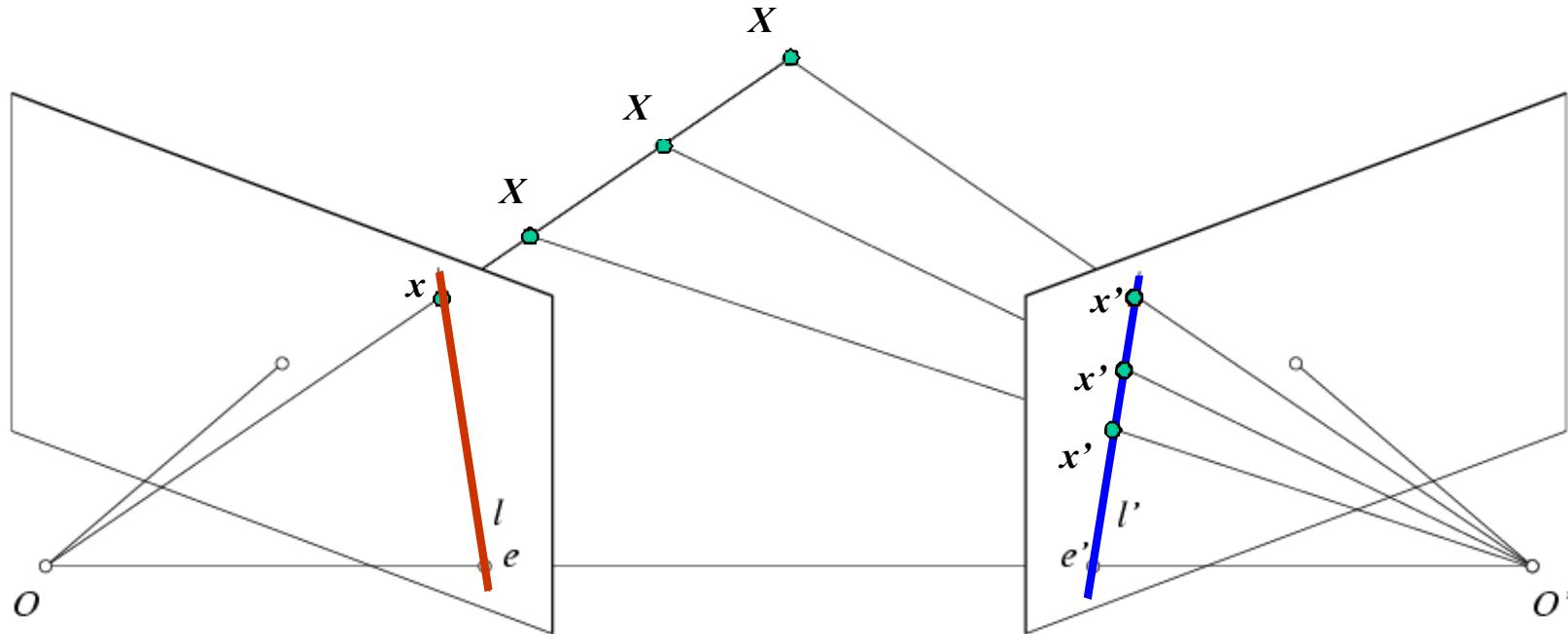
Рассмотрим случай, когда два снимка одной и той же сцены получены с разных ракурсов (разные центры проекций)

Эпиполярная геометрия



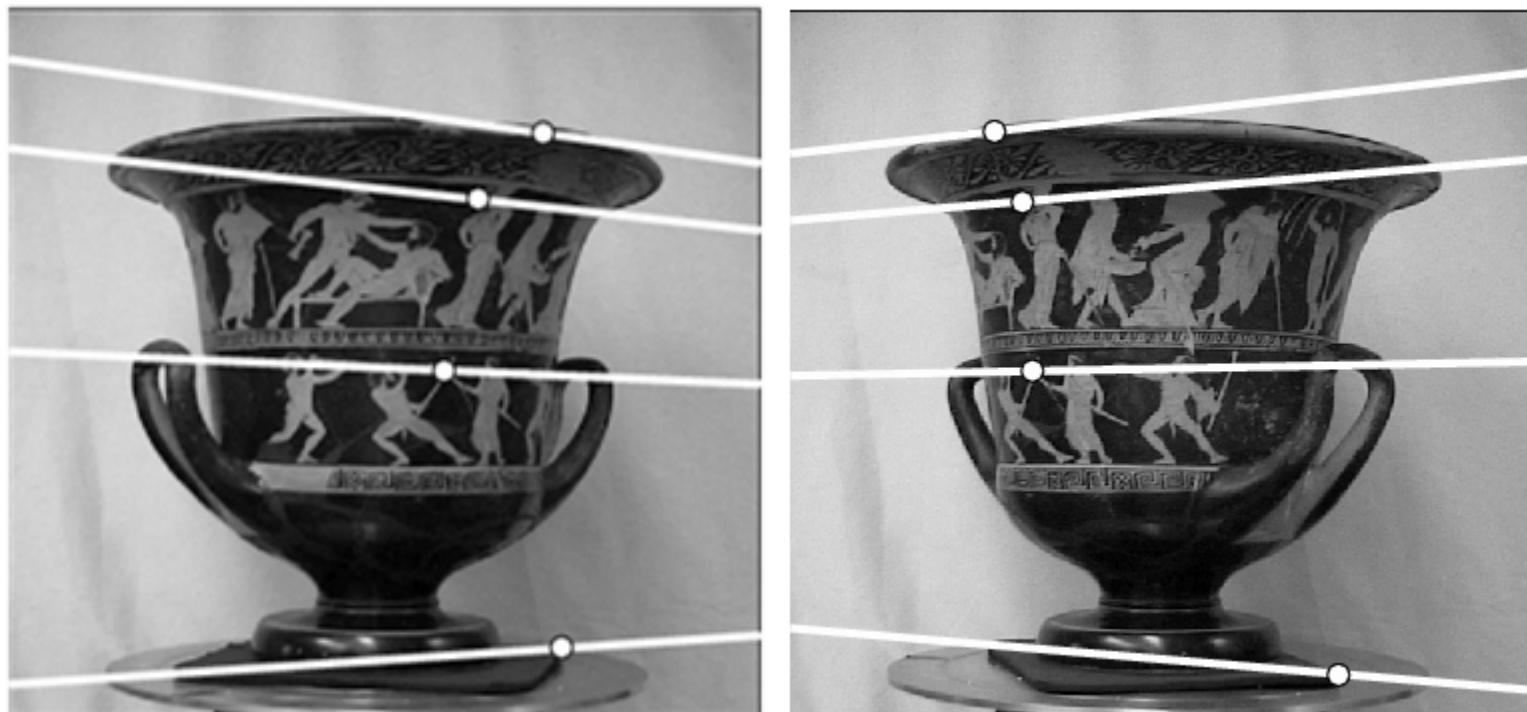
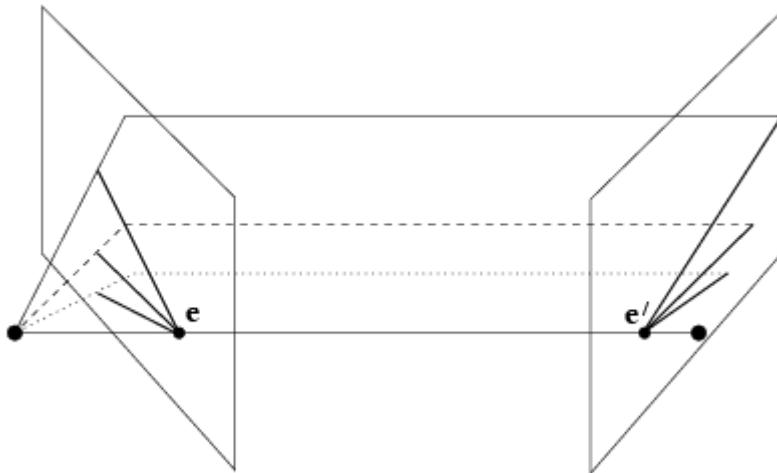
- **Базовая линия (Baseline)** – линия, соединяющая центры камер
- **Эпиполярная плоскость (Epipolar Plane)** – плоскость, проходящая через базовую линию (пучок плоскостей))
- **Эпиполи/Эпиполяры (Epipoles)**
 - = пересечение базовой линии с картинными плоскостями
 - = проекции центра проекций второй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

Эпиполярное ограничение



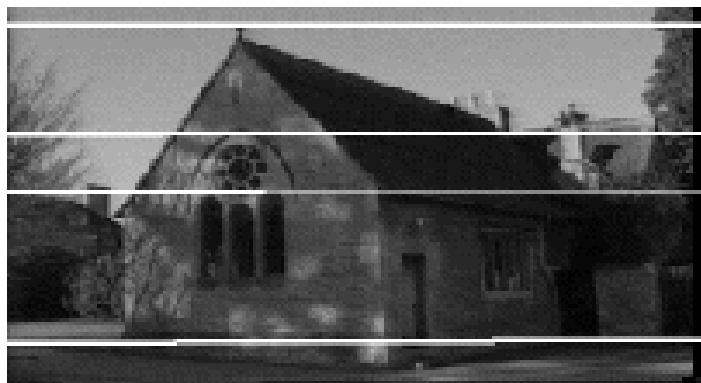
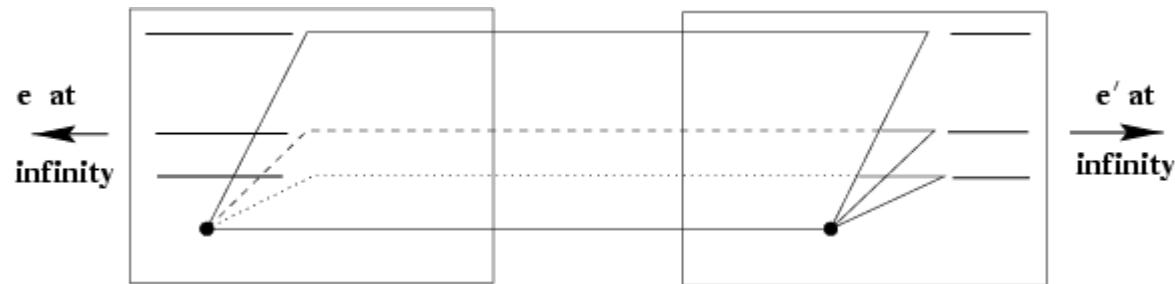
- Пусть даны изображения I и I' с центрами проекции O и O' , и проекция x точки X (неизвестной) на изображение I .
- Что мы можем сказать про точку x' – проекцию X на I' ?
 - x' должна лежать на эпиполярной линии l'
- Эпиполярное ограничение:
 - Проекции x и x' точки сцены X на изображения I и I' должны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

Пример: Сближающиеся камеры



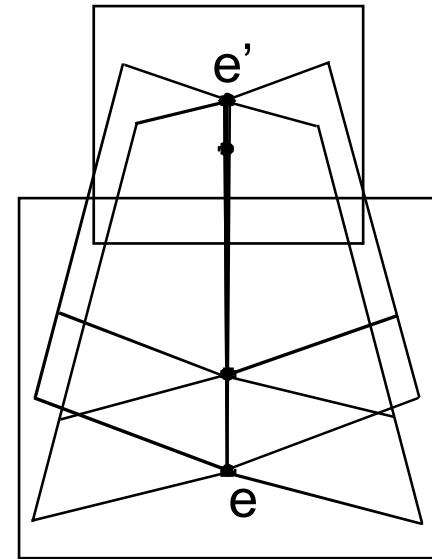
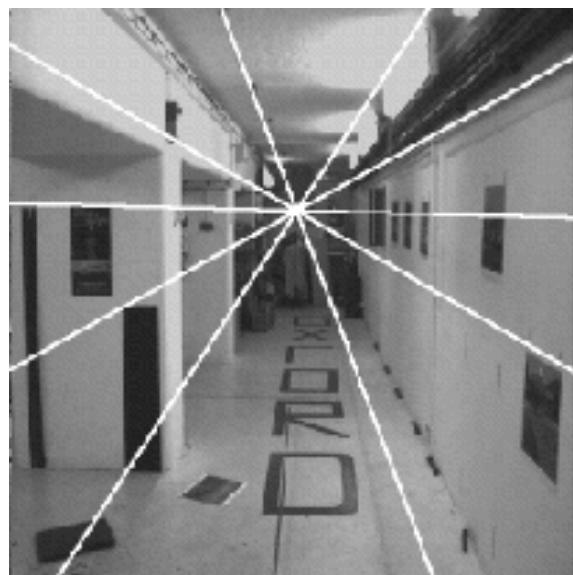
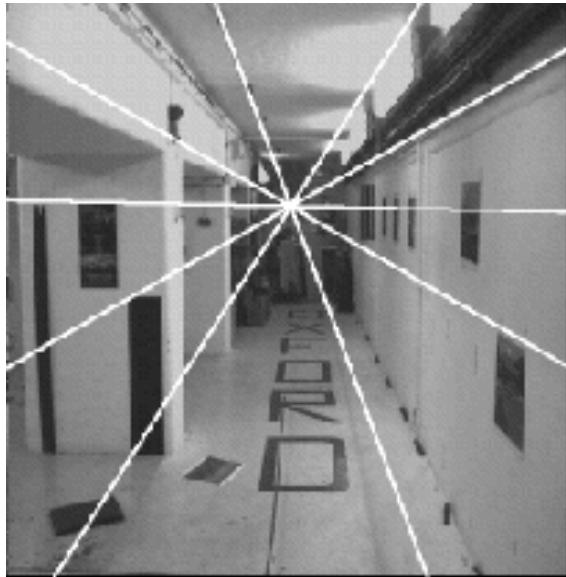


Пример: стереопара



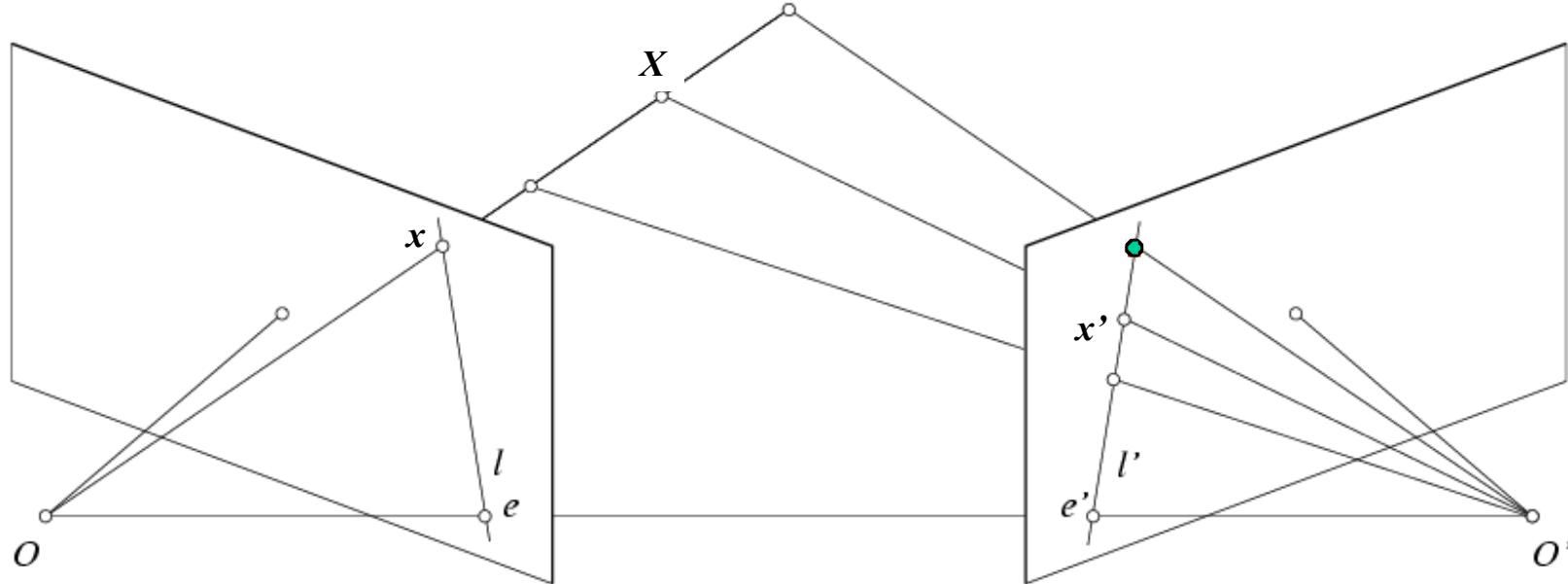
Движение, параллельное картинной плоскости

Пример: движение вперед



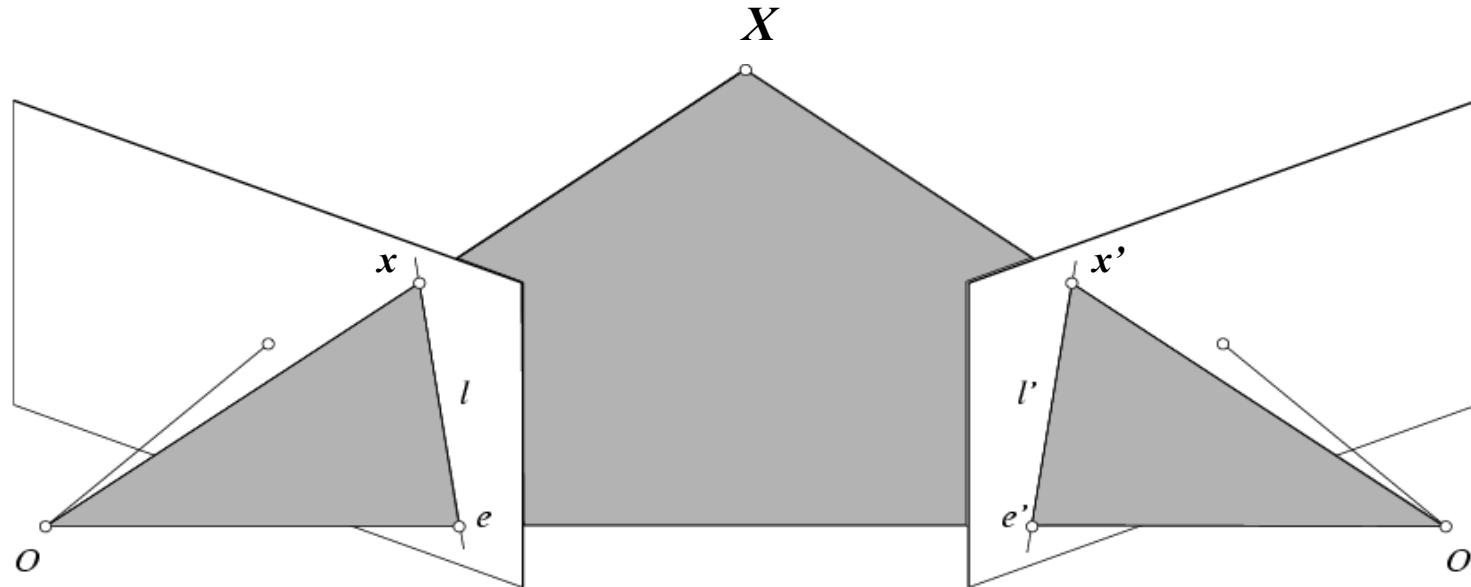
У эпиполей одинаковые координаты в обоих изображениях.
Точки движутся вдоль линий, исходящих из e – «фокуса расширения»

Эпиполярное ограничение



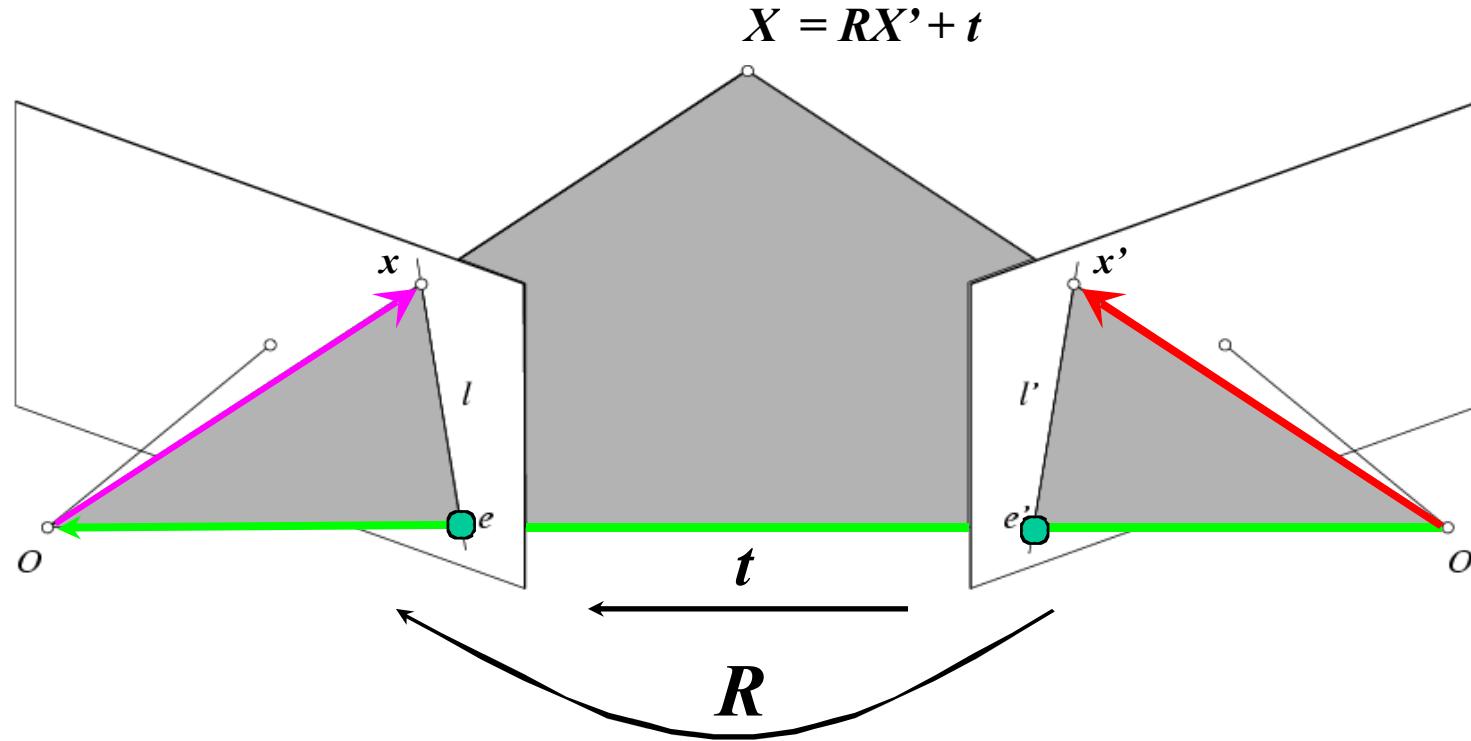
- **Интересный вывод:**
 - Линия l' не зависит от X , если x и калибровка камер известна
 - Линия l не зависит от X , если x' и калибровка камер известна
 - Между x , x' и калибровкой камер существует зависимость
 - Попробуем её вывести

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 $[I|0]$

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

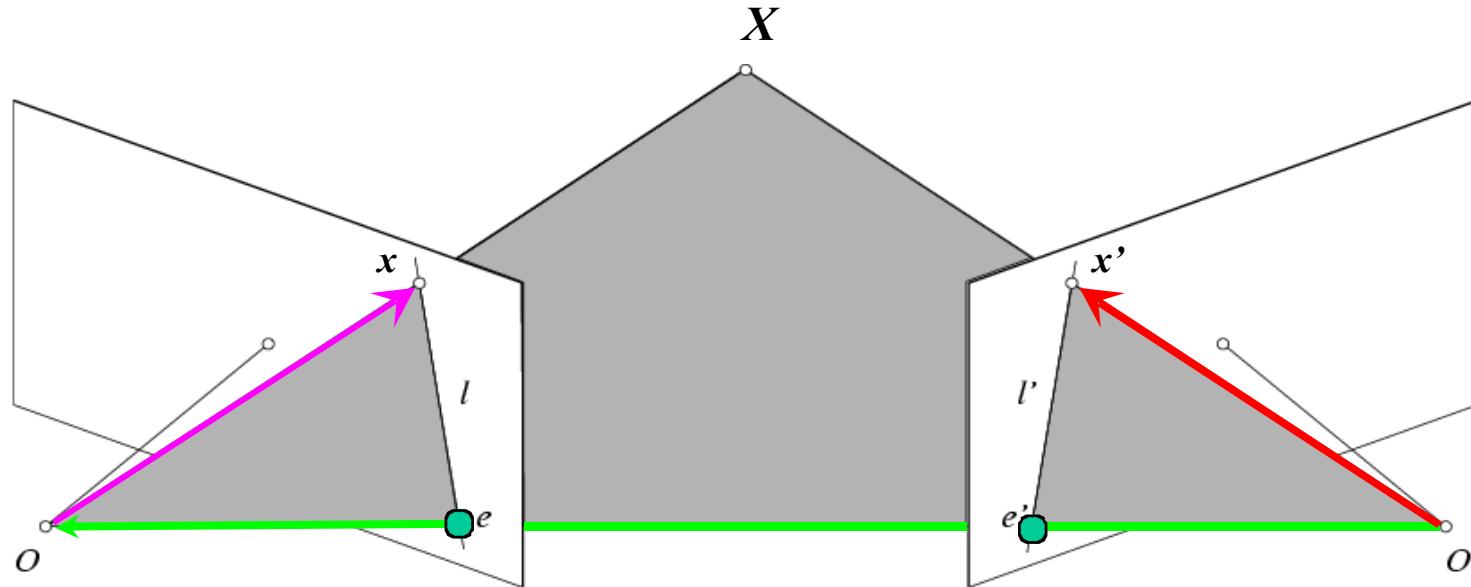
$$x = (u, v, w)^T$$

Матрица камеры: $[R^T | -R^T t]$

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

Вектора x , t , и Rx' копланарны

Калибранный случай



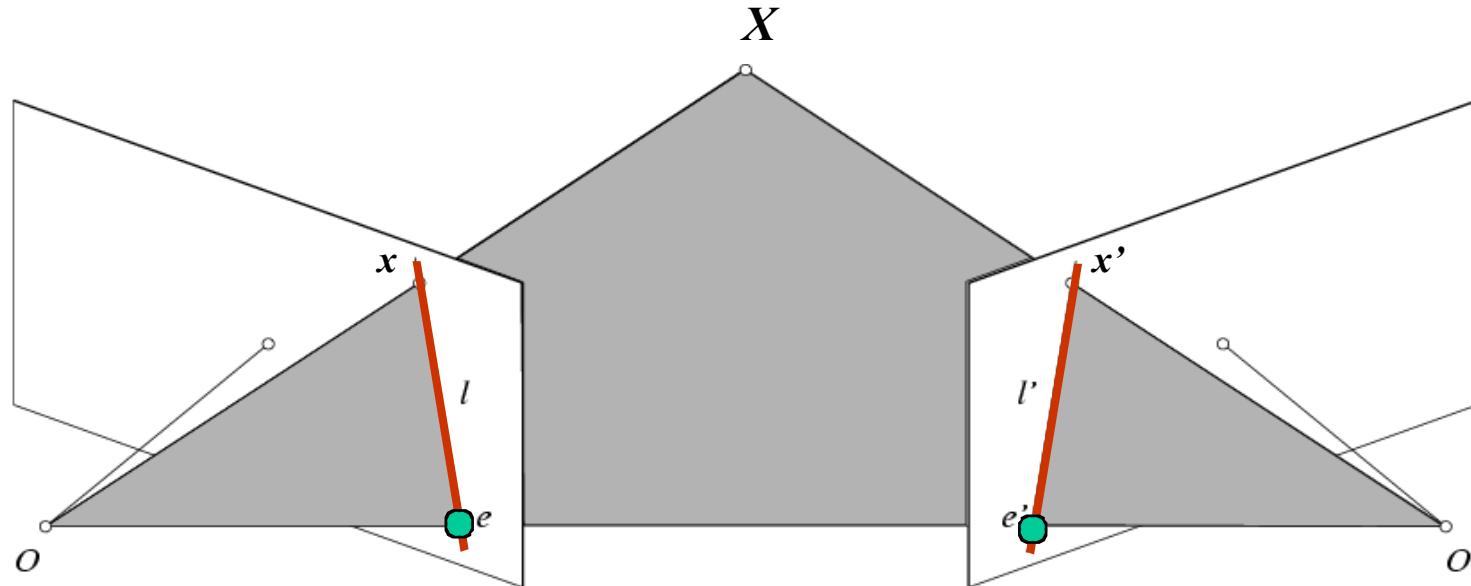
$$x \cdot [t \times (Rx')] = 0 \quad \rightarrow \quad x^T E x' = 0 \quad \text{где} \quad E = [t_x]R$$



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

Вектора x , t , и Rx' копланарны

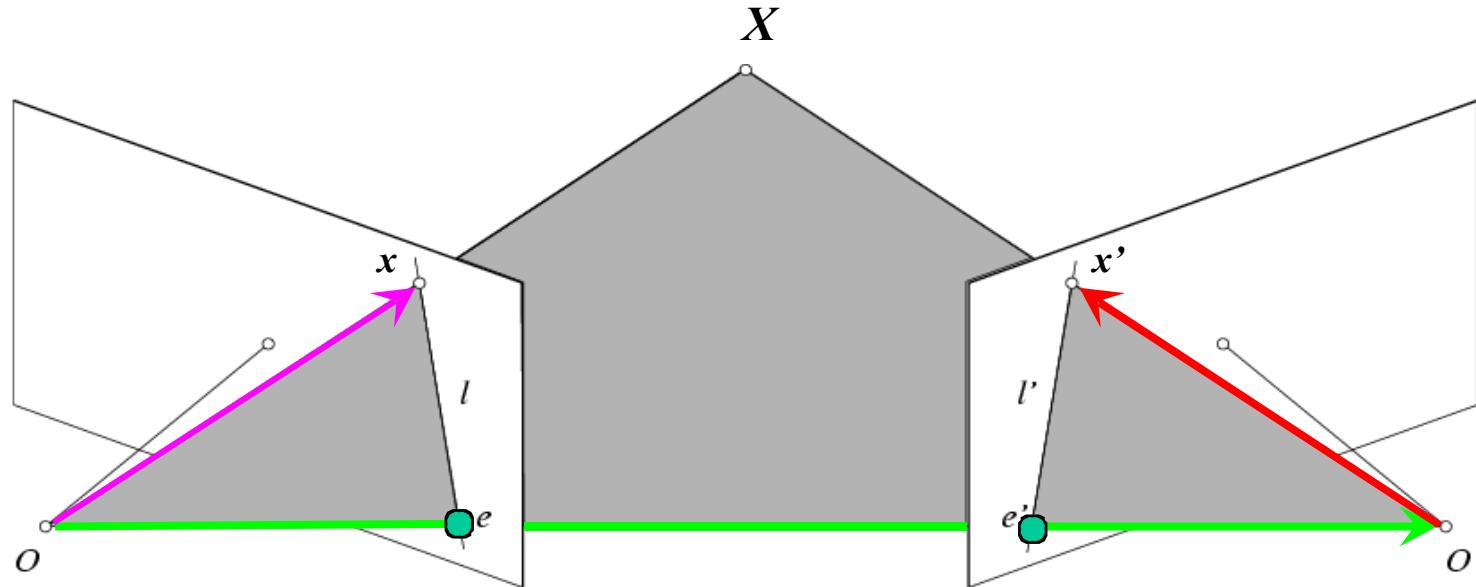
Свойства существенной матрицы



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

- $E x'$ - 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

Некалибранный случай

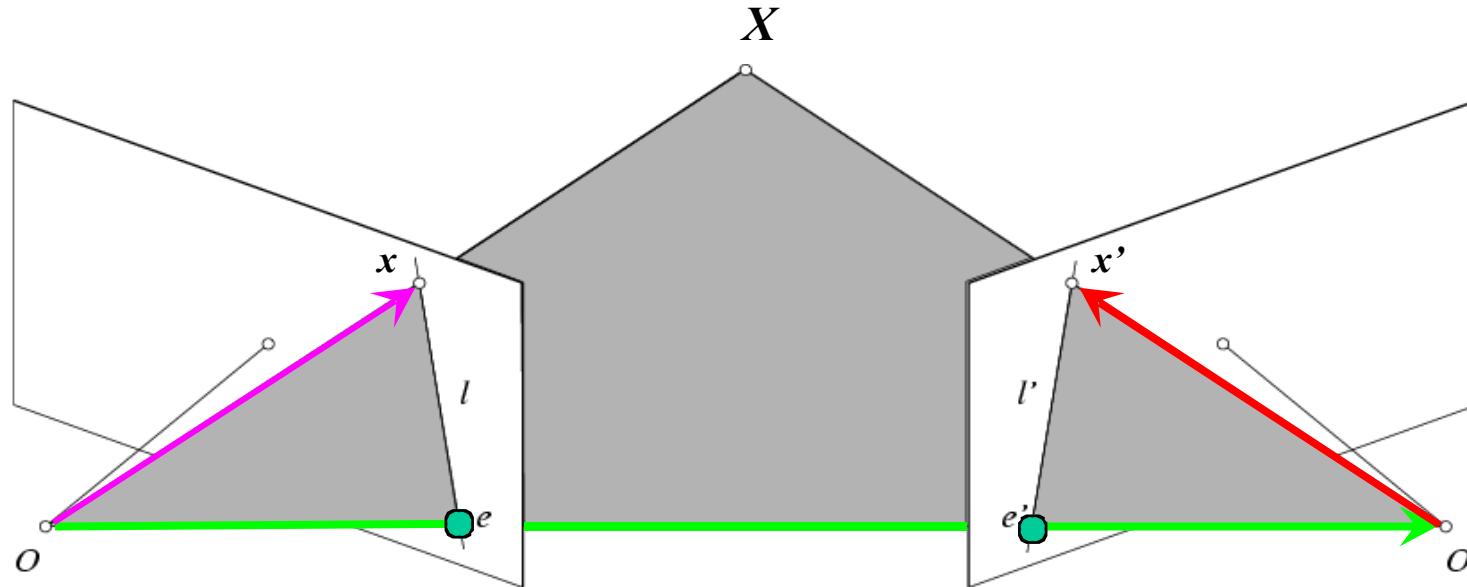


- Пусть матрицы калибровки K и K' для обеих камер неизвестны
- Запишем эпиполярное ограничение через неизвестные нормализованные координаты:

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

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

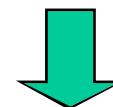
Некалибранный случай



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

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

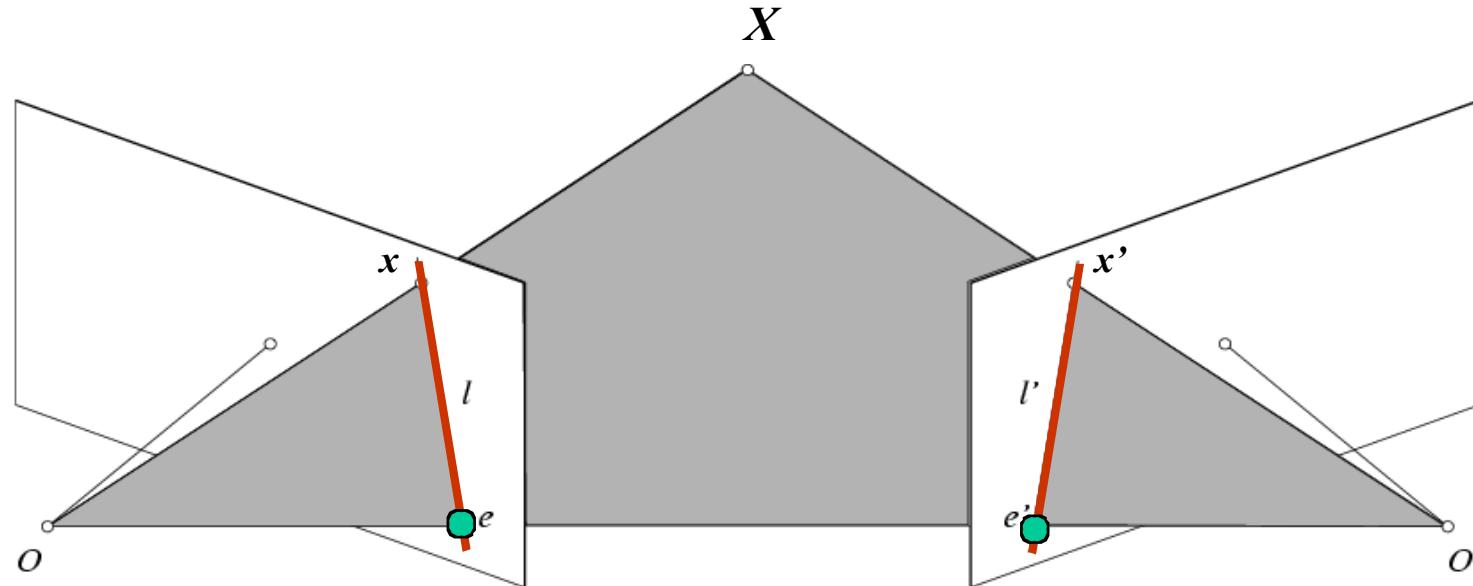
$$x' = K' \hat{x}'$$



**Фундаментальная матрица
(Fundamental Matrix)**

(Faugeras and Luong, 1992)

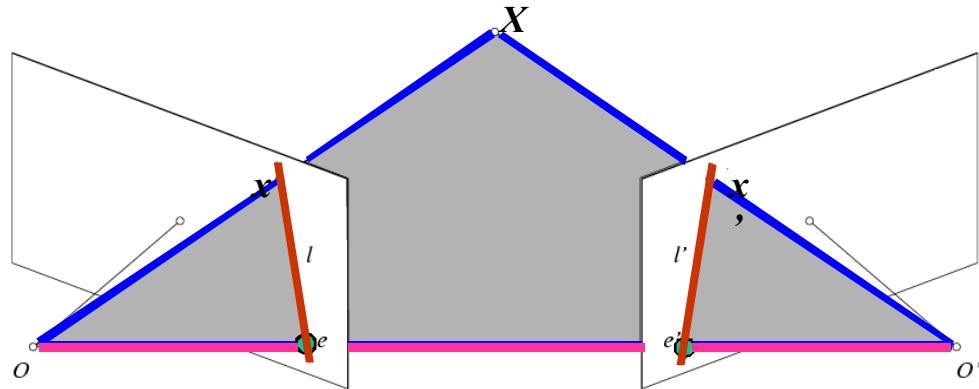
Свойства



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

- 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
(калиброванный случай):

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

Фундаментальная матрица
(некалиброванный случай):

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

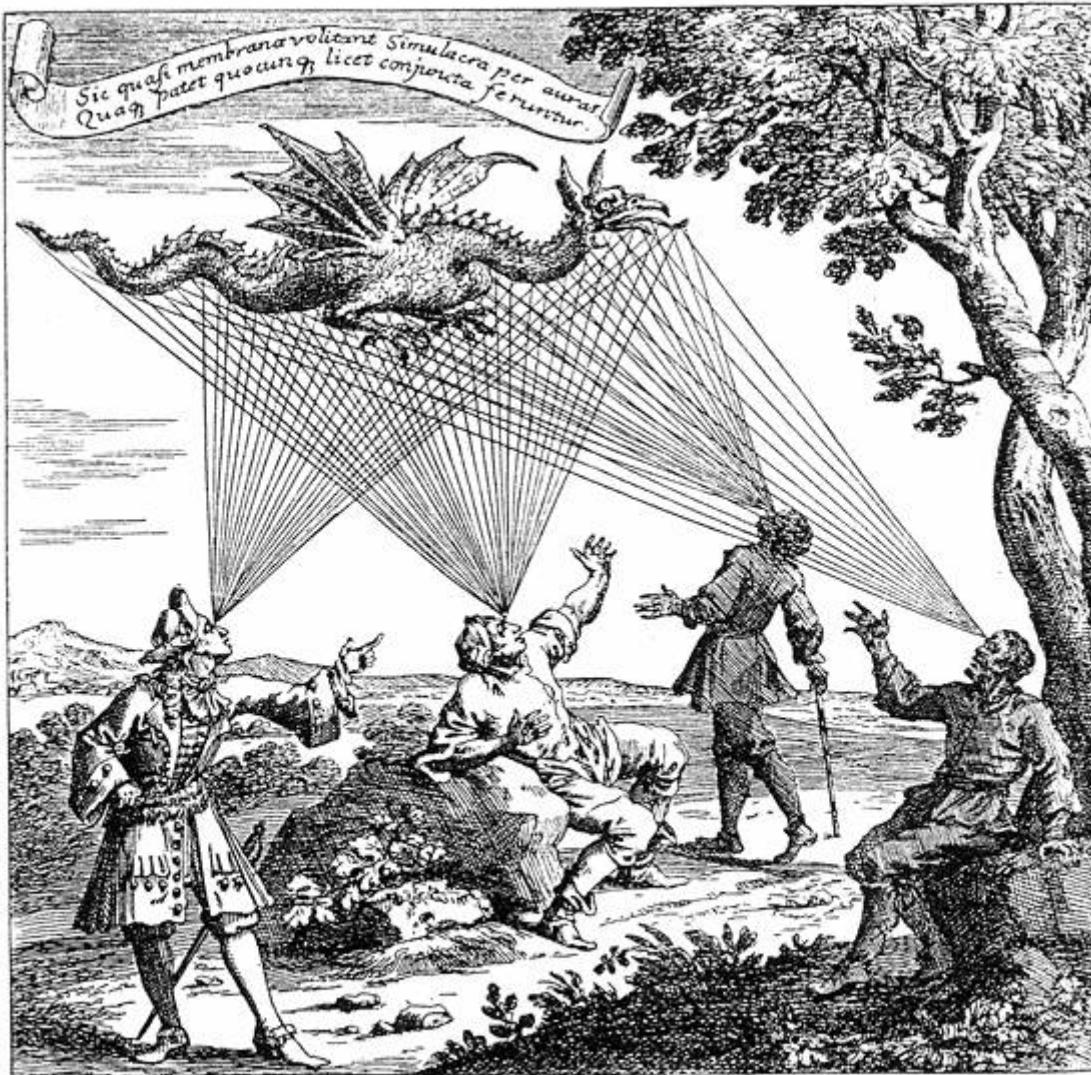
Существенная и фундаментальная матрицы описывают зависимости между калибровкой камер и парам соответствующих точек на изображениях

План лекций



- Перспективная камера
 - Модель перспективной камеры
 - Отличия от реальной камеры
- Многовидовая геометрия
 - Гомография
 - Эпиполярная геометрия
- «Структура из движения» и подзадачи
- Применение – строим Рим за 24 часа

Структура из движения



Драконъ, видимый подъ различными углами зреинъ
По гравюре на изди изъ „Oculus artificialis teledioptricus“ Наталя. 1702 года.

Structure
from
Motion
(SfM)

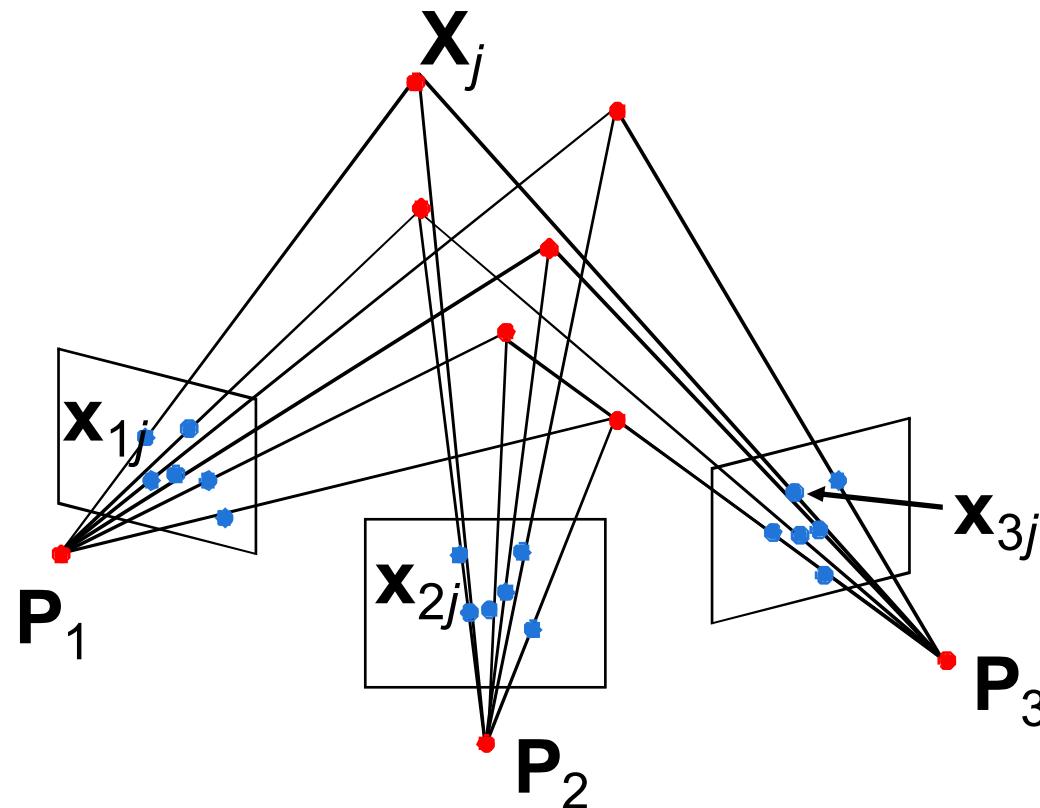
Структура из движения



- Дано: m изображений n фиксированных 3D точек

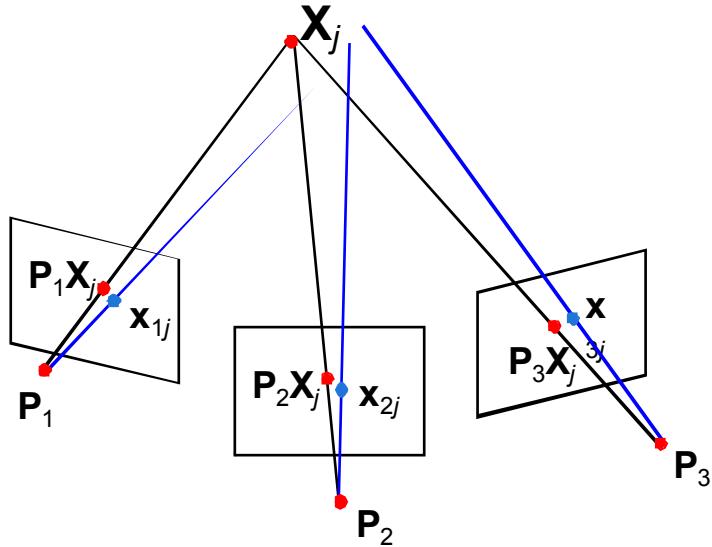
$$\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n$$

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





Целевая функция

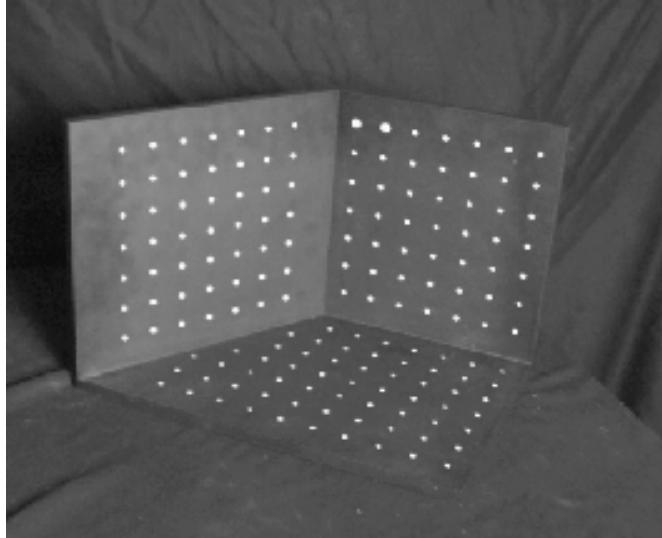


Оптимальная целевая функция –
сумма ошибок репроекций 3D точек
X на все изображения:

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

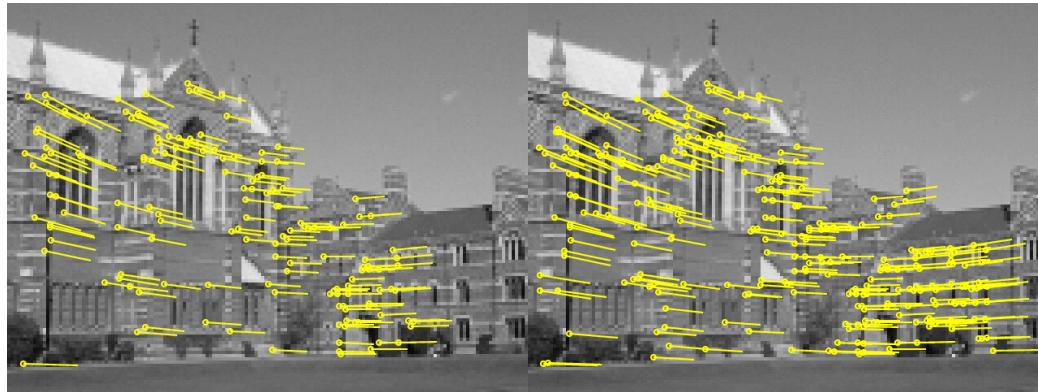
- Нелинейная целевая функция с большим количеством параметров
 - 3n точек
 - 6m параметров для калиброванного случая, 11m для некалиброванного
- Решается градиентным спуском, но плохо – масса локальных минимумов
- Поэтому нужно получить хорошее начальное приближение, в этом основная проблема

Что мы можем знать об изображениях?



Геометрию объекта и
характерные точки

2D – 3D соответствия



Набор
предположительных
соответствий особых
точек

2D – 2D соответствия

Подзадачи



- **Калибровка камеры (PnP-проблема):** Дан набор 2D-3D соответствий. Определить параметры камеры.
- **Оценка движения камеры:** Дан набор соответствующих точек в 2x и более изображениях. Определить матрицы калибровки камер для этих изображений (видов)
- **Соответствие (стерео-сопоставление) :** Данна точка на одном изображении. Как мы можем ограничить положение соответствующей точки на втором изображении?
- **Геометрия сцены (структура) :** Пусть даны соответствующие точки (проекции) в 2D на 2x и более изображениях. Определить 3D координаты точки.



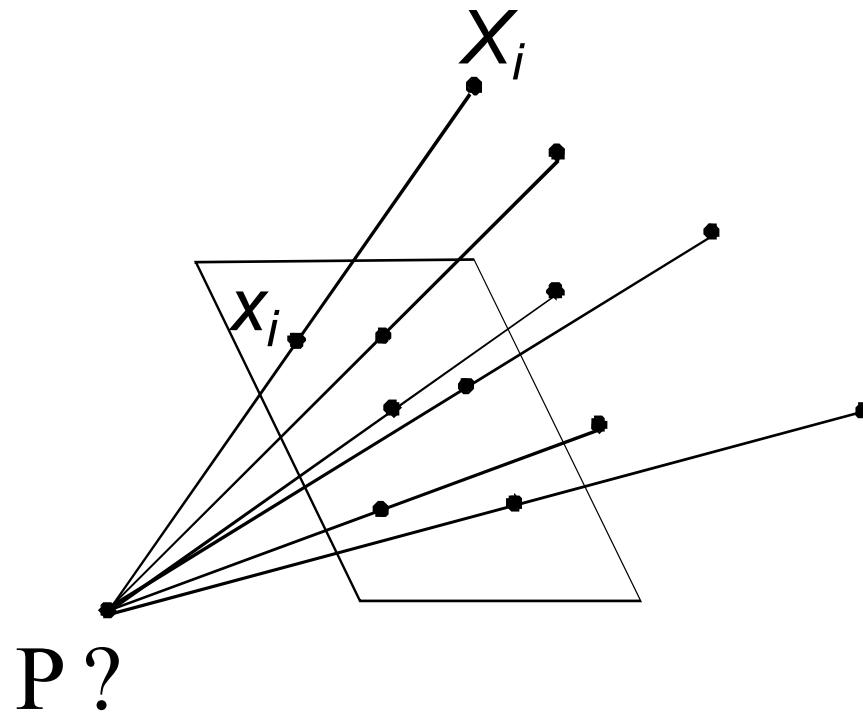
Подзадачи

- **Калибровка камеры (PnP-проблема):** Дан набор 2D-3D соответствий. Определить параметры камеры.
- **Оценка движения камеры:** Дан набор соответствующих точек в 2x и более изображениях. Определить матрицы калибровки камер для этих изображений (видов)
- **Соответствие (стерео-сопоставление) :** Данна точка на одном изображении. Как мы можем ограничить положение соответствующей точки на втором изображении?
- **Геометрия сцены (структура) :** Пусть даны соответствующие точки (проекции) в 2D на 2x и более изображениях. Определить 3D координаты точки.

Калибровка камеры (PnP-проблема)



Perspective- n -point problem (PnP) - даны n точек с известными 3D координатами X_i и известными проекциями x_i , оценить параметры камеры





DLT-метод

Direct Linear Transformation (прямое линейное преобразование)

$$\lambda \mathbf{x}_i = \mathbf{P}\mathbf{X}_i \quad \lambda \begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1^T \\ \mathbf{P}_2^T \\ \mathbf{P}_3^T \end{bmatrix} \mathbf{X}_i \quad \mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = 0$$

$$\begin{bmatrix} 0 & -w_i \mathbf{X}_i^T & y_i \mathbf{X}_i^T \\ w_i \mathbf{X}_i^T & 0 & -x_i \mathbf{X}_i^T \\ -y_i \mathbf{X}_i^T & x_i \mathbf{X}_i^T & 0 \end{bmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = 0$$

- Два линейно независимых уравнения (берут обычно первые два)

Калибровка камеры



$$\begin{bmatrix} 0^T & X_1^T & -y_1 X_1^T \\ X_1^T & 0^T & -x_1 X_1^T \\ \dots & \dots & \dots \\ 0^T & X_n^T & -y_n X_n^T \\ X_n^T & 0^T & -x_n X_n^T \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0 \quad Ap = 0$$
$$\|p\| = 1$$

- У Р 11 степеней свободы (12 параметров, но масштаб произвольный)
- Каждая пара (x, X) даёт нам уравнения
- 6 соответствий необходимо для решения (рбр)
- Однородные наименьшие квадраты



Калибровка камеры

$$\begin{bmatrix} 0^T & \mathbf{X}_1^T & -y_1 \mathbf{X}_1^T \\ \mathbf{X}_1^T & 0^T & -x_1 \mathbf{X}_1^T \\ \dots & \dots & \dots \\ 0^T & \mathbf{X}_n^T & -y_n \mathbf{X}_n^T \\ \mathbf{X}_n^T & 0^T & -x_n \mathbf{X}_n^T \end{bmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = 0 \quad \begin{array}{l} \mathbf{A}\mathbf{p} = 0 \\ \|\mathbf{p}\| = 1 \end{array}$$

Для копланарных точек, удовлетворяющих
уравнению $\Pi^T \mathbf{X} = 0$, получим вырожденные
решения $(\Pi, 0, 0)$, $(0, \Pi, 0)$, или $(0, 0, \Pi)$

Калибровка камеры



- Когда мы получили матрицу проекции, нам необходимо извлечь из неё внутренние и внешние параметры калибровки
- Представим матрицу проецирования в следующем виде:

$$P = KR^T [I \mid -T] = [M \mid -MT]$$

- Для разложения M в KR^T воспользуемся RQ-факторизацией
 - Q – ортогональная матрица
 - K – верхнетреугольная
- Потом найдем T

“Gold Standard”



- Метод «Золотого стандарта» – метод, оптимизирующий параметры по наиболее корректной, оптимальной метрике
- В случае калибровки камеры, это расстояния от проекций 3D точек до их измеренных значений (ошибка репроекции):

$$\min_P \sum_i d^2(x_i, P X_i)$$

- Требует использования нелинейных методов оптимизации (градиентный спуск и т.д.)

Схема калибровки камеры



- DLT-метод для получения матрицы проецирования
- RQ-факторизация для извлечения матрицы внутренней калибровки K и внешней калибровки R и T
- Уточнение параметров калибровки с помощью нелинейной оптимизации ошибки репроекции:

$$\min_P \sum_i d^2(x_i, P X_i)$$

Варианты задачи



3х точек должно быть достаточно для внешней калибровки (6 неизвестных)

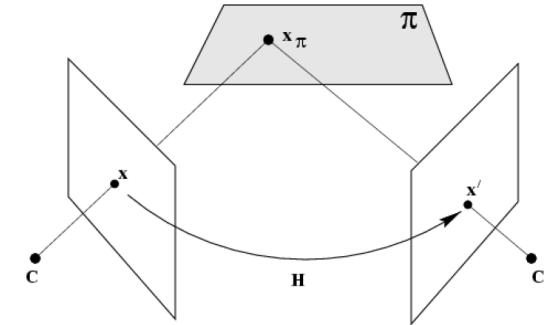
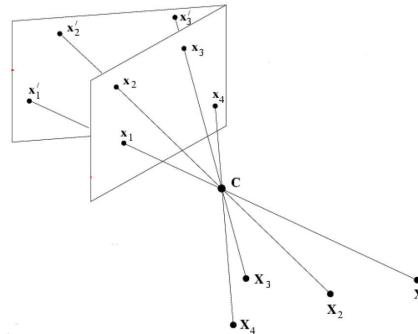
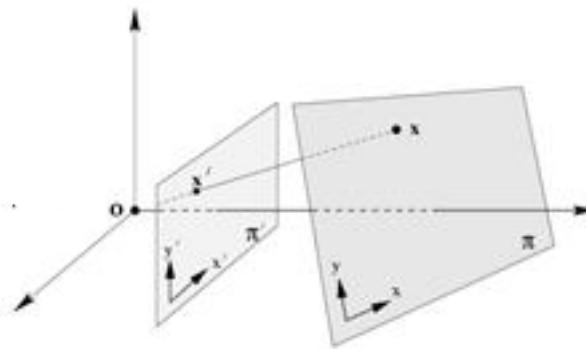
- P3P – задача имеет от 1 до 4x решений
- P4P, P5P – предложены линейные методы, которые находят единственное решение, кроме вырожденных случаев
- P4P – есть решения для поиска одновременно внешней калибровки и фокусного расстояния
 - M. Bujnak, Z. Kukelova, T. Pajdla. A general solution to the P4P problem for camera with unknown focal length. CVPR 2008
 - Через разложение по базисам Грёбнера



Подзадачи

- **Калибровка камеры (PnP-проблема):** Дан набор 2D-3D соответствий. Определить параметры камеры.
- **Оценка движения камеры:** Дан набор соответствующих точек в 2x и более изображениях. Определить матрицы калибровки камер для этих изображений (видов)
- **Соответствие (стерео-сопоставление) :** Данна точка на одном изображении. Как мы можем ограничить положение соответствующей точки на втором изображении?
- **Геометрия сцены (структура) :** Пусть даны соответствующие точки (проекции) в 2D на 2x и более изображениях. Определить 3D координаты точки.

Оценка гомографии



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

Гомография хорошо описывает преобразование между 2мя изображениями в следующих случаях:

- Плоская сцена
- Вращающаяся камера

Дан набор соответствующих точек в 2х изображениях, определить гомографию между ними

Вычисление гомографии (DLT)



- (x, x') – известные соответствия на 2x изображениях
- Линейный (DLT) метод:
- Составим систему линейных уравнений $\lambda x' = Hx$
 - Каждое соответствие даст 2 уравнения на параметры H
 - 4 соответствия дадут 8 уравнений на 8 неизвестных
 - Никакие 3 не должны лежать на одной прямой!
 - Можем взять и больше соответствий...
- Сведем к однородной задаче наименьших квадратов
- Из 4x соответствий (x, x') никакие 3 не должны лежать на одной прямой

Вычисление гомографии (Gold)



- Критерий получше (transfer error)

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

- Вычислить достаточно просто

- Оптимальный критерий (distance error)

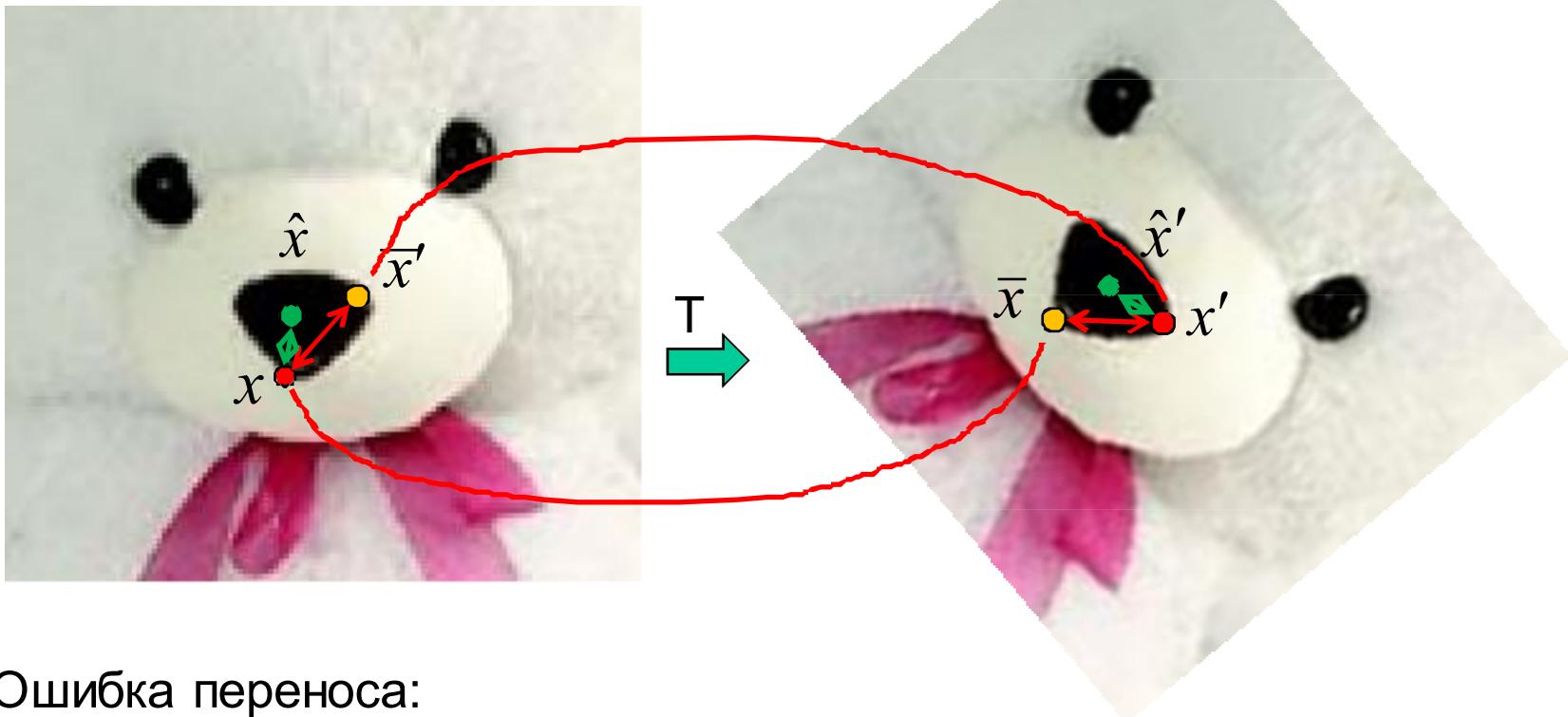
$$\sum_i d^2(x'_i, \hat{x}'_i) + d^2(x_i, \hat{x}_i)$$

где $\hat{x}'_i = H\hat{x}_i$ - идеальное соответствие

- Нужно вычислять «идеальные соответствия», это сложно

- Для робастности применяем RANSAC

Геометрические преобразования



Ошибка переноса:

$$\sum_i d^2(x'_i, \bar{x}_i) + d^2(x_i, \bar{x}'_i) \quad \bar{x} = Tx \quad \bar{x}' = T^{-1}x'$$

Геометрическая ошибка:

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

Калибровка камер (движение)



Дан набор соответствующих точек в 2x и более изображениях, определить матрицы калибровки камер для этих изображений (видов)

Существенная матрица:

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

Фундаментальная матрица:

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

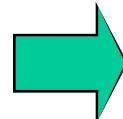
8-и точечный алгоритм



- Фундаментальная матрица накладывает ограничения на пары соответствующих точек
- Попробуем её вычислить напрямую из соответствий, без калибровки камеры

- Даны соответствия (x, x') : $x = (u, v, 1)^T$, $x' = (u', v', 1)^T$

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



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

- 1 соответствие 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

$$|F|^2 = 1$$

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 $r \geq s \geq t$
- $F' = U\text{diag}(r,s,0)V^T$

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

Нормализованный 8-точечный



(Hartley, 1995)

- Переносим центр данных с изображения в начало координат, и масштабируем так, чтобы среднее квадратичное расстояние от точек до начала координат было 2 пикселя
- Применяем 8и точечный метод для вычисления F из нормализованных точек
- Приводим ранг матрицы F к 2
- Преобразуем матрицу к нормальным единицам: если T и T' нормализующие преобразования, тогда фундаментальная матрица в исходных координатах равна $T^T F T'$

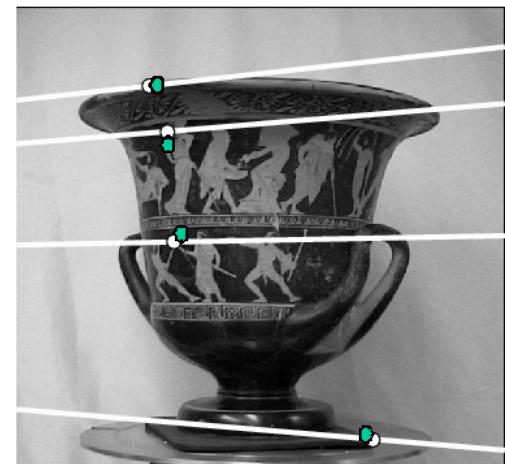
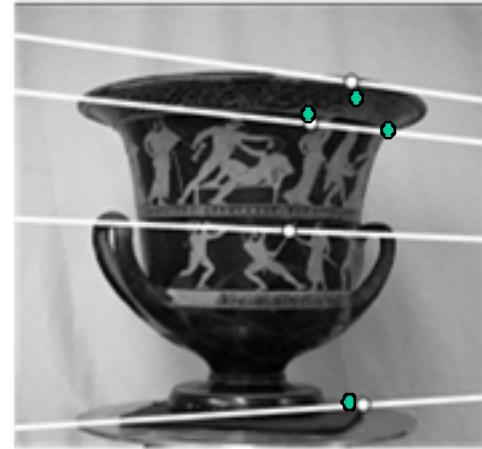


“Gold standard”

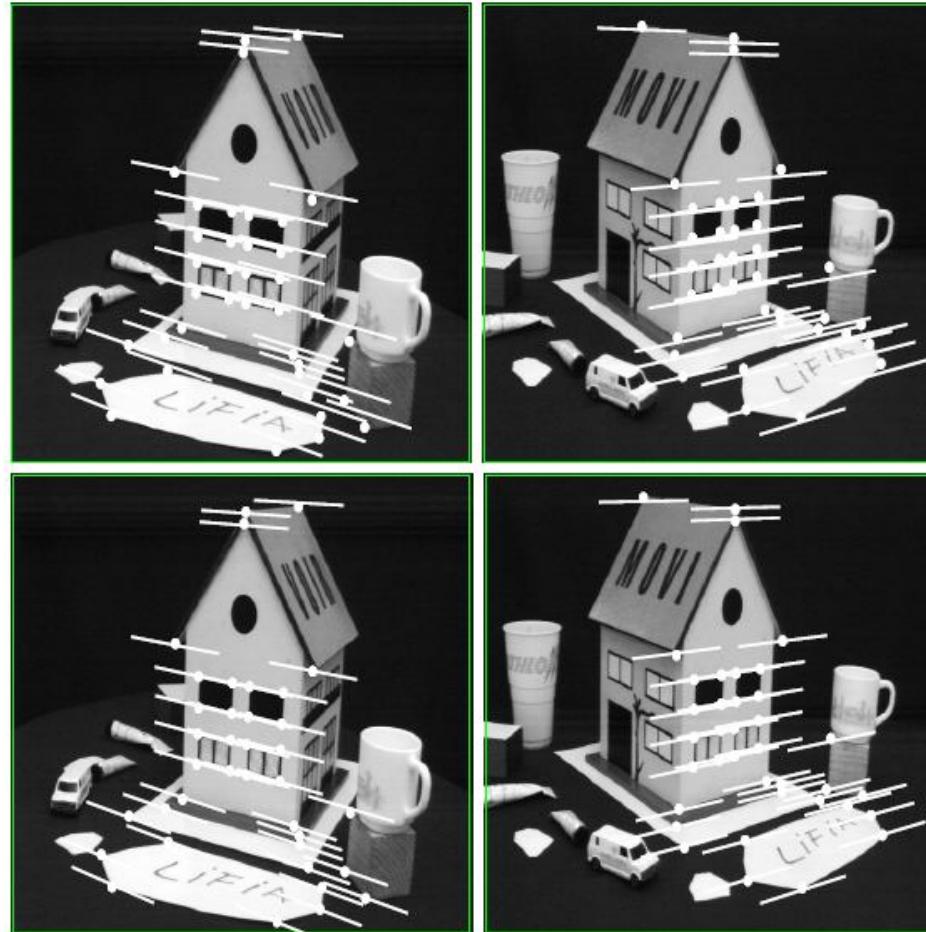
- Ошибка $\sum_{i=1}^N (x_i^T F x_i')^2$ - алгебраическая
 - Не имеет физического смысла
- «Оптимальная» ошибка:

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

- Т.е. сумма квадратов расстояний от измеренных точек, до «идеальных»
- Требует нелинейной оптимизации



Сравнение алгоритмов



	8-point	Normalized 8-point	Nonlinear least squares
Av. Dist. 1	2.33 pixels	0.92 pixel	0.86 pixel
Av. Dist. 2	2.18 pixels	0.85 pixel	0.80 pixel

Если данные зашумлены?



- В соответствиях могут быть ошибки!
 - Ложные соответствия
 - Нужно использовать робастные методы
 - Обычно используют метод RANSAC с ошибкой:

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

- Ошибка переноса (transfer error)
- Затем уточняют по всем найденным корректным соответствиям



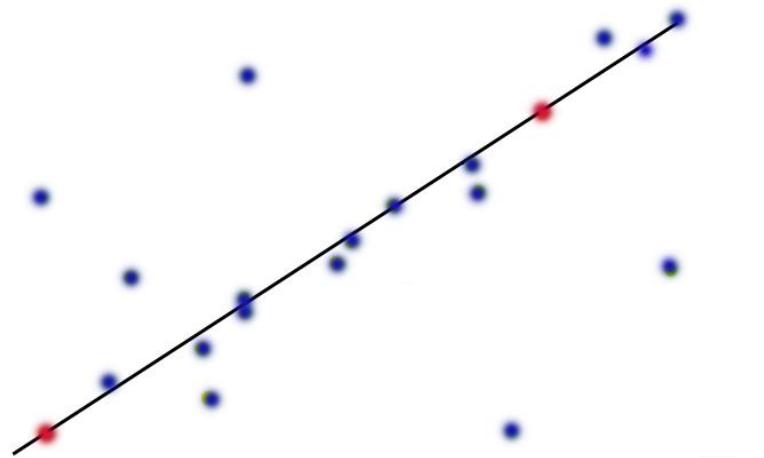
- *Идея* – проведение оценки не по всем данным, а выборке, не содержащей выбросов
- Поскольку какие элементы данных – выбросы – заранее неизвестно, то...
 - ...строим много выборок случайным образом!
- По каждой выборке строим гипотезу
- Среди всех гипотез выберем ту, которая лучше всего согласуется со всеми данными
- Random sample consensus (RANSAC)

M. A. Fischler, R. C. Bolles. [Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography](#). Comm. of the ACM, Vol 24, pp 381-395, 1981.

Выборка как основа схемы

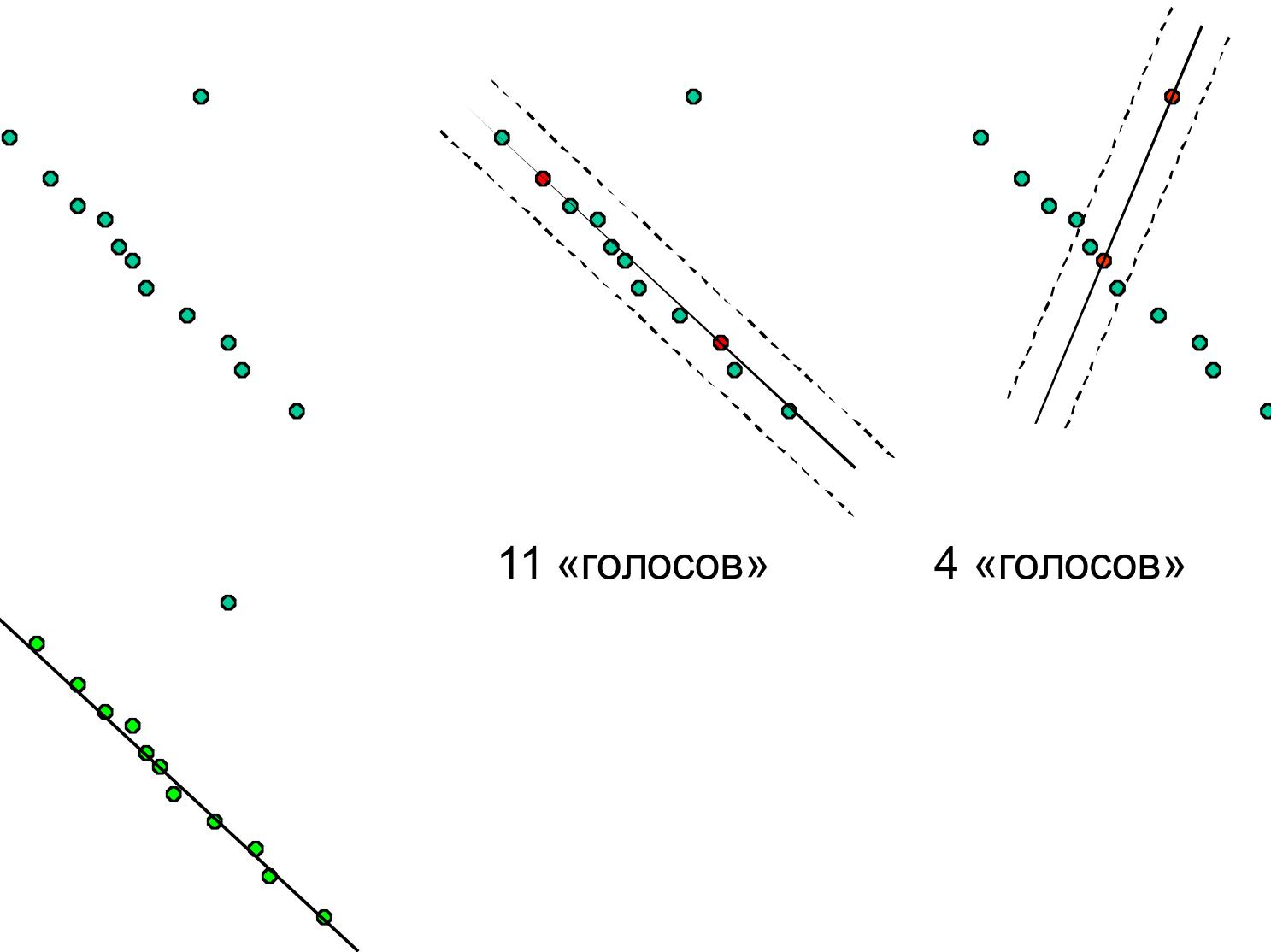


- Основой RANSAC является оценка модели по небольшой выборке $S \subset x$
- Проблема - количество таких выборок огромно
- Поэтому будем строить гипотезы по выборке минимального размера
- Для прямой:



А для геометрических преобразований сколько?

Пример RANSAC



Базовая схема RANSAC



Повторяем N раз:

- (i) Построение случайной выборки $S \subset X$
- (ii) Построение гипотезы Θ по выборке S
- (iii) Оценка качества гипотезы Θ по набору исходных данных X
- (iv) Если гипотеза Θ лучше предыдущих, запоминаем её

После завершения итераций:

- (i) Построение чистой выборки X' путём фильтрации выбросов, не удовлетворяющих Θ_{best}
- (ii) Уточнение гипотезы Θ_{best} по X'

Сколько гипотез нужно проверить?



$$\left(1 - (1 - e)^s\right)^N = 1 - p$$

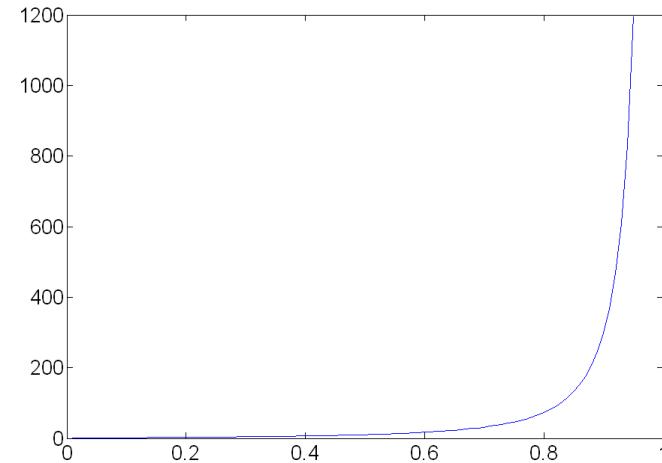
$$N = \log(1 - p) / \log\left(1 - (1 - e)^s\right)$$

- N – количество выборок
- p – вероятность получить хорошую выборку за N итераций;
- s – количество элементов в выборке
- ϵ – процент хороших точек в наборе

Количество итераций



s	proportion of outliers e							
	5%	10%	20%	25%	30%	40%	50%	
2	2	3	5	6	7	11	17	
3	3	4	7	9	11	19	35	
4	3	5	9	13	17	34	72	
5	4	6	12	17	26	57	146	
6	4	7	16	24	37	97	293	
7	4	8	20	33	54	163	588	
8	5	9	26	44	78	272	1177	



- Количество выборок быстро растет с ростом размера выборки и доли выбросов
- Как быть, если мы не знаем долю выбросов в данных?

Адаптивное завершение алгоритма



- Доля e обычно заранее неизвестна, поэтому начинаем с грубой оценки (пр.: 50%), затем вычисляем оценку хороших точек для каждой гипотезы
- Процедура:
 - $N = \infty$, $sample_count = 0$
 - While $N > sample_count$
 - Строим выборку, гипотезу, оцениваем кол-во inliers
 - Установим $e = 1 - (\text{number of inliers})/(\text{total number of points})$
 - Перевычислим N по e :
$$N = \log(1 - p) / \log(1 - (1 - e)^s)$$
 - Увеличим $sample_count$ на 1

Функции качества



Первоначально было предложено 2 способа оценки качества гипотез:

RANSAC

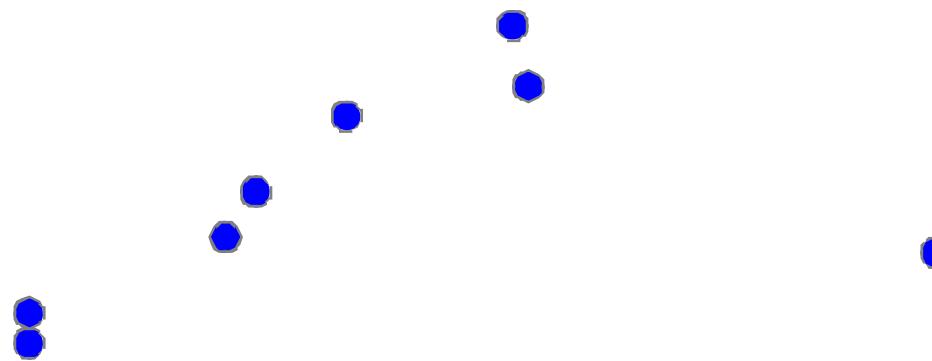
$$R(\theta) = \sum_i p(\varepsilon_i(\theta)^2), \quad p(\varepsilon_i^2) = \begin{cases} 1 & \varepsilon_i^2 \leq T^2 \\ 0 & \varepsilon_i^2 > T^2 \end{cases}, i = \overline{1, n}$$

$\varepsilon_i(\theta)$ - невязка i -ой точки и оцениваемой гипотезы

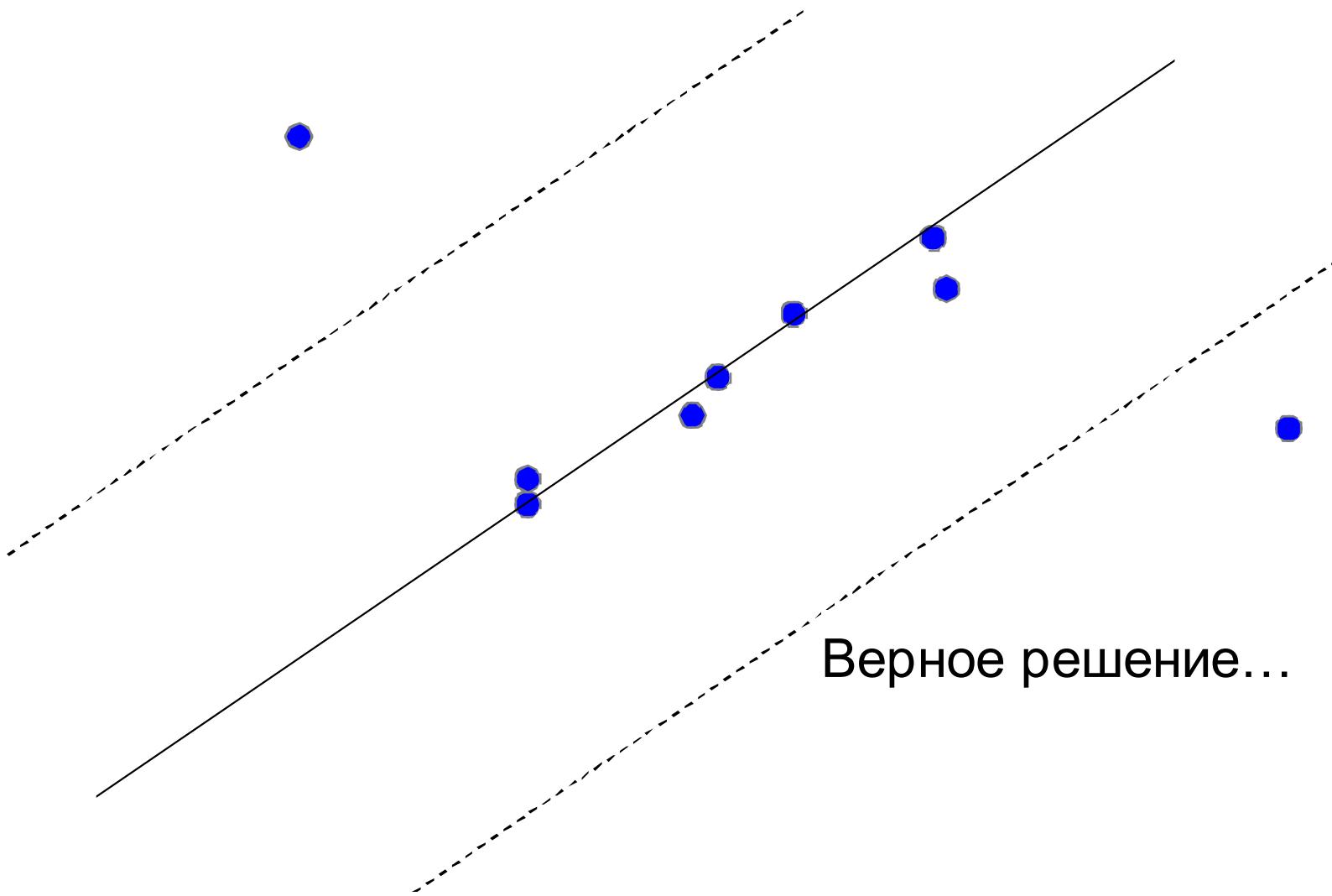
LMS (Least median squares)

$$R(\theta) = median(\varepsilon_i(\theta)^2), i = \overline{1, n}$$

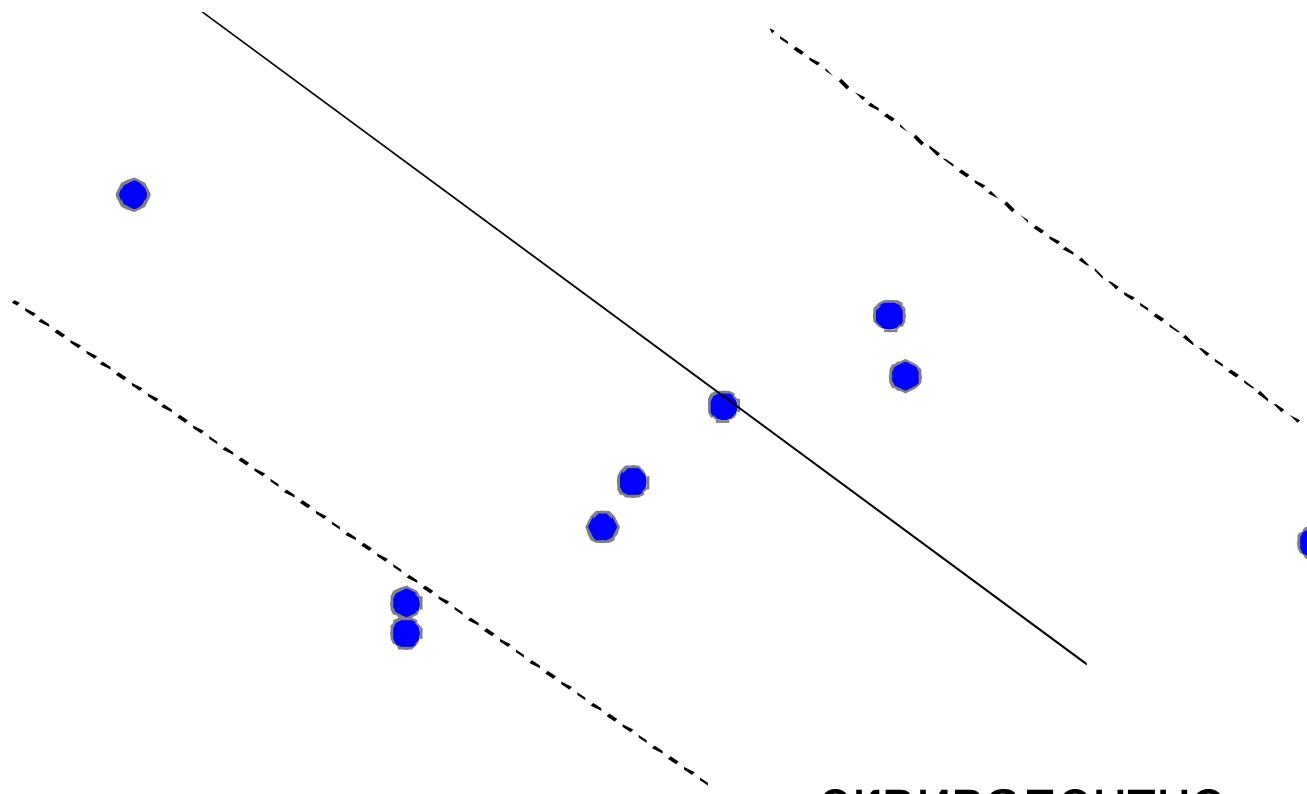
Большой порог – проблема RANSAC!



Большой порог – проблема!

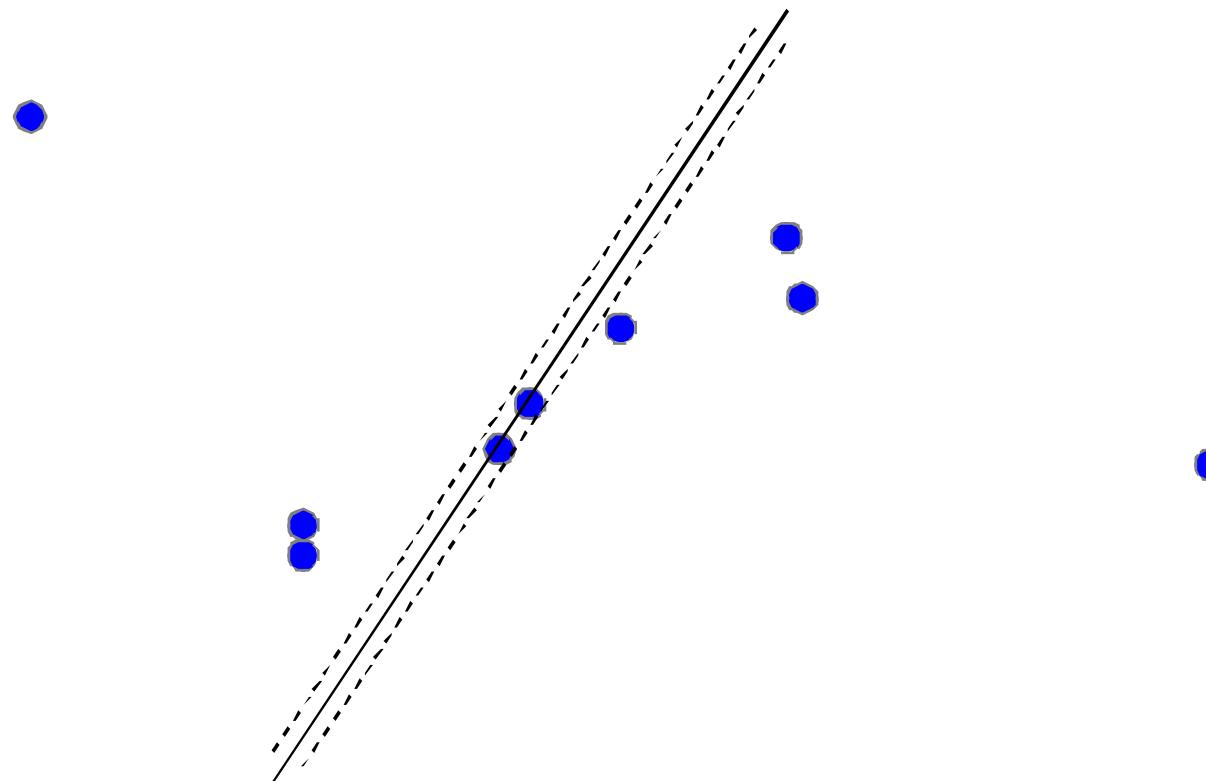


Большой порог – проблема!

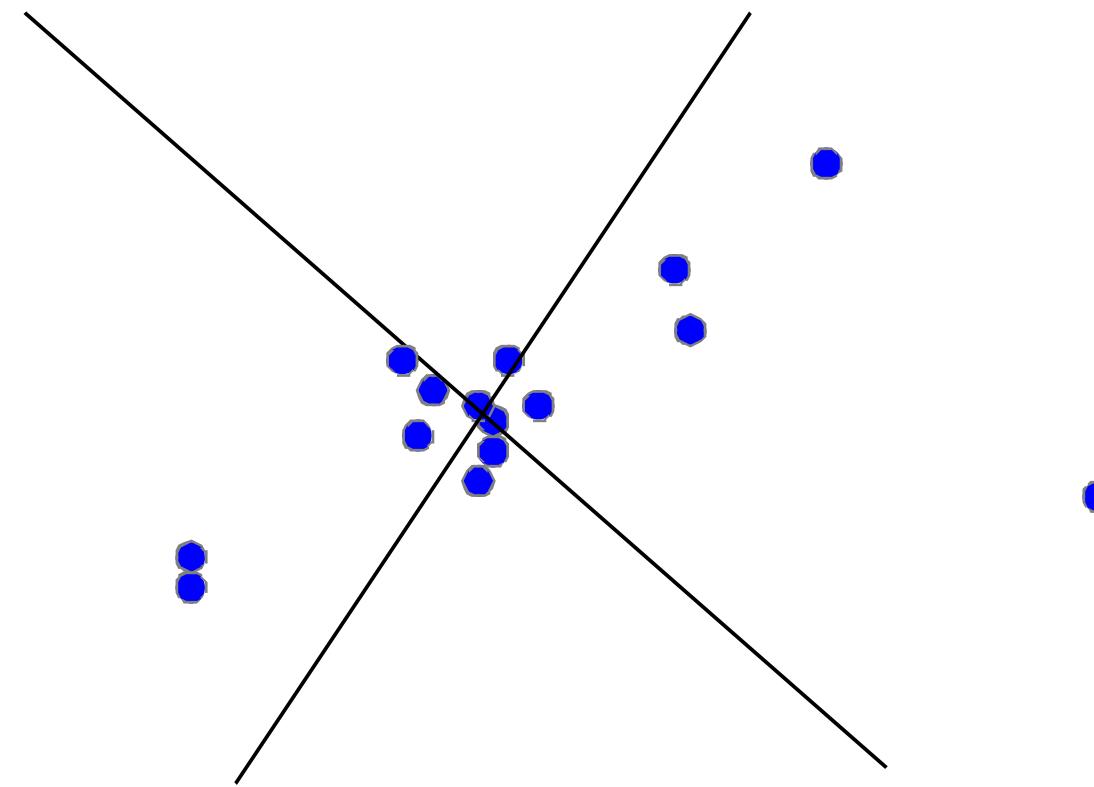


...эквивалентно
неверному

Маленький порог – проблема!



Проблема LMS



Нет хорошего
решения, если
выбросов >50%

Медиана ошибки одинакова
для обоих решений

Функции качества



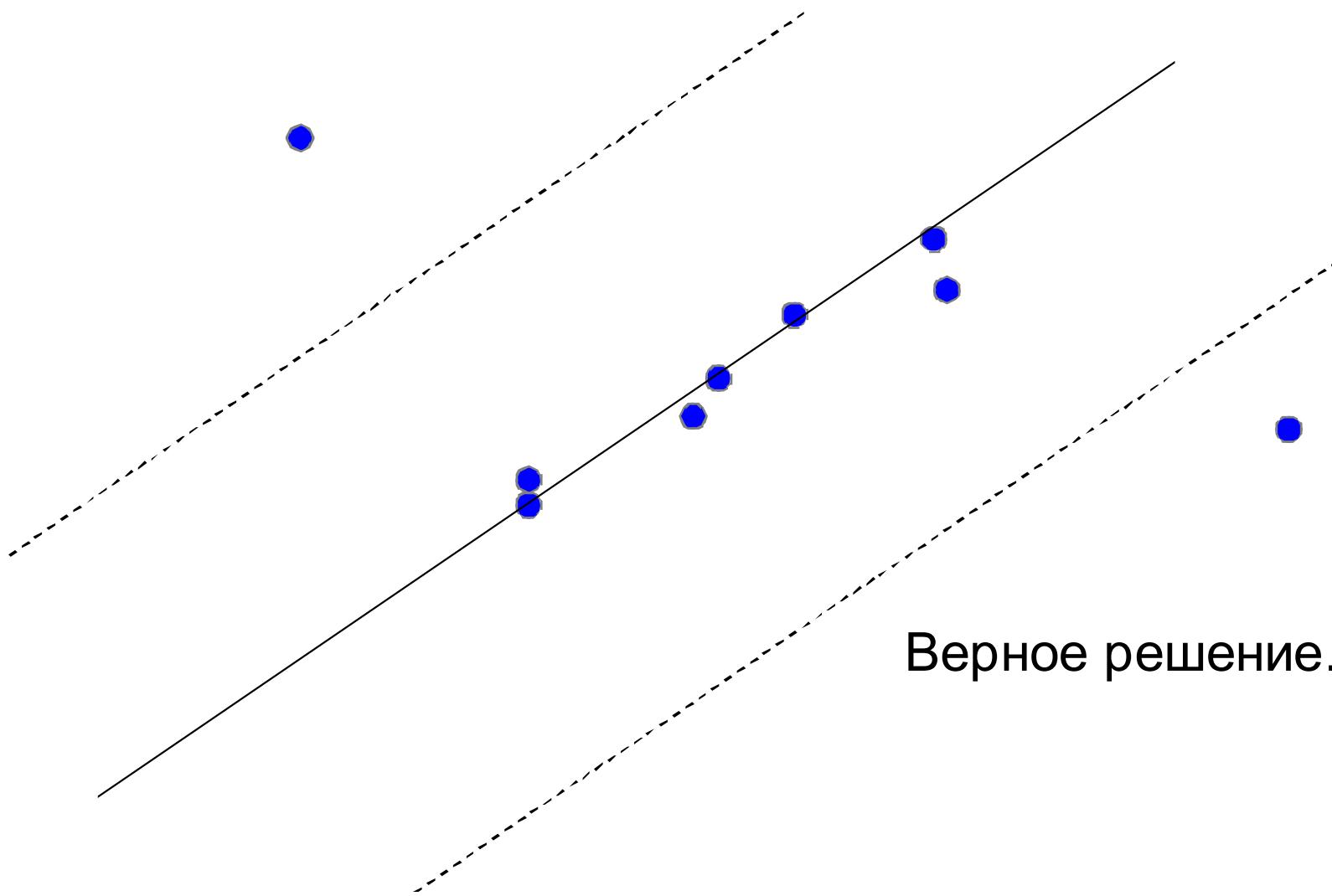
- **M-SAC**

- Возьмём робастную функцию, называемую М-оценкой, в качестве целевой функции:

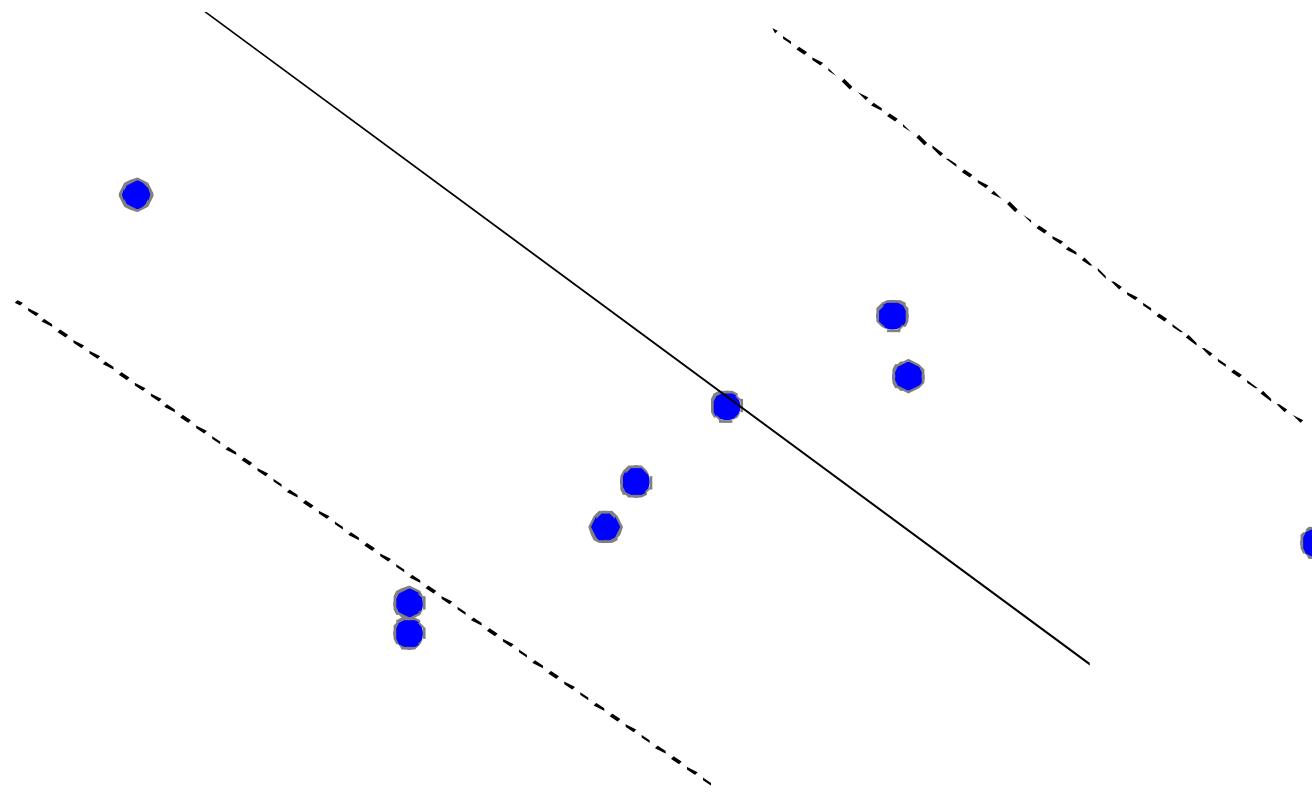
$$R(\theta) = \sum_i p(\varepsilon_i(\theta)^2), \quad p(\varepsilon_i^2) = \begin{cases} \varepsilon_i^2 & \varepsilon_i^2 \leq T^2 \\ T^2 & \varepsilon_i^2 > T^2 \end{cases}, i = 1, n$$

- **M-SAC** дает более точную оценку без увеличения вычислительной сложности

M-SAC



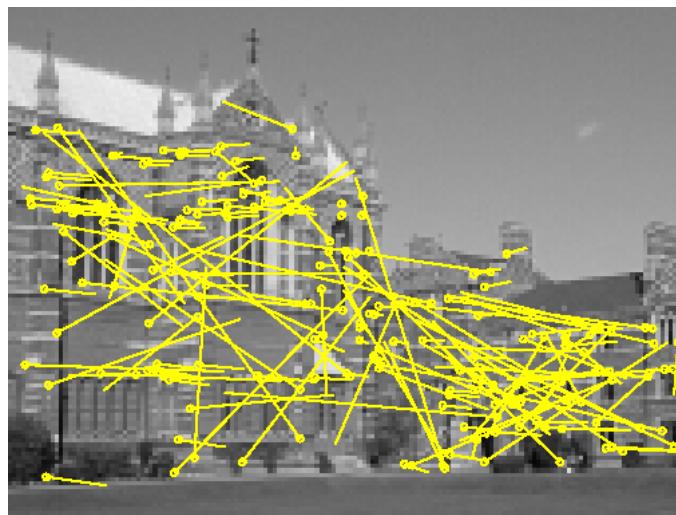
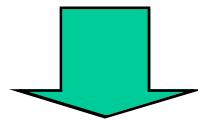
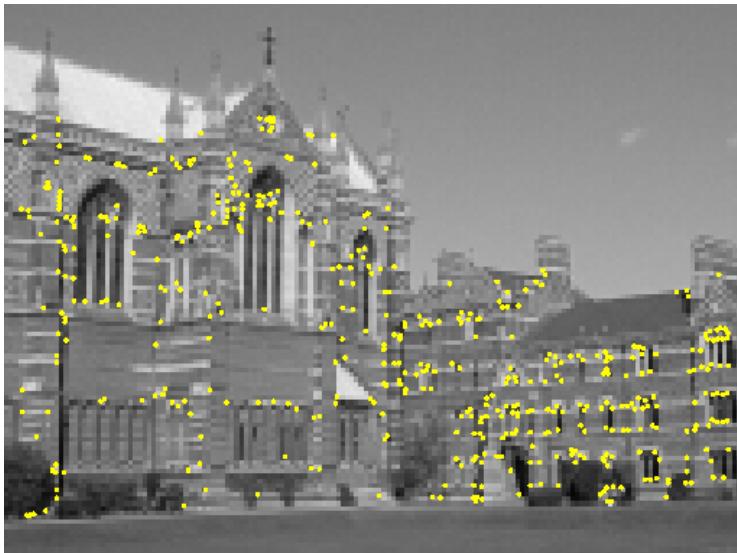
M-SAC



Лучше неверного



Пример использования



Сопоставление изображений

Сопоставление
особенностей по
дескрипторам –
много ложных

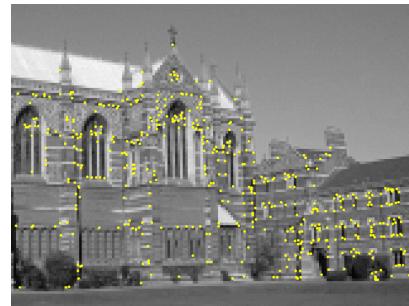
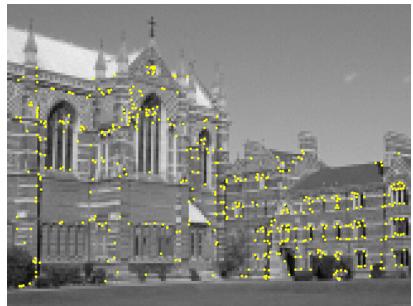


Алгоритм сопоставления

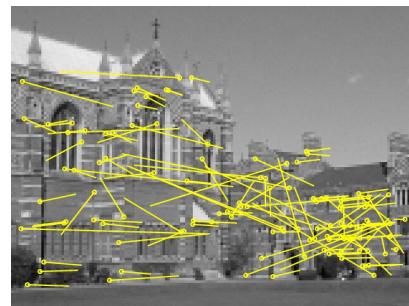
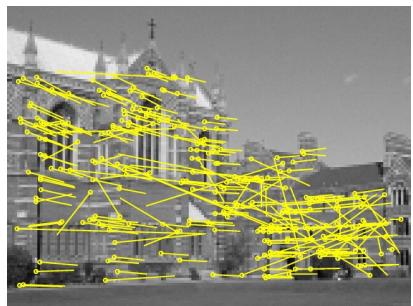
- Дано $\{(x, x')\}$ – набор пар соответствующих точек на изображениях I и I'
- Вычислим модель Θ эпиполярной геометрии между изображениями I и I' по парам соответствующих точек с помощью RANSAC
- Отфильтруем выбросы в $\{(x, x')\}$ по модели Θ
 - Если ошибка $r(x, x') > T$ – (x, x') - выброс
- Уточним модель Θ
 - Метод наименьших квадратов по всем Inliers
 - Пересчёт inliers и outliers
 - Итеративный пересчёт
 - Нелинейная оптимизация по всем данным с М-функцией в качестве целевой



Ложные соответствия

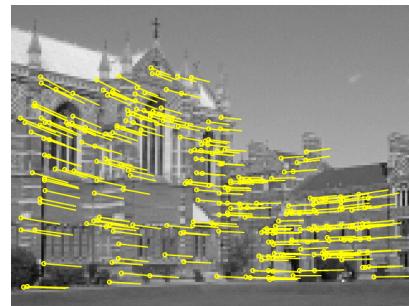
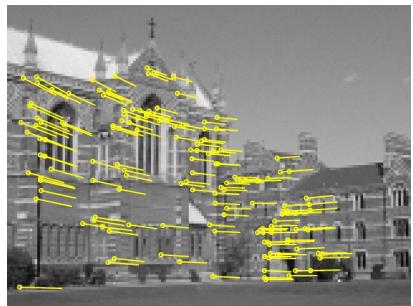


500 особенностей



Соответствия (268)

Выбросы (117)



Хороших соответствий
(151)



Калибровка камер

- Вычисление фундаментальной матрицы еще называют «слабая калибровка»
- Если известны матрицы калибровки двух камер, тогда мы можем вычислить существенную матрицу: $E = K^T F K'$
- Существенная матрица дает относительный сдвиг и поворот камеры (внешнюю калибровку)

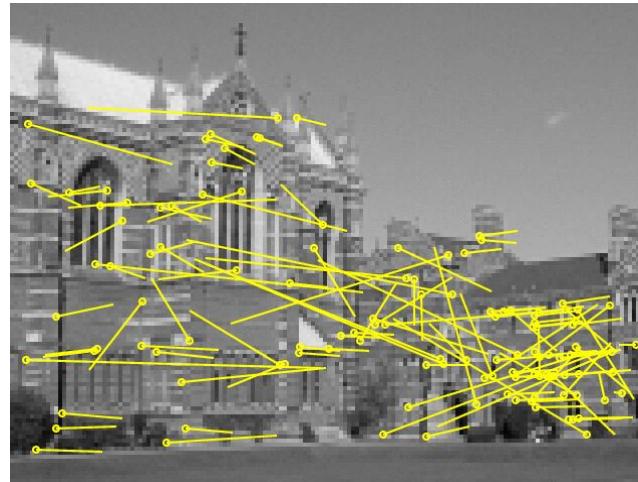
$$E = [t_x]R$$



Подзадачи

- **Калибровка камеры (PnP-проблема):** Дан набор 2D-3D соответствий. Определить параметры камеры.
- **Оценка движения камеры:** Дан набор соответствующих точек в 2x и более изображениях. Определить матрицы калибровки камер для этих изображений (видов)
- **Соответствие (стерео-сопоставление) :** Данна точка на одном изображении. Как мы можем ограничить положение соответствующей точки на втором изображении?
- **Геометрия сцены (структура) :** Пусть даны соответствующие точки (проекции) в 2D на 2x и более изображениях. Определить 3D координаты точки.

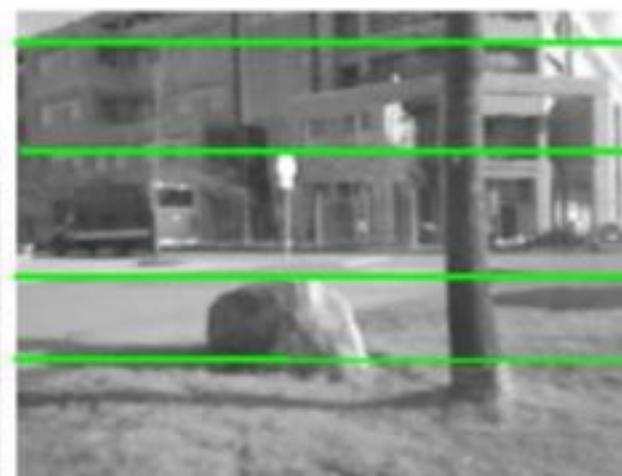
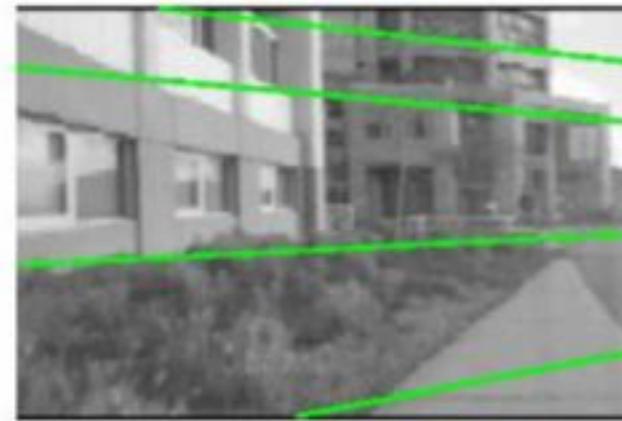
Упрощение стерео-сопоставления



- При сопоставлении локальных особенностей возникает много ошибок
- Можем robustно (RANSAC) оценить фундаментальную матрицу по множеству соответствий
- Используем фундаментальную матрицу для фильтрации ложных соответствий
 - Если ошибка переноса/репроекции больше порог - выброс

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

Упрощение стерео-сопоставления



Зная существенную матрицу, можно для любой точки на изображении вычислить соответствующую эпиполярную линию на другом изображении, и искать соответствие только на линии

Подзадачи

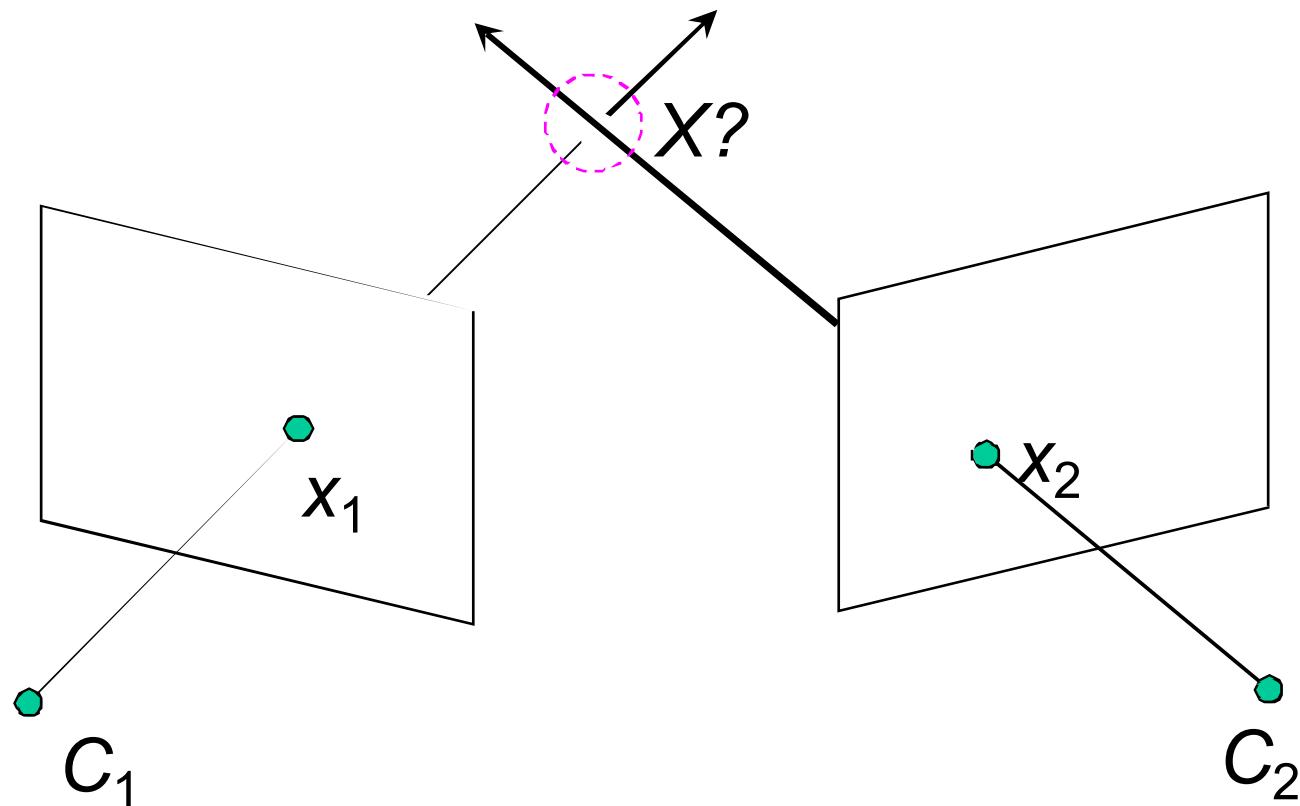


- **Калибровка камеры (PnP-проблема):** Дан набор 2D-3D соответствий. Определить параметры камеры.
- **Оценка движения камеры:** Дан набор соответствующих точек в 2x и более изображениях. Определить матрицы калибровки камер для этих изображений (видов)
- **Соответствие (стерео-сопоставление) :** Данна точка на одном изображении. Как мы можем ограничить положение соответствующей точки на втором изображении?
- **Геометрия сцены (структура) :** Пусть даны соответствующие точки (проекции) в 2D на 2x и более изображениях. Определить 3D координаты точки.

Триангуляция («структур» сцены)



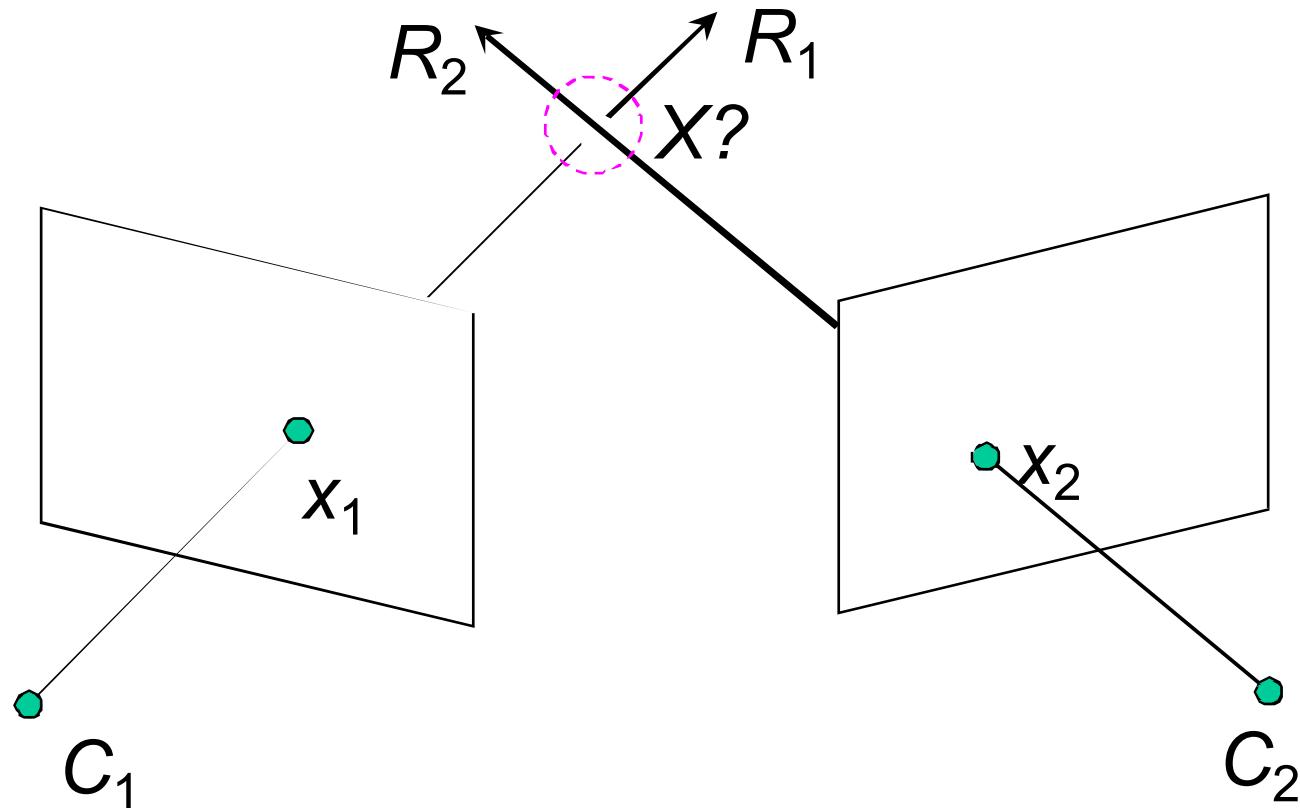
Даны проекции x_1, x_2 точки $X=(X, Y, Z)$ на 2 или более изображения (с известными матрицами калибровки), найти координаты точки



Триангуляция



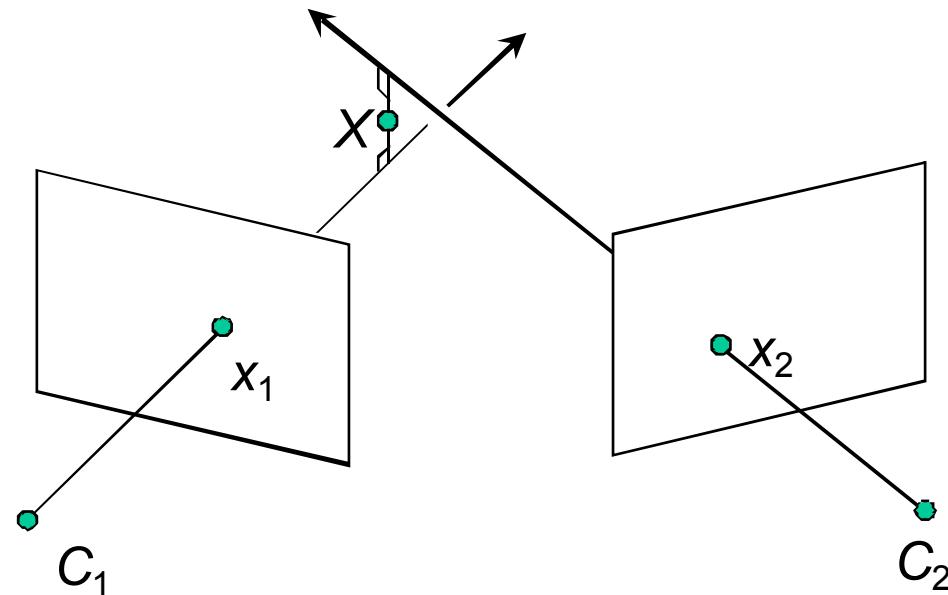
Можно найти пересечение лучей, соответствующих точкам x_1 и x_2 , однако из-за ошибок измерения и оценки они не пересекаются



Геометрический подход



- Найдем кратчайший отрезок, соединяющий два луча и пусть X будет средней точкой на этом отрезке
- Алгоритм средней точки (midpoint)



Линейный подход



Проекция на каждую камеру даст 2 уравнения на X:

$$\lambda_1 x_1 = P_1 X \quad x_1 \times P_1 X = 0 \quad [x_{1\times}] P_1 X = 0$$

$$\lambda_2 x_2 = P_2 X \quad x_2 \times P_2 X = 0 \quad [x_{2\times}] P_2 X = 0$$

$$x_1(p_1^{3T} X) - (p_1^{1T} X) = 0$$

$$y_2(p_1^{3T} X) - (p_1^{2T} X) = 0$$

Проекция на обе камеры
приведет к знакомой задаче на
однородные наименьшие
квадраты

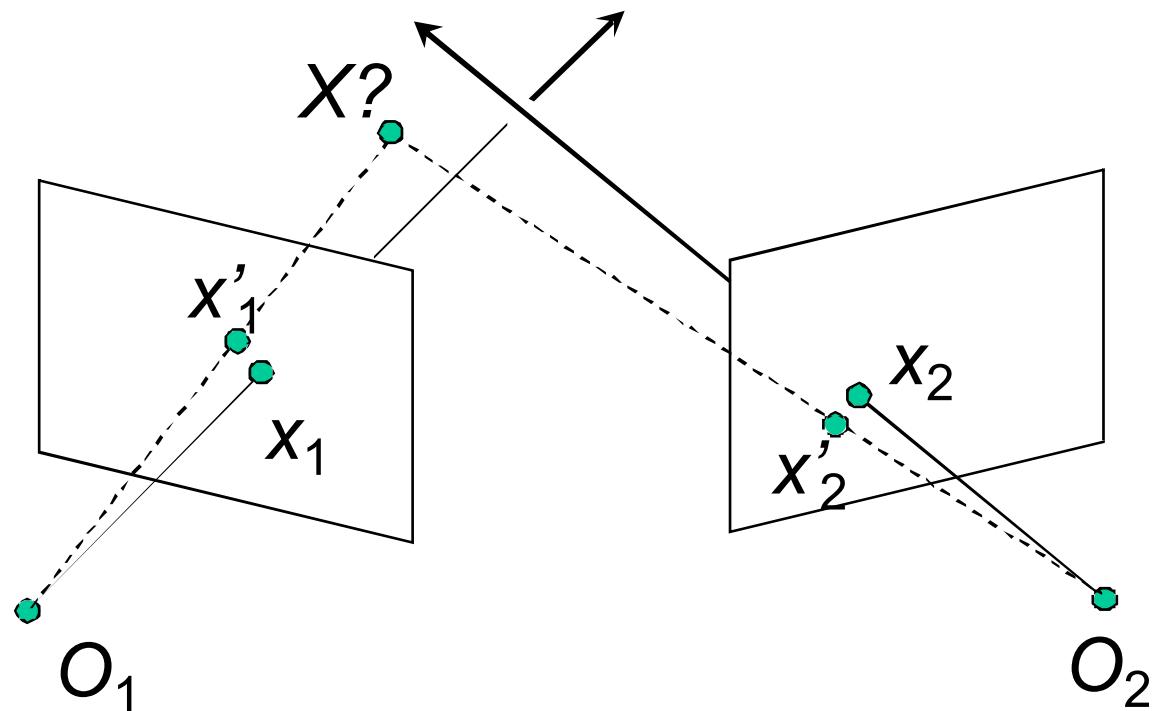
$$A = \begin{bmatrix} x_1 p_1^{3T} - p_1^{1T} \\ y_1 p_1^{3T} - p_1^{2T} \\ x_2 p_2^{3T} - p_2^{1T} \\ y_2 p_2^{3T} - p_2^{2T} \end{bmatrix} x = 0 \quad AX = 0 \quad \|X\| = 1$$

Метод «золотого стандарта»



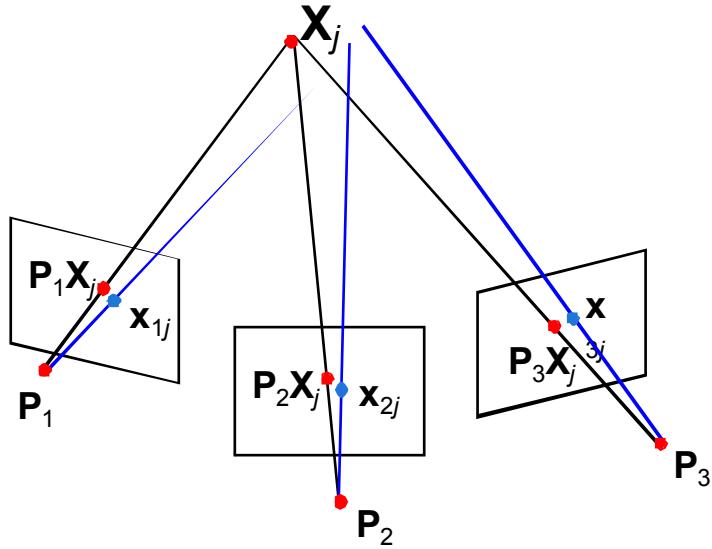
Найти точку X , минимизирующую ошибку репроекции:

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



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

Вернёмся к структуре и движению



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

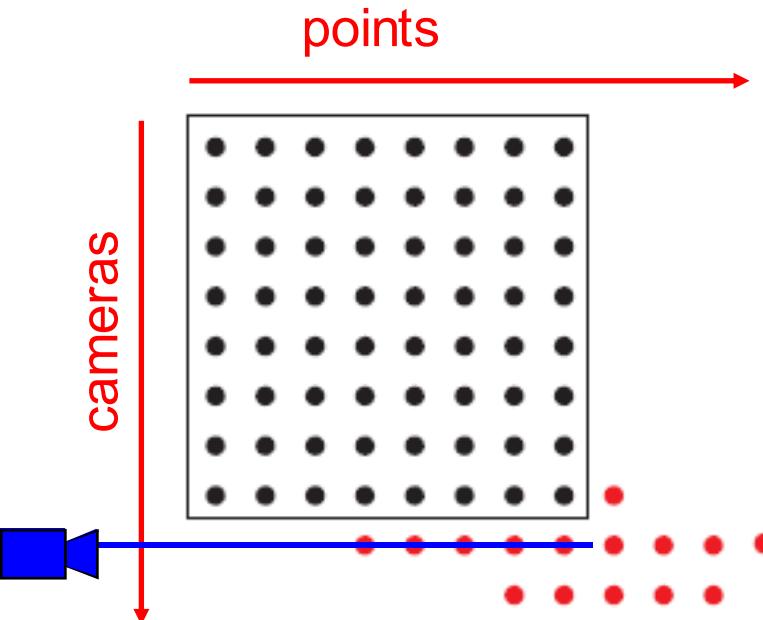
$$E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^m \sum_{j=1}^n D(\mathbf{x}_{ij}, \mathbf{P}_i \mathbf{X}_j)^2$$

- 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



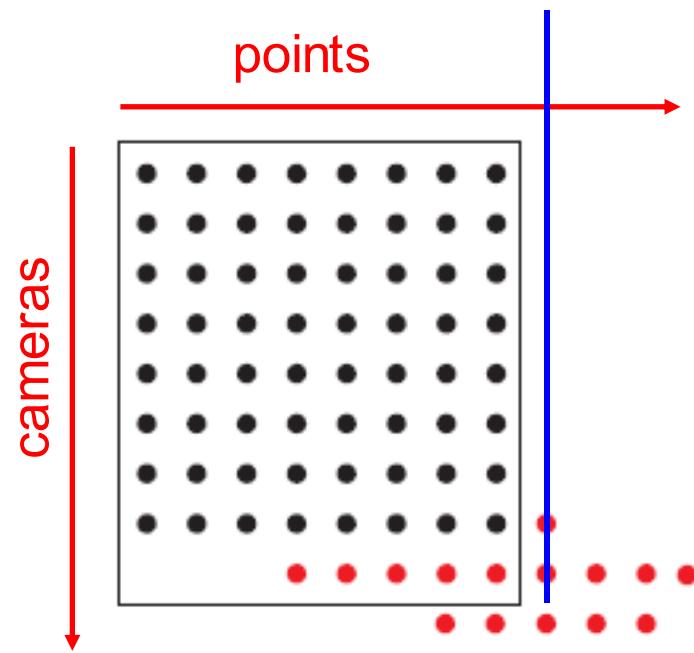
- Инициализируем движение через существенную матрицу двух камер
- Инициализируем структуру
- Для каждого вида:
 - Оцениваем матрицу проекции новой камеры по всем известным 3D точкам, видимым на этом изображении – калибровка



Последовательный подход



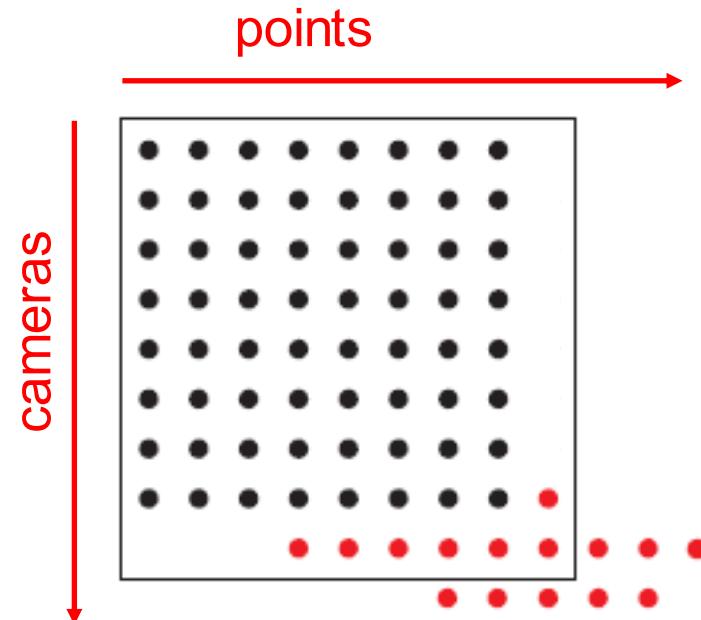
- Инициализируем движение через существенную матрицу двух камер
 - Инициализируем структуру
 - Для каждого вида:
 - Оцениваем матрицу проекции новой камеры по всем известным 3D точкам, видимым на этом изображении – калибровка
 - Уточняем и дополняем структуру, вычисляем новые 3D точки, уточняем существующие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



Последовательный подход



- Инициализируем движение через существенную матрицу двух камер
- Инициализируем структуру
- Для каждого 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

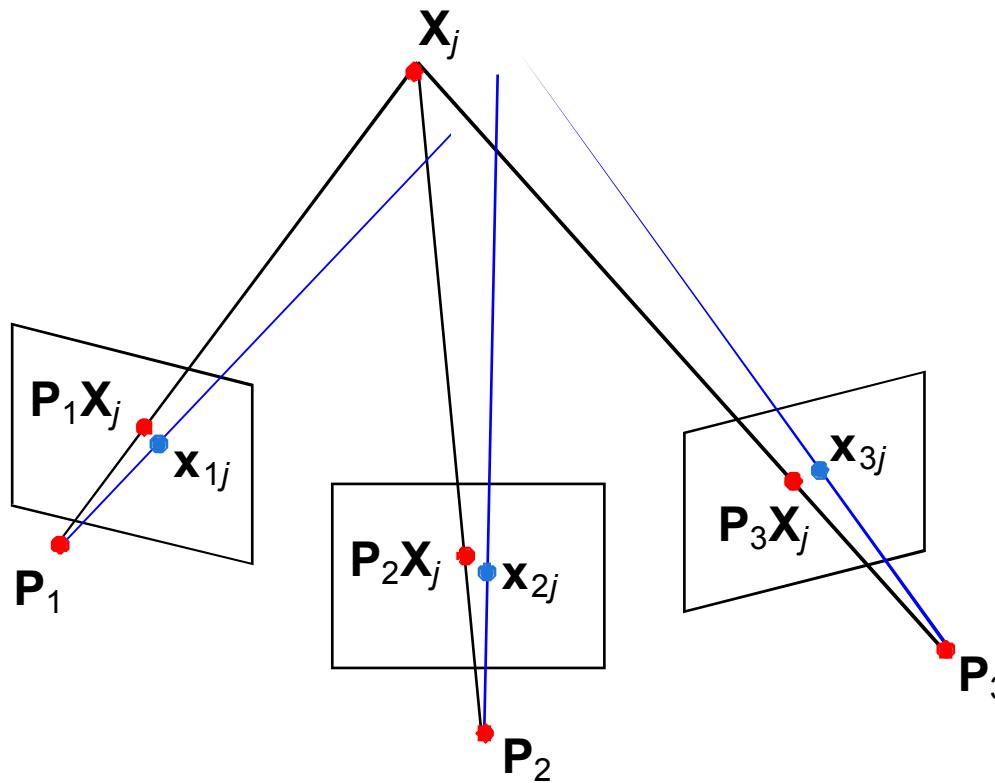


Метод связок (Bundle adjustment)



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

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

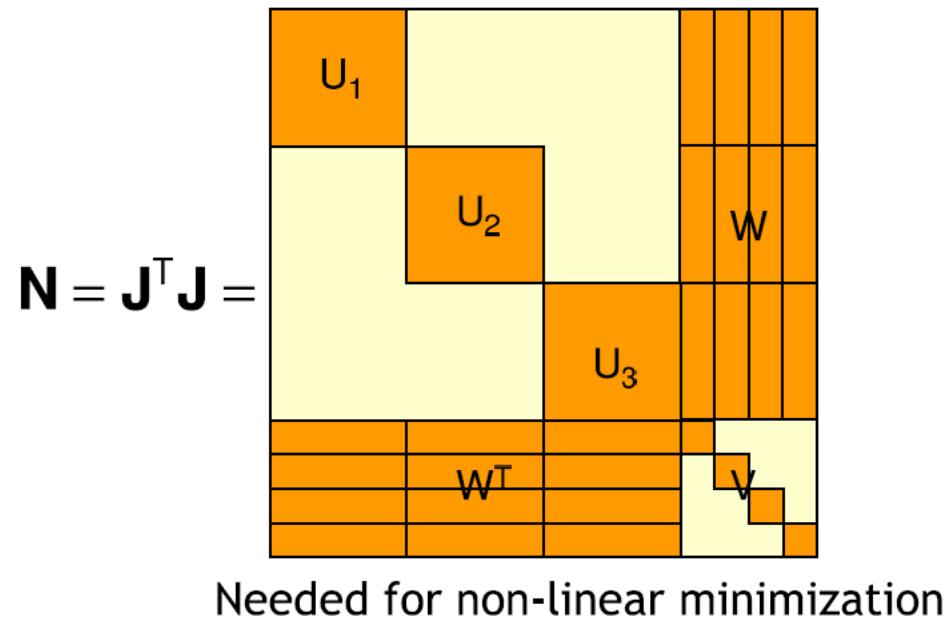
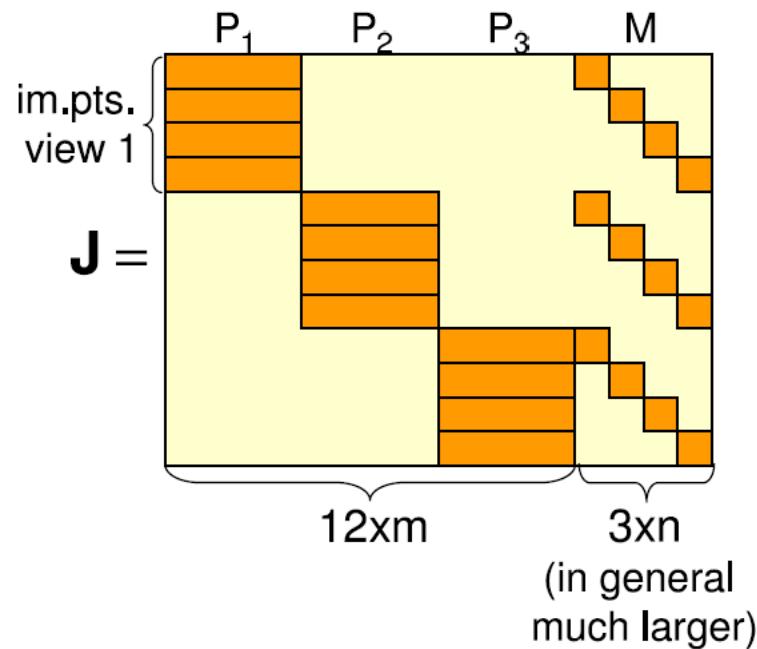


Итеративная оптимизация



- Имеем задачу $X = f(P)$, где X – вектор измерений, P – параметры, $f()$ – функция
- Есть начальное приближение $e_0 = f(P_0) - X$
- Предположим, что можно приблизить $f(P_0 + \Delta) \approx f(P_0) + J\Delta$
- Ищем P_1 , минимизирующее
$$f(P_1) - X = f(P_0) + J\Delta - X = e_0 + J\Delta$$
- Задача, найти минимум $\|e_0 + J\Delta\|$ по Δ
- Или $(e_0 + J\Delta)^T (e_0 + J\Delta)$
- Нужно решить нормальные уравнения $J^T J\Delta = -J^T e_0$

Блочная структура матриц



- Огромный размер матриц, но блочная структура
- Специальные реализации методов оптимизации для матриц такой формы
- Методы улучшения структуры матриц для повышения производительности

Неоднозначность решения



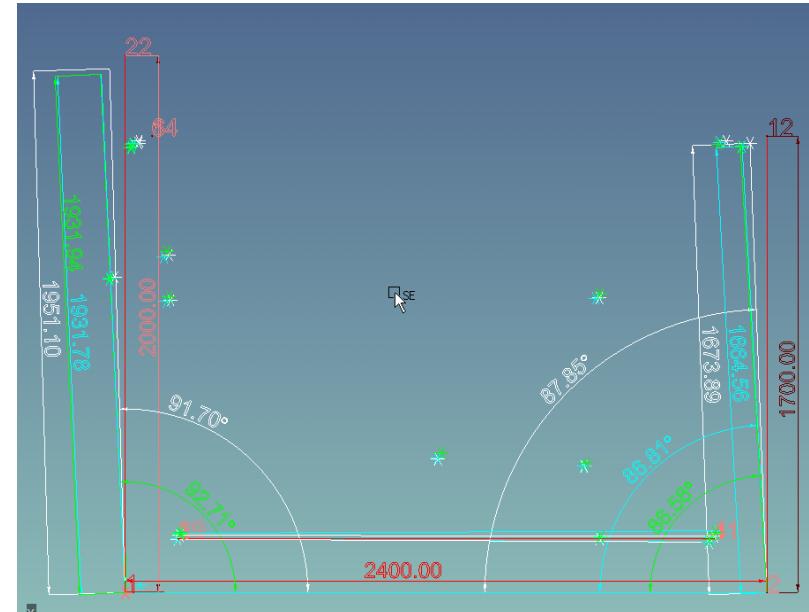
- Если мы умножим всю сцену на некоторый коэффициент k и в то же время умножим матрицы камер на $1/k$, проекции точек сцены на изображения не изменятся:

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

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

Нужна дополнительная информация

Фотограмметрия



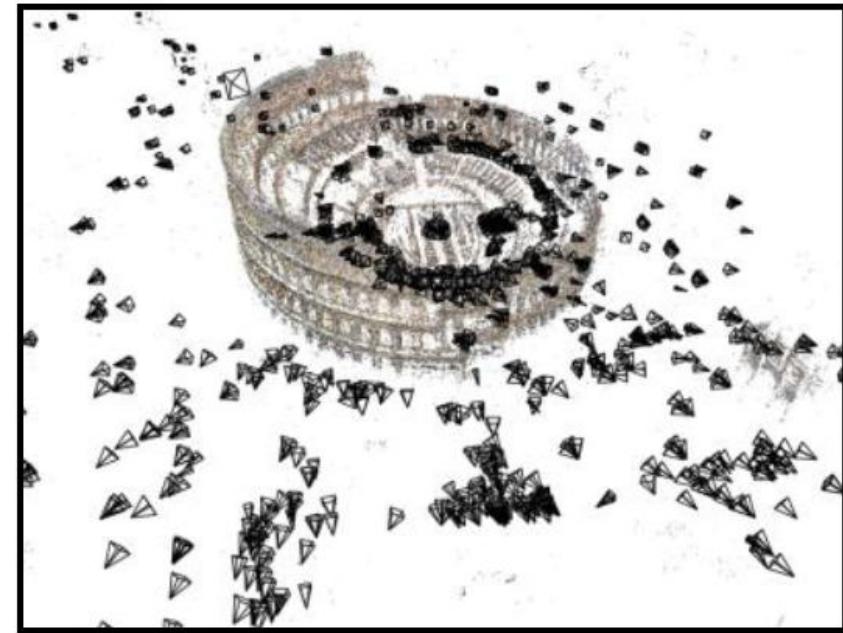
Чтобы провести измерения по изображениям (задача фотограмметрии), необходимо знать расстояние между минимум 2мя точками для нормализации сцены

План лекций



- Цифровой фотоаппарат
- Модель перспективной камеры
- Многовидовая геометрия
 - Гомография
 - Эпиполярная геометрия
- «Структура из движения» и подзадачи
- **Применение – строим Рим за 24 часа**

Большая задача



Коллекция изображений
из интернета

Трёхмерная модель

В 24 часа...

Коллекции изображений



flickr® from YAHOO!

Signed in as Jimantha  

Home You Organize & Create Contacts Groups Explore

Search Photos Groups People

Everyone's Uploads colosseum **SEARCH** Full Text | Tags Only Advanced Search

Sort: **Relevant** | Recent | Interesting

View: **Small** | Medium | Detail | Slideshow 

						
From Joanol	From Aquillant	From Gustav...	From Stuck in...	From panos_r	From Historicus	From @ Lude's...
						
From WTL photos	From 'Toshio'	From Aquillant	From 'Toshio'	From simone pelle...	From beldez	From [-Bryan-]
						
From david.bank...	From Raphael...	From Luismaxx	From Mark.WV	From bekahpage	From James Rye	From psioniks

Несколько миллионов изображений по тегу
«Rome»

Обычные фотографии



- Получены разными пользователями с разных фотоаппаратов



iPhone 3G

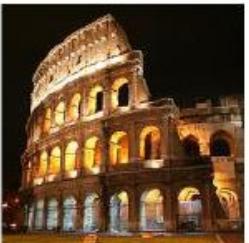


Nikon D3

Обычные фотографии



Colosseum Rome
from oksana...



The Colosseum, Rome
from martin97uk



Italy - Blue Hour & Trails of...
from 9_Lucids_Photography ...



Colosseum - Rome, Italy
from Ingrid Siljeus



Rome Colosseum
from david.lean (www.david-lean.com)



Colosseum, Rome
from Amnon



Colosseum, Rome
from Giampaolo Macor



The Colosseum, Rome
from Christopher Chan



Colosseum Rome
from orlando72



Rome Italy Colosseum
from Locoss



Rome Colosseum
from scott.photos



Colosseum at night, Rome, DSC_5108
from paulhydros



Rome - Colosseum
from mr_center1



Colosseum, Ancient Rome
from "Ana_Medved" (aka) ouay!



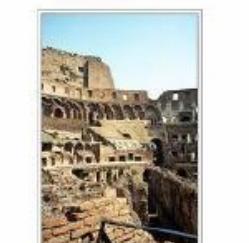
THE COLOSSEUM, Rome, Italy (i)
from tom_m_wag



THE COLOSSEUM, Rome, Italy (ii)
from tom_m_wag



INSIDE THE COLOSSEUM, Rome, Italy ...
from sl_vpg



INSIDE THE COLOSSEUM, Rome, Italy ...
from sl_vpg

- Разное время года, разное время суток
- Случайный порядок изображений

Загрузим из интернета



Code for querying and downloading Flickr images - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://graphics.cs.cmu.edu/projects/im2gps/flickr_code

Code for finding and downloading images on Flickr

Due to recent changes in the Flickr API, the archive below contains a newer version that no longer uses negative text constraints during the query to avoid duplicate images. The search algorithm is slightly different from previously posted versions of this code.

This code was used to generate the data set for

IM2GPS: estimating geographic information from a single image

and an earlier version was used for

[Scene Completion Using Millions of Photographs](#)

The code is written in Python and Matlab and tested in Linux.

[Link to code.](#) (3.1 MB .zip)

Below is the readme which is included with the code.

This code was originally written by Tamara Berg, then extended by James Hays (jhheys@cs.cmu.edu)

An older version of this code was used to generate the geotagged database for IM2GPS: estimating geographic information from a single image.
James Hays and Alexei A. Efros. CVPR 2008.

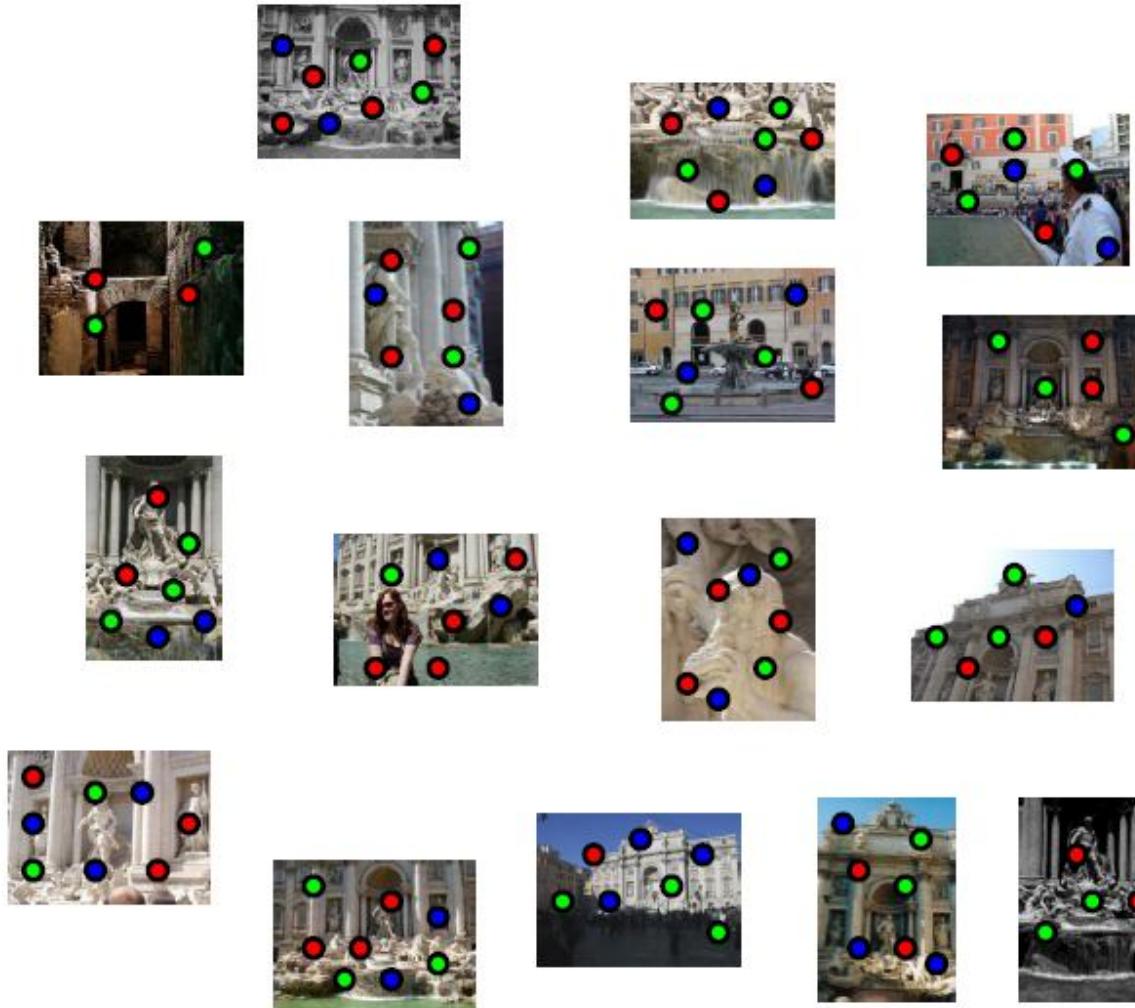
Done

http://graphics.cs.cmu.edu/projects/im2gps/flickr_code.html

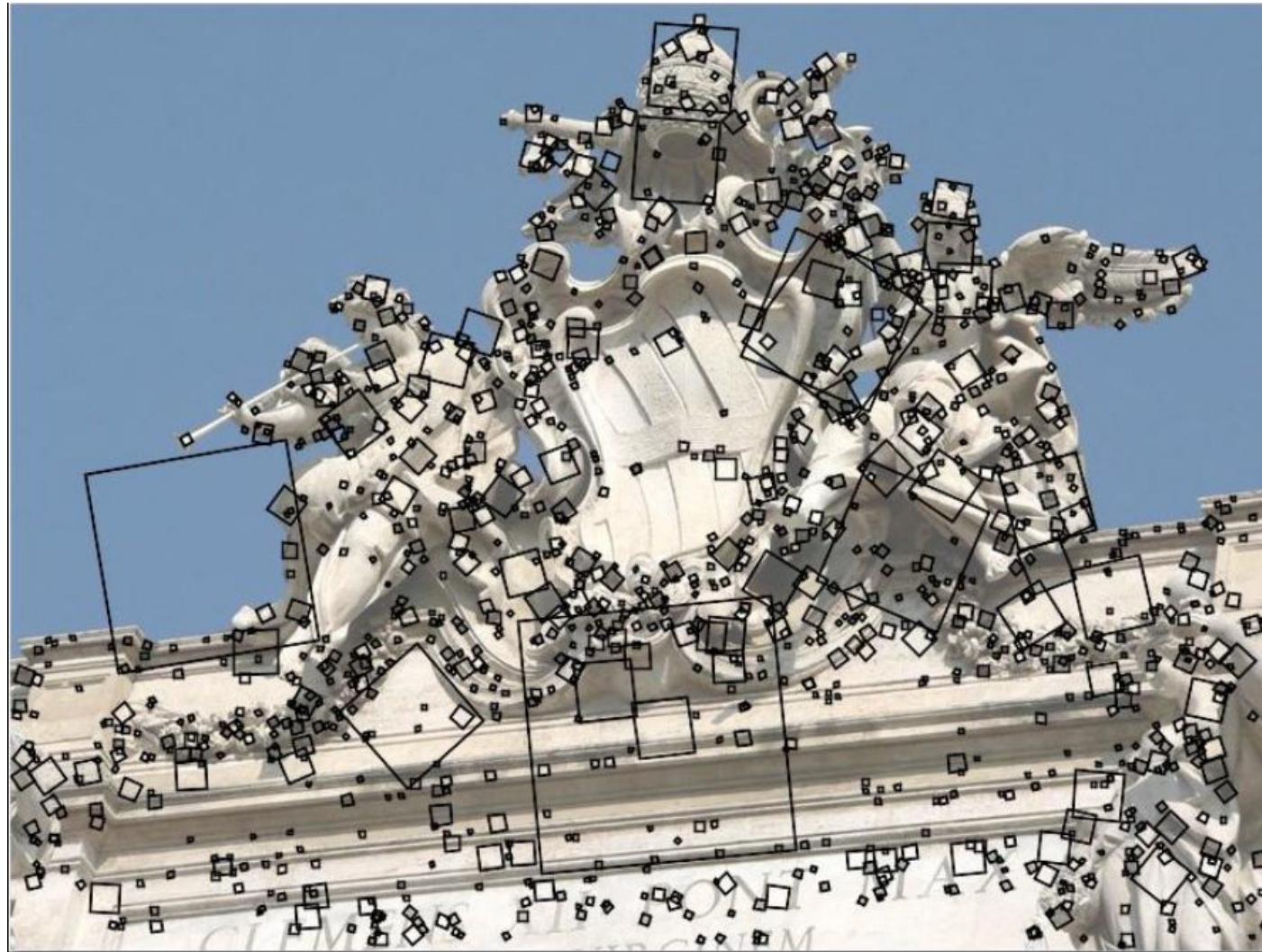
Найдем соответствующие точки



Поиск особых точек (SIFT)



Найденные особые точки

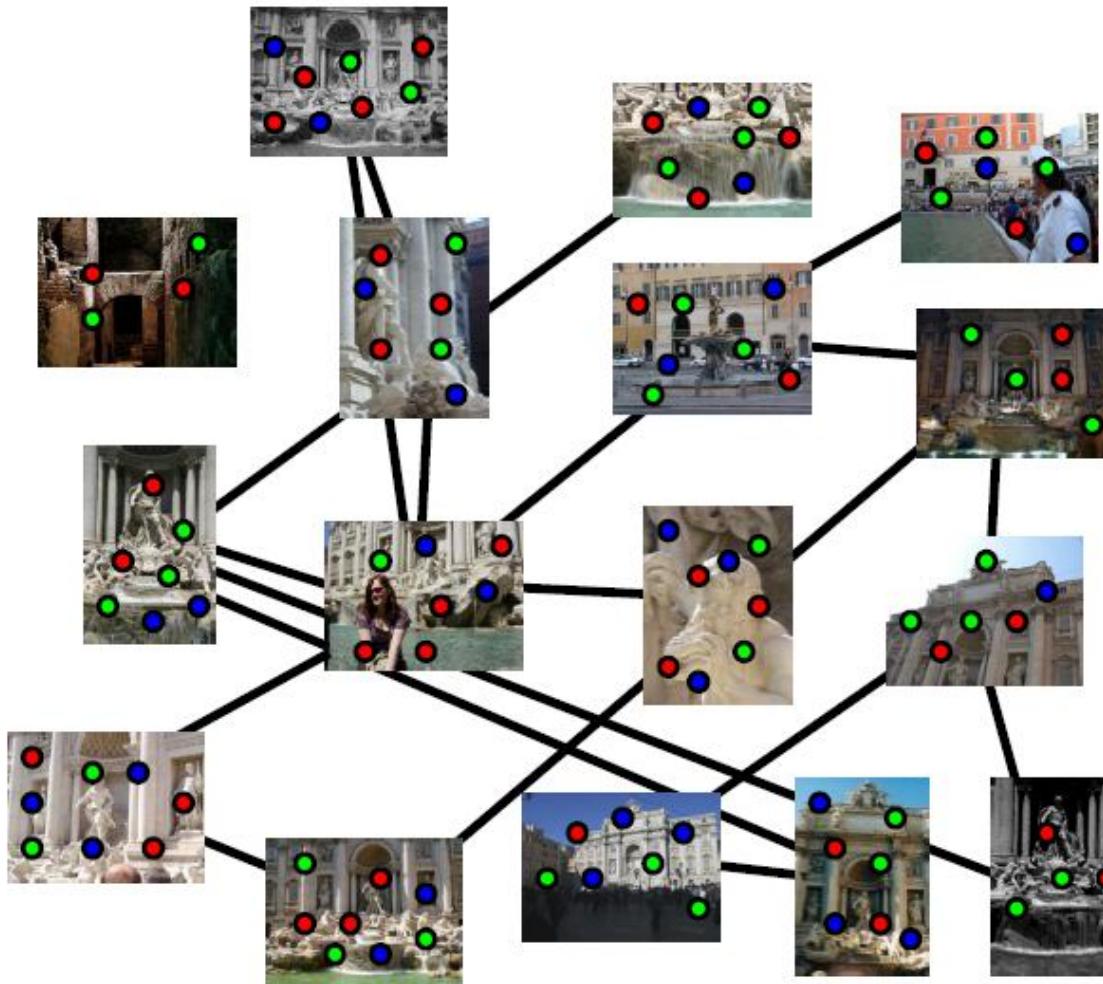


Сопоставление изображений



- Сопоставляем точки по дескрипторам SIFT
- Фильтруем ложные соответствия по эпиполярной геометрии
- Для этого будем с помощью RANSAC вычислять фундаментальную матрицу
- Пары, между которыми много соответствий, будем считать «связанными»

Граф связности изображений



Объем задачи

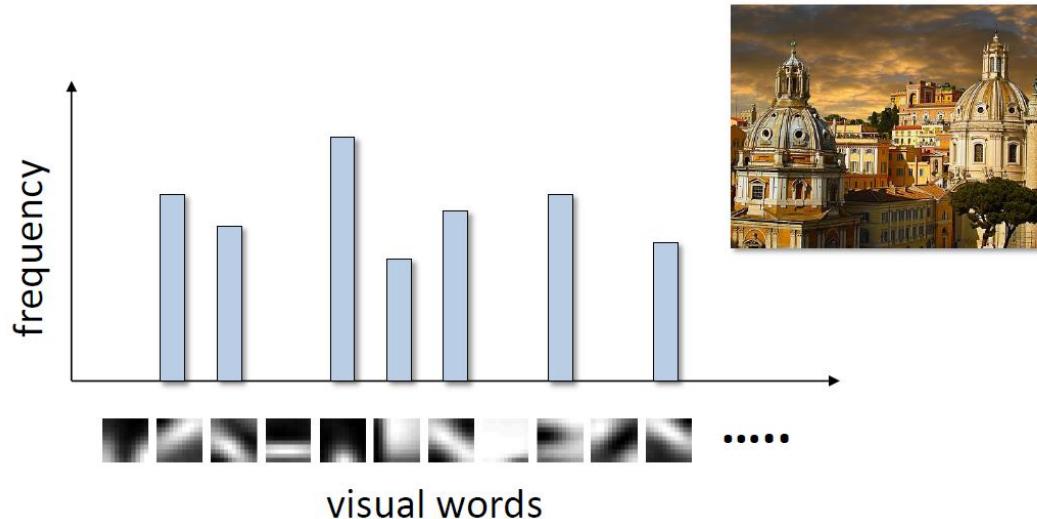


- 250K изображений -> 31M пар изображений
- 2 пары изображений в секунду
 - 1 год на 500 машинах
- 1M изображений -> 500 000M пар изображений
 - 15 лет на 500 машинах

Поиск похожих изображений



- Метод «Мешок слов»
 - Обучаем словарь визуальных слов размером 100К
 - Строим гистограмму частот для каждого изображения:



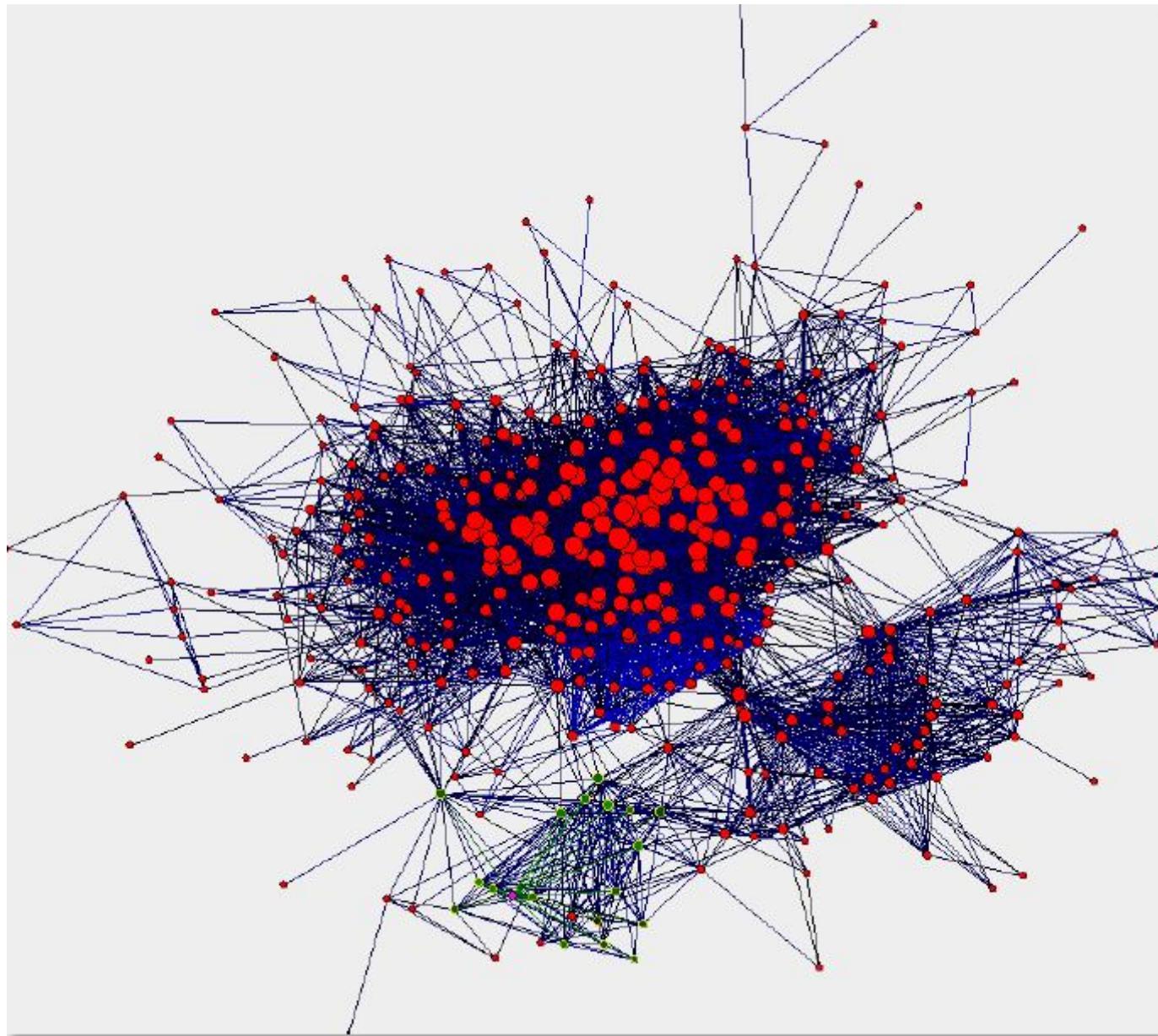
- Для каждого изображения ищем 40 наиболее похожих
- Сопоставляем точки между этими парами изображений
- Выбираем хорошие пары

Результат сопоставлений

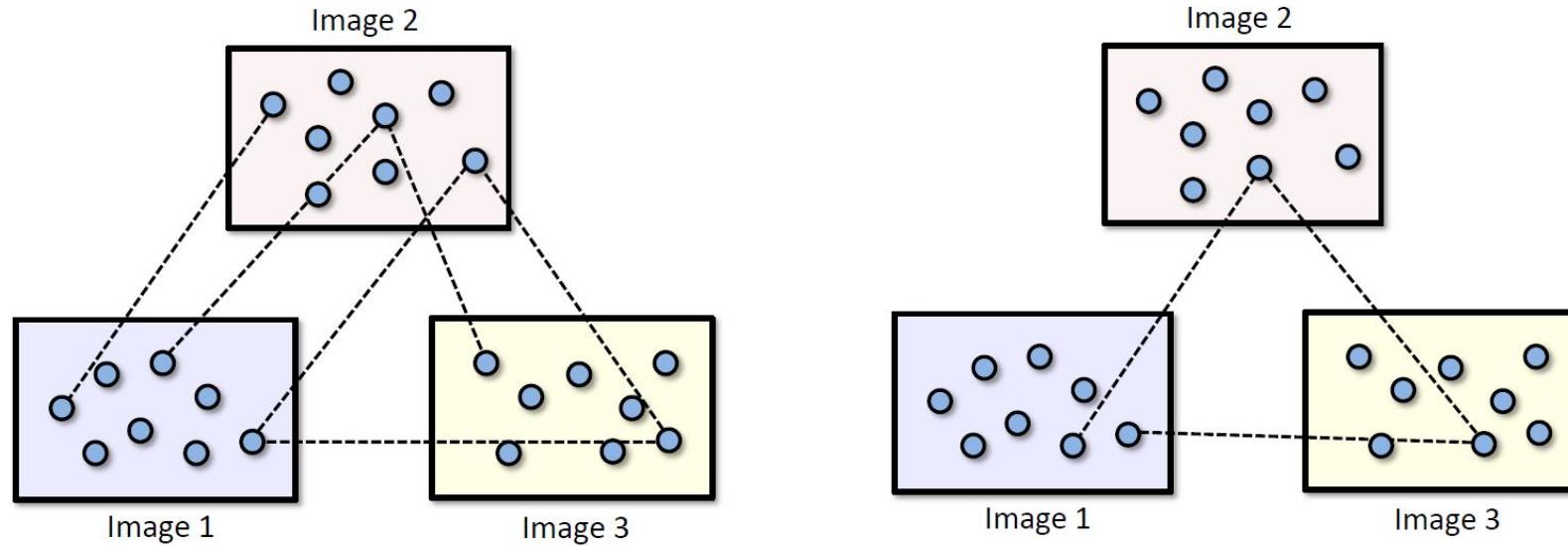


Мощность SIFT!

Граф связности изображений

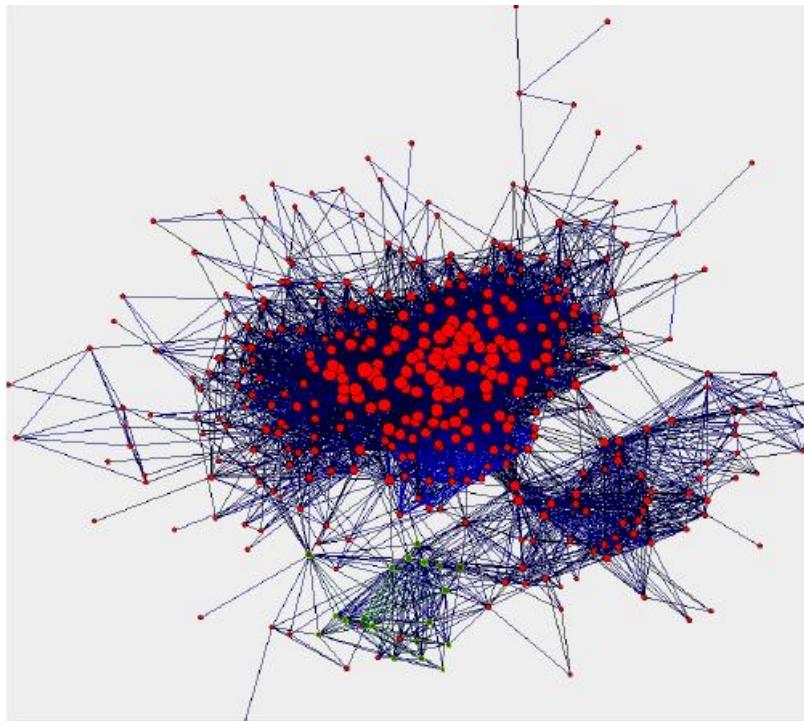


Построение «следов»

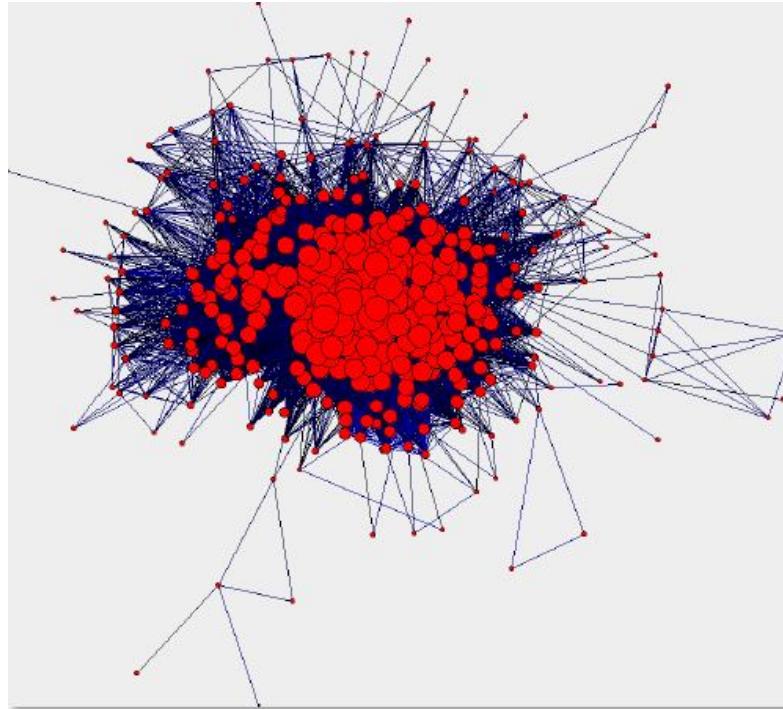


- Объединяем соответствия в «следы»
 - Отбрасываем незамкнутые следы, как потенциальные ошибки
- Пример (3000 изображений), всего 1.5M+ следов
 - 79% длины 2
 - 90% длины ≤ 3
 - 98% длины ≤ 10
 - Самый длинный след 385 точек

Граф связности изображений

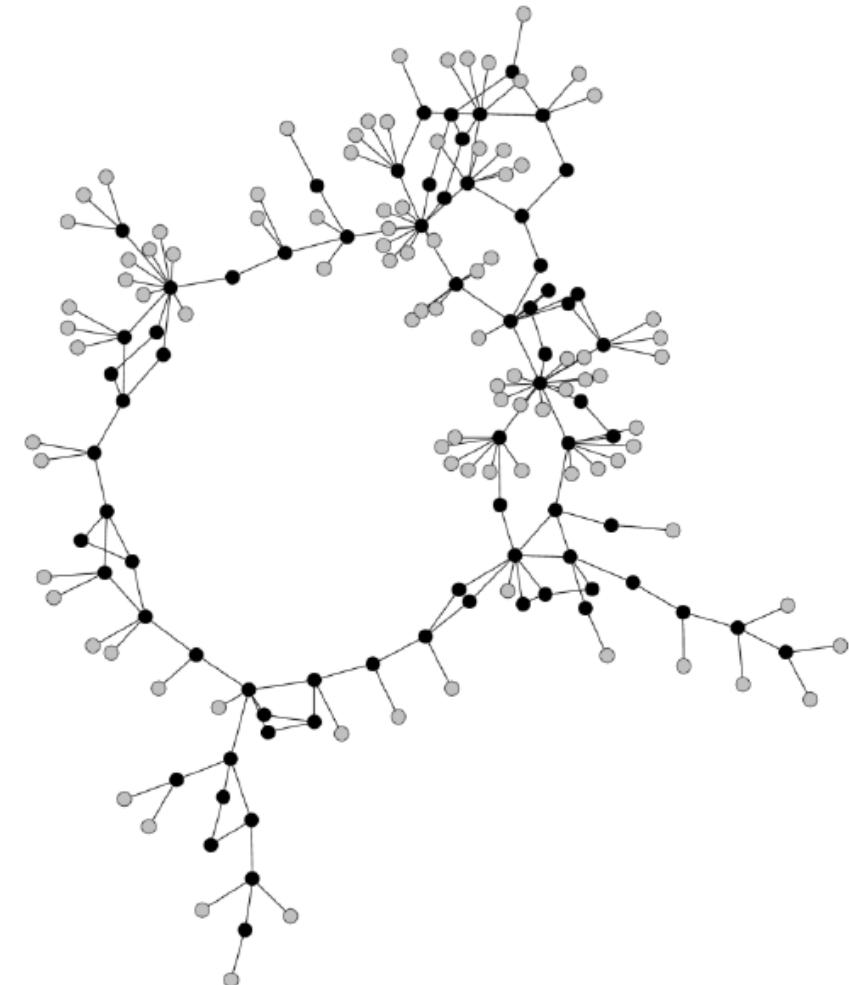
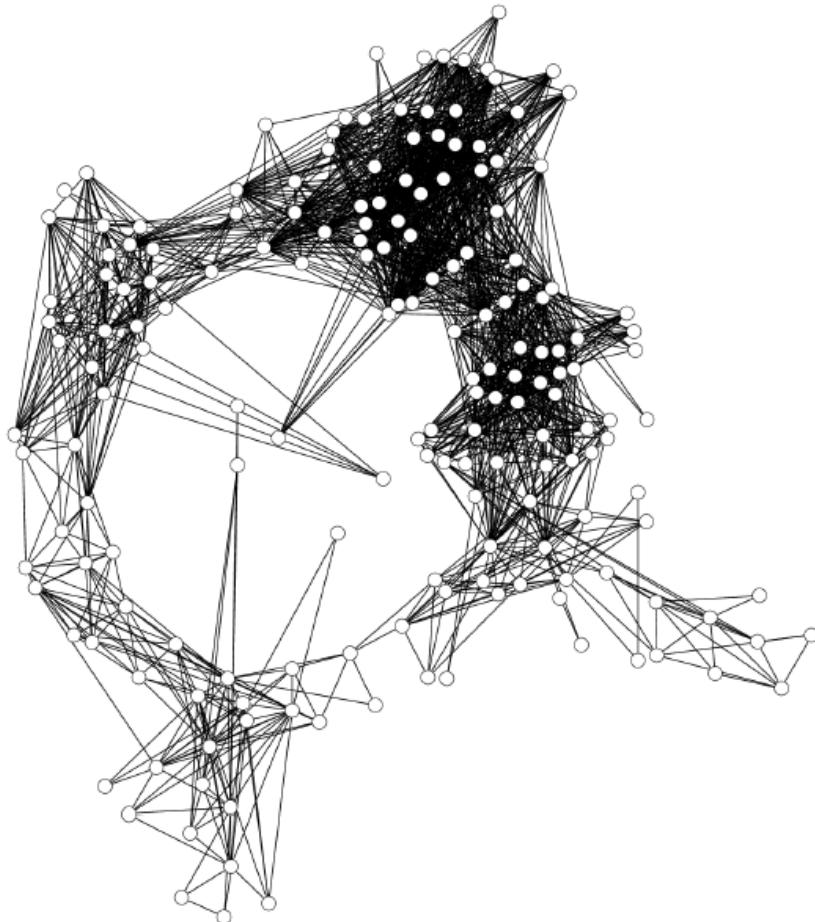


Исходные
соответствия



После построения следов

«Скелет» графа



Фильтруем изображения, оставляя только «ключевые» для упрощения последующей реконструкции

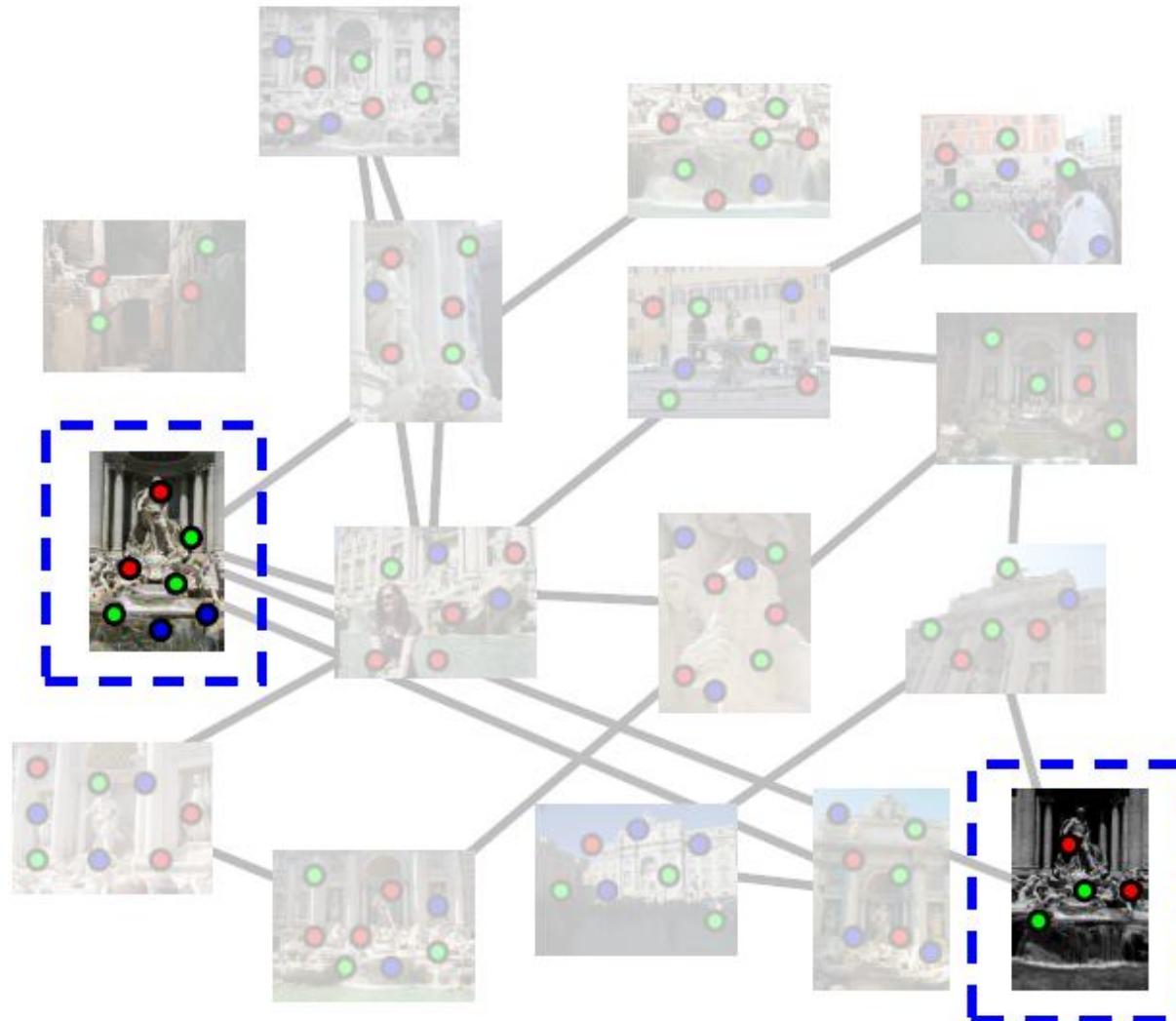
[Snavely, et al., “Skeletal sets for efficient structure from motion,” CVPR 2008]

Стандартная схема



- Выбор опорных пар изображений
- Вычисление структуры по паре изображений
 - Построение матрицы внутренней калибровки
 - Вычисление фундаментальной и существенной матрицы
 - Триангуляция точек
 - Уточнение методом связок
- Добавление новых изображений
 - Калибровка камеры по известным 2D/3D соответствиям
 - Триангуляция новых точек
 - Уточнение методом связок

Выбор опорной пары



Выбор опорной пары



(+) много точек



(-) малая база



(+) большая база



(-) мало точек



(+) много точек



(+) большая база

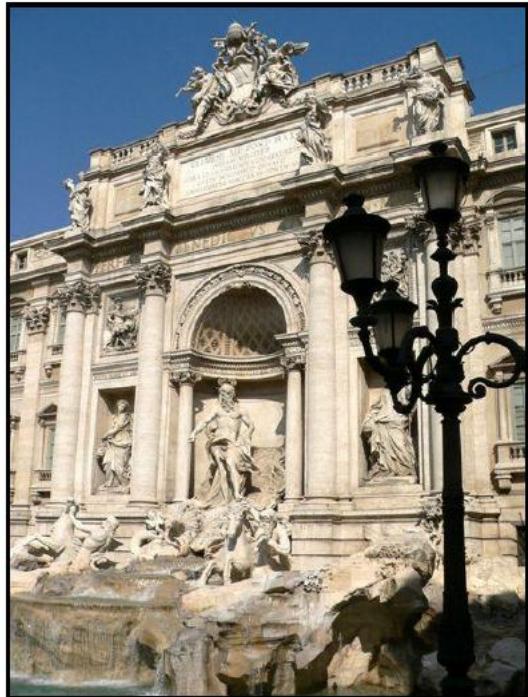
Выбор опорной пары



- Возможны разные эвристики
- Гомография плохо описывает сцену, когда сцена не плоская, и существенный параллакс (большая база)
- Пример:
 - Возьмем пары, в которых > 100 соответствий
 - Вычислим гомографию и фундаментальную матрицу
 - Выберем пару изображений, минимизирующую отношение:

$$\frac{\text{numInliers}_{\text{homography}}}{\text{numInliers}_{\text{Fmatrix}}}$$

Внутренняя калибровка



File size : 85111 bytes
File date : 2005:12:16 04:17:12
Camera make : **Panasonic**
Camera model : **DMC-FZ20**
Date/Time : 2005:03:19 12:52:33
Resolution : 450 x 600
Flash used : No
Focal length : **6.0mm**
Exposure time: 0.0012 s (1/800)
Aperture : f/5.6
ISO equiv. : 80
Whitebalance : Auto
Metering Mode: matrix
Exposure : program (auto)

Структура EXIF-данных

Параметры камеры



Digital Photography Review
dpreview.com™

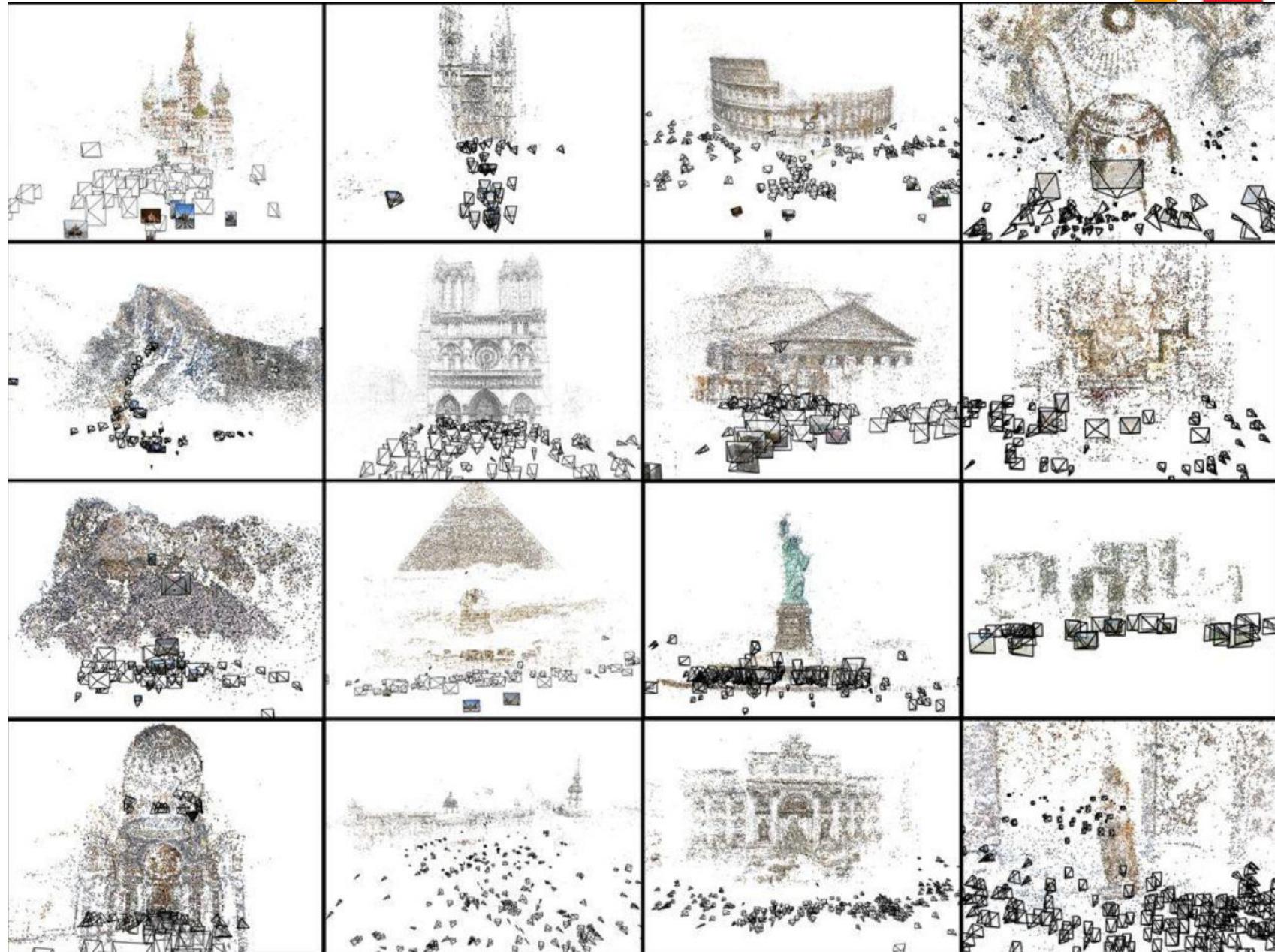
search

Latest News
Reviews / Previews
Lens Reviews
Camera Database
Timeline
Buying Guide
Sample Galleries
Challenges
Discussion Forums
Learn / Glossary
Feedback
Newsletter
Blog
Feeds
About

Panasonic Lumix DMC-FZ20 digital camera specifications

Panasonic Lumix DMC-FZ20	
Image	
More information	<ul style="list-style-type: none">In-depth reviewSamples galleryAnnounced 21-Jul-04All Panasonic productsPanasonic website
Discussion	<ul style="list-style-type: none">Panasonic Talk ForumFind related discussion
Owners opinions	 Read owners opinions (201) Post / Edit your opinion
Format	Compact, SLR-like
Price (street)	Sensor size 1/2.5 " (5.75 x 4.31 mm, 0.24 cm²)
Also known as	
Release Status	Discontinued
Max resolution	2560 x 1920
Low resolution	2048 x 1536, 1920 x 1080, 1600 x 1200, 1280 x 960, 640 x 480
Image ratio w:h	4:3, 16:9
Effective pixels	5.0 million
Sensor photo detectors	5.3 million
Sensor size	1/2.5 " (5.75 x 4.31 mm, 0.24 cm²)
Pixel density	20 MP/cm²
Sensor type	CCD

http://www.dpreview.com/reviews/specs/Panasonic/panasonic_dmcfz20.asp



Dubrovnik, Croatia.

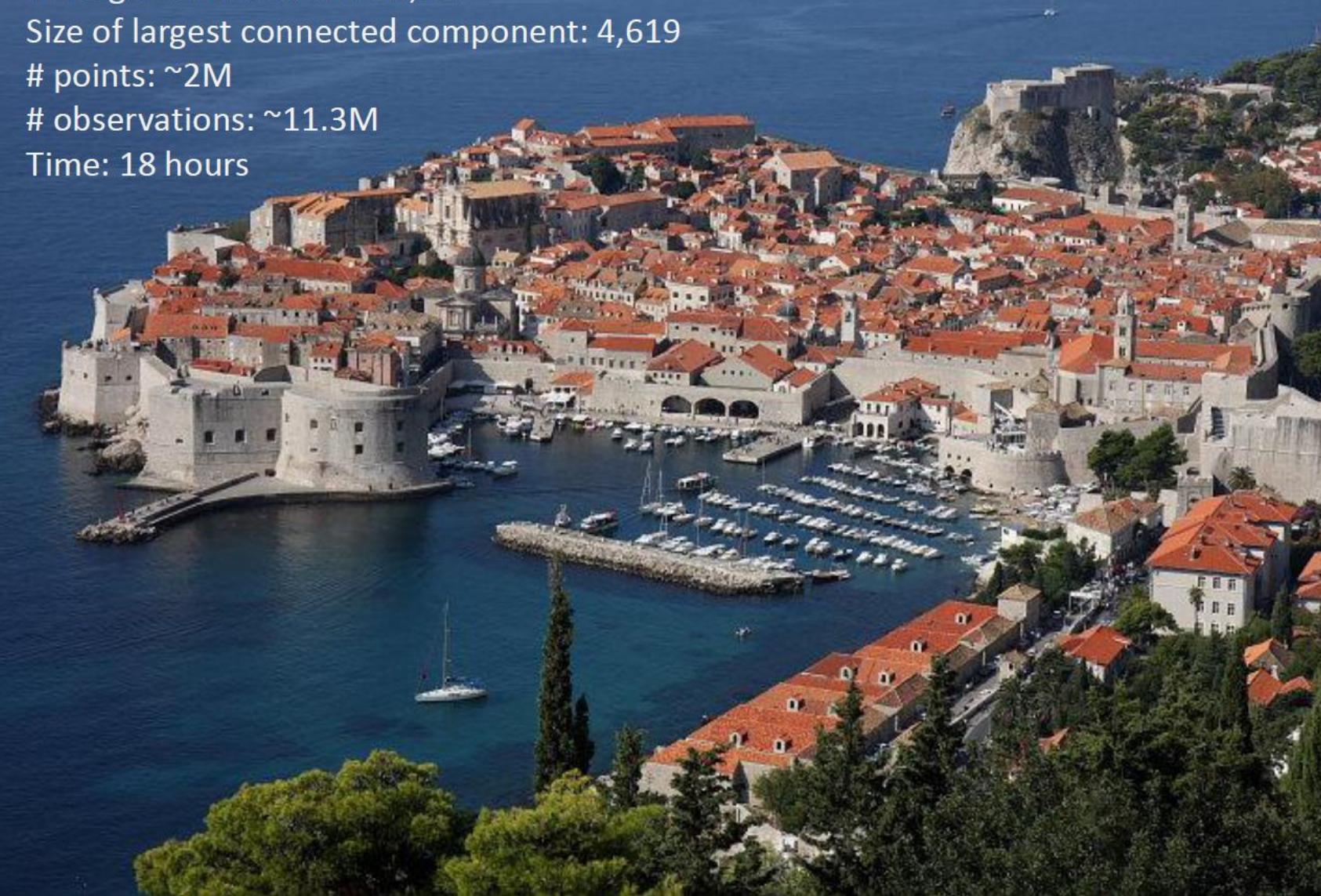
images downloaded: 57,845

Size of largest connected component: 4,619

points: ~2M

observations: ~11.3M

Time: 18 hours



Дубровник





BigSFM: Reconstructing the World from Internet Photos

Our group is working on building a 3D model of the world from online photo collections, and our research spans several areas, from image features, to large-scale image matching, to structure-from-motion optimization, to applications such as location recognition. This page summarizes our work, has links to [code](#) and [datasets](#) we have made available, and has a description of each [project](#).

Cornell Participants (and alumni): [Noah Snavely](#) (Cornell), [David Crandall](#) (Indiana University), [Daniel Hauagge](#) (Cornell), [Kyle Wilson](#) (Cornell), [Song Cao](#) (Cornell), [Yin Lou](#) (Cornell), [Yunpeng Li](#) (EPFL), [Andrew Owens](#) (MIT), [Johannes Gehrke](#) (Cornell) [Dan Huttenlocher](#) (Cornell)

Other Collaborators: [Sameer Agarwal](#) (University of Washington and Google), [Brian Curless](#) (UW), [Yasutaka Furukawa](#) (Google), [Steve Seitz](#) (UW), [Ian Simon](#) (UW), [Rick Szeliski](#) (Microsoft Research)

We gratefully acknowledge the support of the following sponsors:



[National Science Foundation](#)



[Amazon Web Services in Education](#)



[MIT Lincoln Labs](#)

Microsoft

[Microsoft](#)

Ψ

[IU Data to Insight Center](#)

<http://www.cs.cornell.edu/projects/bigsfm/>

Photosynth



Географическая привязка по миру



Yunpeng Li, Noah Snavely, Dan Huttenlocher, Pascal Fua. Worldwide Pose Estimation. ECCV 2012



Scene Chronology

Kevin Matzen and Noah Snavely
Cornell University



ECCV 2014

Резюме



- Разреженная трёхмерная реконструкция – один из наиболее проработанных разделов компьютерного зрения
- Умеем:
 - Калибровать камеру по шаблону
 - Определять взаимное расположение камер
 - Триангулировать точки
 - Определять положение камеры относительно объекта по 2D-3D соответствиям
 - Сопоставлять изображения на больших коллекциях
- Направление развития:
 - Масштабирование и ускорение задачи
 - Учёт временной компоненты, обработка динамических сцен