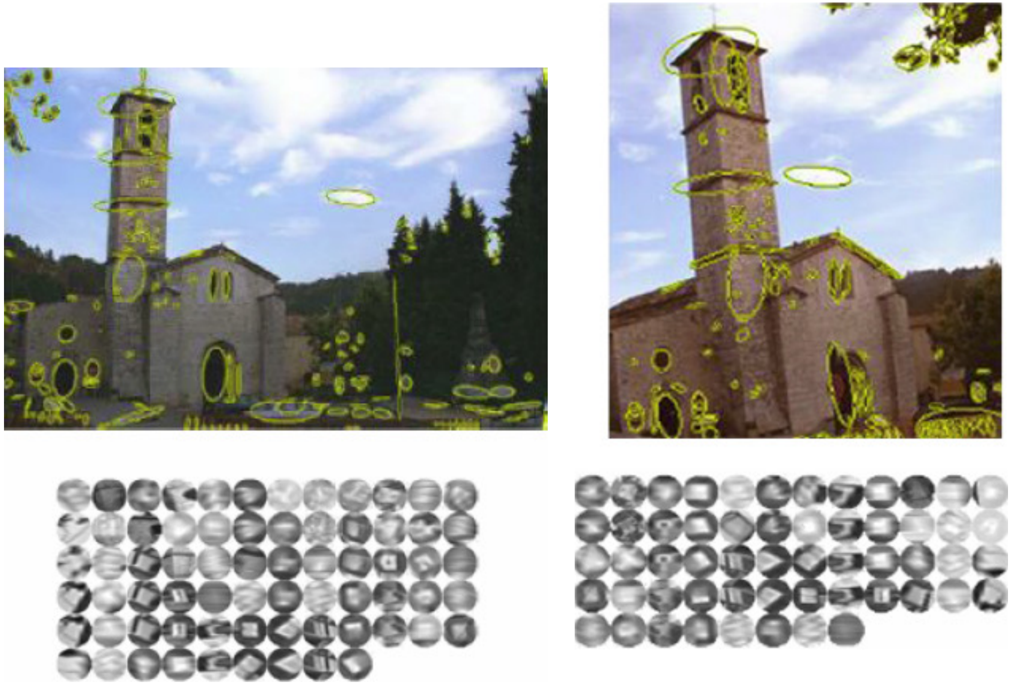
Сопоставление точек

Техническое зрение

Сопоставление точек

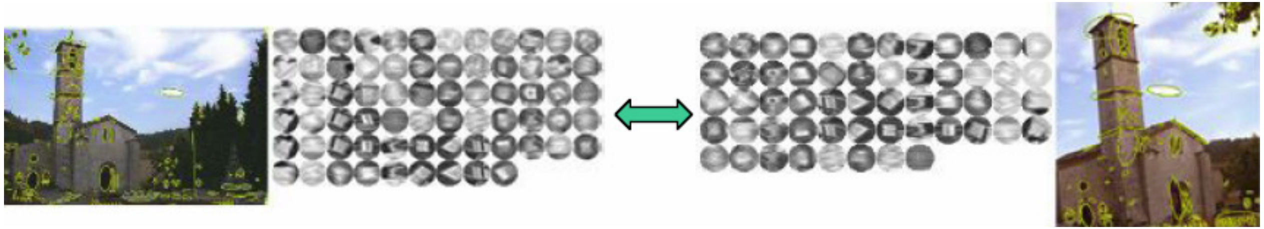
Сопоставление точек

- Имеем набор выделенных особых точек и их дескрипторы.
- Как сопоставить одинаковые точки на разных изображениях?



Сопоставление точек

- Для сопоставления точек необходимо сгенерировать пары-кандидаты: для каждого патча в одном изображении находим несколько наиболее похожих по выбранной метрики патчей на другом изображении.
- Способы выбора пар-кандидатов точек:
  1. Полный перебор:
     - Для каждой особенности вычисляем расстояния до всех особенностей второго изображения и берем лучшую.
  2. Ускоренные приближенные меры:
     - Иерархические структуры (kd-trees, vocabulary trees).
     - Хэширование.



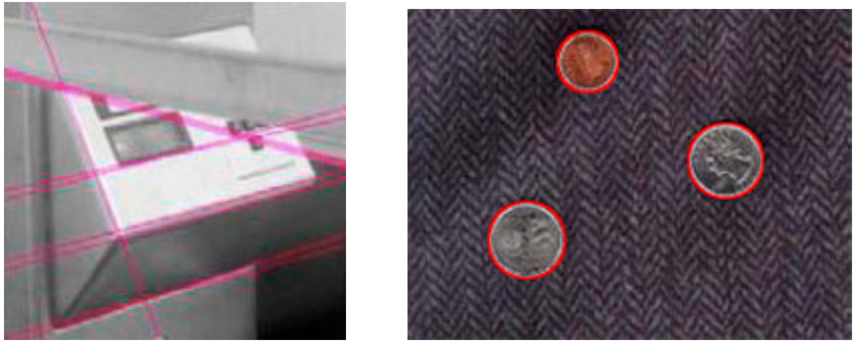
Геометрические модели структур

- Локальные особенности принадлежат определенным геометрическим структурам:
  1. углы окон лежат на прямых;
  2. края окон лежат на прямых.
- На основе этих структур можно вычислить их геометрические модели.

Слева – выделенные особенности,
справа – геометрические модели

Описание объектов

- Простые модели – параметрические кривые:



Слева – линии, справа – окружности

- Сложная модель – автомобиль:

Описание объектов

- **Параметрические кривые:** $F(x, a) = 0$ – параметрическая модель,

где a – параметры модели,

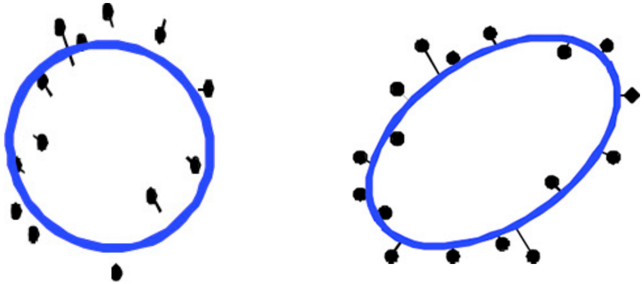
x – вектор, соответствующий некоторым точкам в пространстве,

$X = \{x_i\}$ – множество векторов, соответствующих точкам в пространстве.

- **Прямая линия:** $F(x, a) = a_1 x_1 + a_2 x_2 + a_3 = 0.$
- **Окружность:** $F(x, a) = (x_1 - a_1)^2 + (x_2 - a_2)^2 - a_3 = 0.$
- **Коника:** $F(x, a) = a_1 x_1^2 + a_2 x_1 x_2 + a_3 x_2^2 + a_4 x_1 + a_5 x_2 + a_6 = 0.$
    - Коническое сечение плоскости с круговым конусом или кривая второго порядка.

Описание объектов

- **Параметрические кривые:** в случае двумерной плоскости $x_1 = x, x_2 = y$.



- Задачи по оценке параметров модели на изображении:
  - Даны точки, удовлетворяющие модели. Необходимо вычислить параметры модели.
  - Дана модель. Определить, какие точки ей удовлетворяют, какие нет.
  - Даны точки, часть из них удовлетворяет модели (inliers), часть не удовлетворяет (outliers). Вычислить параметры модели и разделить данные на inliers и outliers.
  - Model fitting – «подгонка модели».

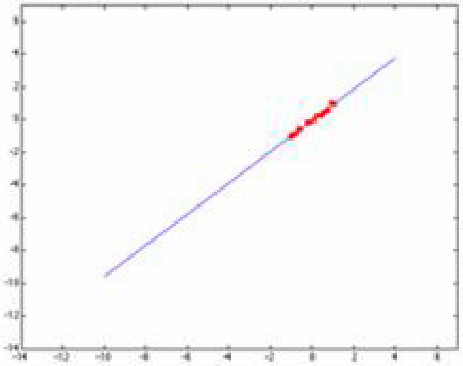
Методы описания объектов

ИТМО

- Прямое линейное преобразование (DLT)
- M-оценки (M-estimators)
- Метод взвешенных наименьших квадратов
- Метод итеративно перевзвешиваемых наименьших квадратов
- RANSAC
- Метод медианных квадратов (LMS)
- M-SAC (M-estimator Sample Consensus)

Прямое линейное преобразование (DLT)

- **Direct Linear Transform (DLT)**
- **Задача:** на двумерном изображении задан набор точек с координатами $(x_i, y_i), i = 1, n$ Необходимо найти прямую, наилучшим образом аппроксимирующую их.
- **Решение:**
  - методом наименьших квадратов рассчитать прямую как минимум квадратов расстояний от точек до прямой;
  - вероятностная формулировка: поиск максимума правдоподобия по расстоянию: $\hat{l} = \arg\max_{l} P[\{(x_i, y_i)\}|l]$.



Прямое линейное преобразование (DLT)

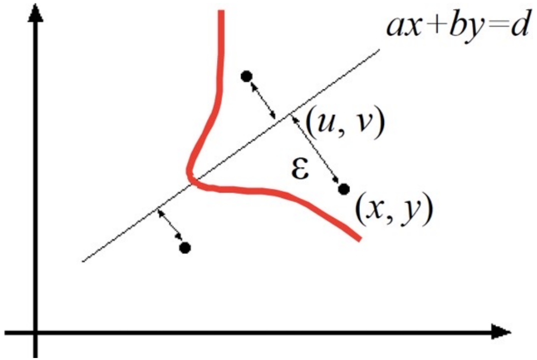
- Модель прямой с зашумленными Гауссовским шумом в перпендикулярном направлении к линии точками:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} + \varepsilon \begin{pmatrix} a \\ b \end{pmatrix},$$

где вектор $[u \quad v]^{\mathrm{T}}$ – точка на линии,

ε – нормально распределенный гауссовский шум с нулевым мат. ожиданием и стандартным отклонением σ,

вектор $[a \quad b]^{\mathrm{T}}$ – нормаль.



Прямое линейное преобразование (DLT)

- Необходимо найти точки с максимумом вероятности нахождения на линии (максимум правдоподобия).
- Правдоподобие точек с параметрами a, b, d вычисляется следующим образом:

$$P(x_1, \dots, x_n | a, b, d) = \prod_{i=1}^{n} P(x_i | a, b, d) \Rightarrow \prod_{i=1}^{n} \mathrm{e}^{-\frac{(ax_i + by_i - d)^2}{2\sigma^2}}$$

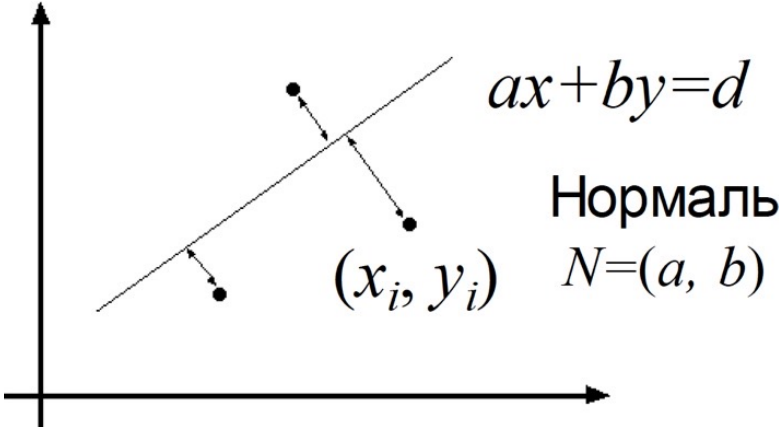
- При использовании натурального логарифма:

$$L(x_1, \dots, x_n | a, b, d) = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (ax_i + by_i - d)^2$$

Прямое линейное преобразование (DLT)

- Поскольку расстояние от точки (x_i, y_i) до линии по нормали равно $|ax + by - d|$, необходимо найти такие параметры a, b, d которые минимизируют функцию E:

$$E = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$



Прямое линейное преобразование (DLT)

- Продифференцируем функцию E по d и приравняем нулю:

$$\frac{\partial E}{\partial d} = \sum_{i=1}^{n} -2(ax_i + by_i - d) = 0,$$

и выразим d:

$$d = \frac{a}{n}\sum_{i=1}^{n} x_i + \frac{b}{n}\sum_{i=1}^{n} x_i = a\bar{x} + b\bar{y}$$

- Подставим полученное выражение в функцию E:

$$E = \sum_{i=1}^{n}\big(a(x_i - \bar{x}) + b(y_i - \bar{y})\big)^2 =$$

$$= \left\| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \dots & \dots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \quad \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = (AN)^{\mathrm{T}}(AN).$$

Прямое линейное преобразование (DLT)

- Продифференцируем $(AN)^T(AN)$ by N:

$$\frac{dE}{dN} = 2\left(A^{\mathrm{T}}A\right)N = 0.$$

- Видно, что решением данного матричного уравнения является собственный вектор $A^{\mathrm{T}}A$, соответствующий *минимальному собственному значению* при условии, что $\|N\|^2 = 1$.
- Выражение $A^{\mathrm{T}}A$ позволяет отыскать *сингулярные числа* матрицы A.

Прямое линейное преобразование (DLT)

- Для упрощения поиска сингулярных чисел рассмотрим сингулярное разложение матриц.
- SVD-процедура (Singular Value Decomposition).
- Можно разложить матрицу $A = UDV^{\mathrm{T}}$,

где U и V – ортогональные матрицы,

D – диагональная матрица, состоящая из сингулярных чисел.

- Справедливы следующие соотношения:

$$A^{\mathrm{T}}A = VDU^{\mathrm{T}}UDV^{\mathrm{T}} = VDDV^{\mathrm{T}} = VD^2V^{\mathrm{T}}$$

Прямое линейное преобразование (DLT)

- Используем SVD-процедуру для вычисления наименьших квадратов.
- Пусть дано уравнение:

$$Ap = 0$$

где норма вектора p: $\|p\| = 1$.

- Для поиска минимального сингулярного числа необходимо минимизировать норму: $\left\|UDV^{\mathrm{T}}p\right\|$.
- С учетом равенства на предыдущем слайде:

$$\left\|UDV^{\mathrm{T}}p\right\| = \left\|DV^{\mathrm{T}}p\right\| \cdot \left\|V^{\mathrm{T}}p\right\| = \|p\|$$

Прямое линейное преобразование (DLT)

- Если $\|V^T p\| = 1$, то необходимо минимизировать:

$$\|DV^T p\|.$$

- Обозначим $y = V^T p$, тогда необходимо минимизировать:

$$\|Dy\|, \text{ если } \|y\| = 1,$$

а в диагональной матрице D столбцы упорядочены по убыванию.

- В таком случае $y = (0, \dots, 0,1)^T$,

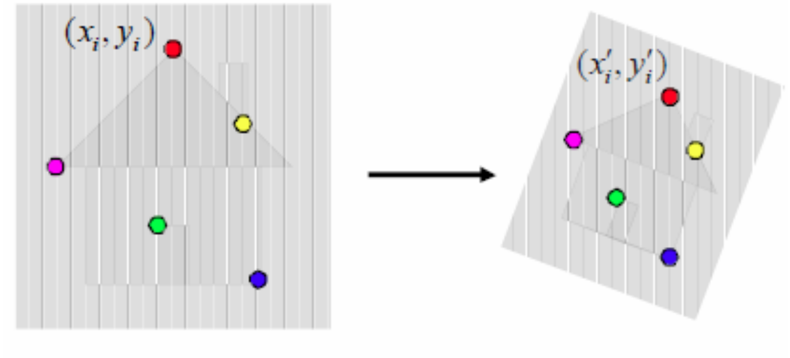
а $p = Vy$ – последний столбец матрицы V.

Прямое линейное преобразование (DLT)

- Используем МНК и SVD-процедуру для построения прямых.
- Пусть задан набор точек (x_i, y_i).
- Для построения линии $ax + by = d$ необходимо:
    1. Вычислить средние значения $(\bar{x}, \bar{y})$;
    2. Сформировать матрицу A содержащую отклонения от средней точки;
    3. Выполнить SVD-процедуру $A = UDV^{\mathrm{T}}$;
    4. Вычислить параметры a и b из последнего столбца матрицы V;
    5. Найти $d = a\bar{x} + b\bar{y}$.
- Метод наименьших квадратов для нахождения параметров моделей и называется DLT (Direct Linear Transform).

Геометрические преобразования

- Оценкой параметров модели также являются геометрические преобразования, рассмотренные ранее на предыдущих лекциях.
- Напомним, что есть набор из нескольких точек $X = (x_i, y_i)$ на одном изображении и соответствующий набор таких же точек $X' = (x'_i, y'_i)$ на другом изображении.
- Необходимо решить матричное уравнение вида XT = X' для определения параметров преобразования T.



Геометрические преобразования

- Решение матричного уравнения:
    1. С помощью псевдообратной матрицы X^+:

$$T = X^+ X',$$

где $X^+ = \left(X^{\mathrm{T}} X\right)^{-1} X^{\mathrm{T}}$ – псевдообратная матрица.

    2. С помощью SVD-процедуры:

$$X = UDV^{\mathrm{T}} \rightarrow X^+ = VD^+U^{\mathrm{T}},$$

где D – диагональная матрица,

причем псевдообратная матрица D^+ состоит из обратных элементов $\frac{1}{d_{ii}}$ матрицы D.

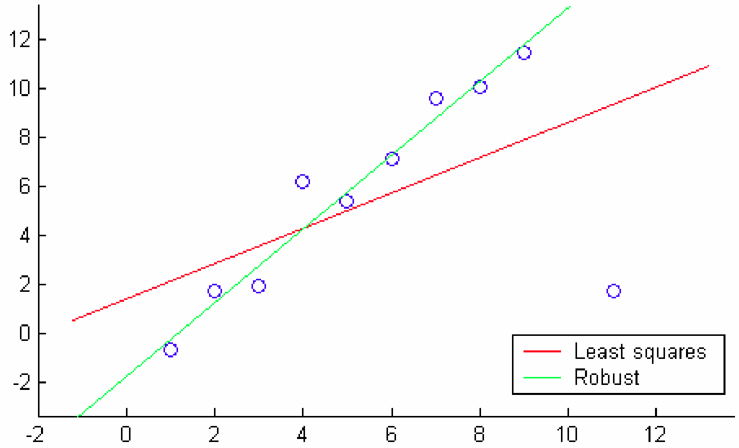
- Из-за этого данное решение более математически привлекательно при использовании в расчетах.

Резюме DLT

- Проблема линейных методов оценки параметров для сложных моделей: оптимизация алгебраической ошибки, не имеющей физического смысла.
- Для каждой модели есть нормальная ошибка, имеющая физический смысл, которую и нужно минимизировать.
    - Для этих целей используется метод «Gold Standard».
- Чаще всего для геометрических моделей оптимальной метрикой является расстояние от точки до кривой, или некой «оптимальной» точки.

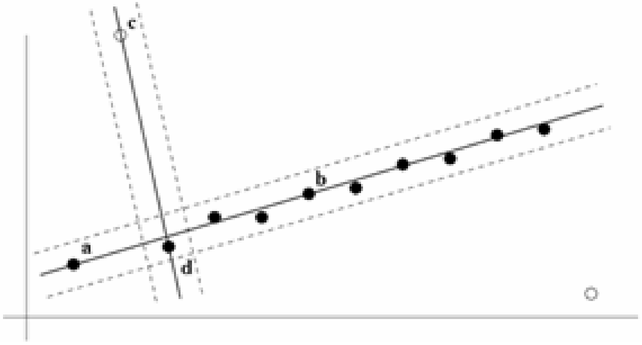
Проблемы DLT

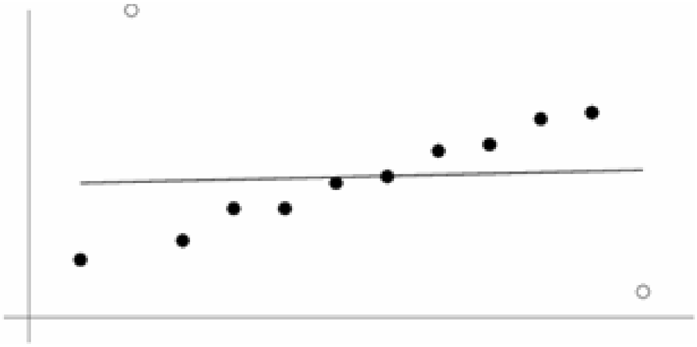
- Зачастую часть полученных точек не порождена моделью (x, a).
- В такой ситуации при оценке методом наименьших квадратов результат может быть сколь угодно далеким от истинного.
- Например, имеем набор пикселей, отобранных по порогу и построим на их основе прямую:



Описание объектов

- Аномалии называются «выбросами» (outliers).
- Точки, удовлетворяющие модели, «не выбросами» (inliers).





М-оценки

- Для уменьшения влияния дальних точек, параметризуем точки в полярных координатах:

$$x\cos\theta + y\sin\theta = R,$$

тогда целевая функция примет вид:

$$(\theta, R) = \arg\min_{(\theta,R)} \sum_i (x_i\cos\theta + y_i\sin\theta - R)^2 \,.$$

- Обозначим $\varepsilon_i = x_i\cos\theta + y_i\sin\theta - R$, и модифицируем целевую функцию:

$$(\theta, R) = \arg\min_{(\theta,R)} \textstyle\sum_i \rho(\varepsilon_i),$$

где в случае $\rho(\varepsilon) = \varepsilon^2$ получим метод наименьших квадратов.

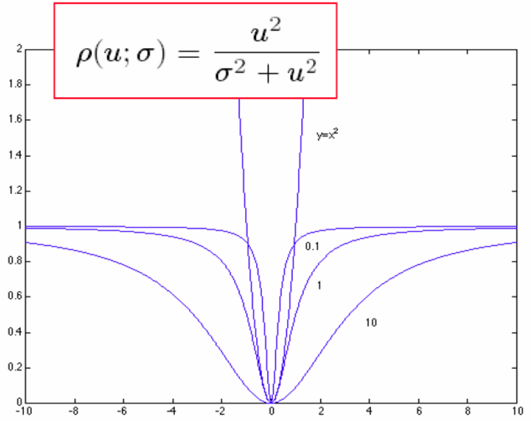
M-оценки

- Обычно минимизируется следующая функция:

$$\sum_i \rho(r_i(x_i, \theta), \sigma),$$

где $r_i(x_i, \theta)$ – невязка i-ой точки при условии параметров модели θ,

ρ – робастная функция с масштабом σ:



Робастная функция ρ ведёт себя как квадрат расстояния при малых значениях u и выравнивается с увеличением значения u.

М-оценки

- В качестве наиболее часто используемых вариантов робастной функции ρ используются следующие функции:
  1. Функция Тьюки:

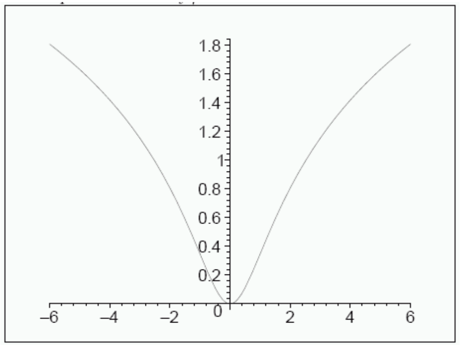
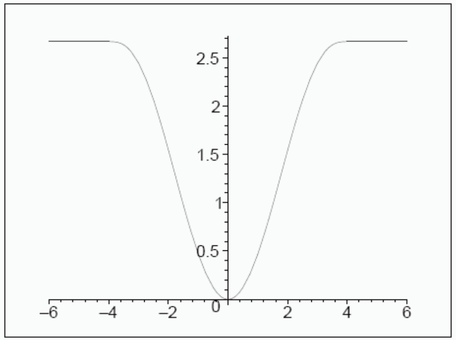
$$\rho(\varepsilon) = \begin{cases} \frac{K^2}{6}\left(1 - \left(1 - \left(\frac{\varepsilon}{K}\right)^2\right)^3\right), \text{если } |\varepsilon| \le K \\ \frac{K^2}{6}, \text{если } |\varepsilon| > K \end{cases}$$

  2. Функция Коши:

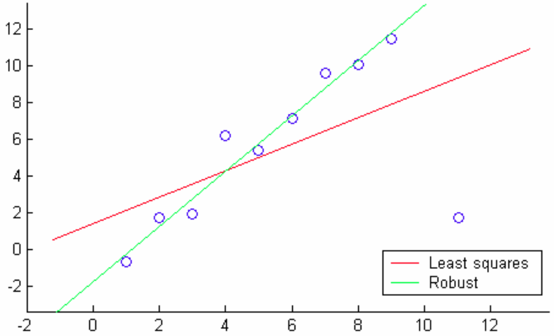
$$\rho(\varepsilon) = \frac{c^2}{2}\log\left(1 + \left(\frac{\varepsilon}{c}\right)^2\right),$$

где K и c – настроечные константы.

M-оценки

Слева функция Тьюки при $K = 4{,}685$, справа функция Коши при $c = 2{,}385$



Сравнение метода наименьших квадратов и робастной модификации оценки

М-оценки

- Как найти минимум целевой функции? При определенных робастных функциях сделать это крайне затруднительно.
- Методы:
  1. Методы нелинейной оптимизации;
  2. Метод взвешиваемых наименьших квадратов;
  3. Метод итеративно перевзвешиваемых наименьших квадратов.

Метод взвешиваемых наименьших квадратов

- На примере поиска прямых:

$$(a, b, d) = \arg \min_{(a,b):\, a^2+b^2=1} \sum_i w_i (ax_i + by_i + d)^2,$$

$$\sum_i w_i = 1,$$

где w_i – вес каждой точки.

- Построим ковариационную матрицу точек, представляющую собой квадратную симметрическую неотрицательно определенную матрицу:
  - На главной диагонали располагаются дисперсии координат точек.
  - Внедиагональные элементы – ковариации между точками.

$$Cov = \begin{bmatrix} \sum_i w_i (x_i - \bar{x})^2 & \sum_i w_i (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_i w_i (x_i - \bar{x})(y_i - \bar{y}) & \sum_i w_i (y_i - \bar{y})^2 \end{bmatrix}.$$

Метод взвешиваемых наименьших квадратов

- **Максимальный** собственный вектор матрицы Cov задает направление прямой $\vec{l}$,
- **Минимальный** – направление нормали $\overrightarrow{(a, b)}$.
- Прямая проходит через **среднюю точку** $(\bar{x}, \bar{y})$,

где $\bar{x} = \frac{\sum_i w_i x_i}{\sum_i w_i}$,

$$\bar{y} = \frac{\sum_i w_i y_i}{\sum_i w_i},$$

$$d = -(a\bar{x} + b\bar{y}).$$

Метод итеративно перевзвешиваемых наименьших квадратов

1. Получить начальное приближение модели методом наименьших квадратов: $\Theta^{(0)} = \left(\rho^{(0)}, \theta^{(0)}\right)$.
2. Установить номер итерации $t = 1$.
3. Для $\Theta^{(t-1)}$ рассчитать текущую оценку шума:

$$\sigma^{(t)} = 1{,}4826 \operatorname{median}_i \left| r_i^{(t)}\left(x_i, \Theta^{(t-1)}\right) \right|,$$

которая является несмещенной (для нормального распределения) робастной оценкой средней ошибки.

Метод итеративно перевзвешиваемых наименьших квадратов

4. Рассчитать веса точек $w_i^{(t)}$ с учетом функции ρ,
   a. в общем случае:

$$w_i = \frac{\rho'\left(\frac{\varepsilon_i}{\sigma}\right)}{\frac{\varepsilon_i}{\sigma}},$$

   b. в случае функции Тьюки:

$$w\left(\frac{\varepsilon}{\sigma}\right) = \begin{cases} \left(1 - \left(\frac{\varepsilon}{\sigma \cdot K}\right)^2\right), \text{если } \left|\frac{\varepsilon}{\sigma}\right| \le K, \\ 0, \text{если } \left|\frac{\varepsilon}{\sigma}\right| > K, \end{cases}$$

   c. в случае функции Коши:

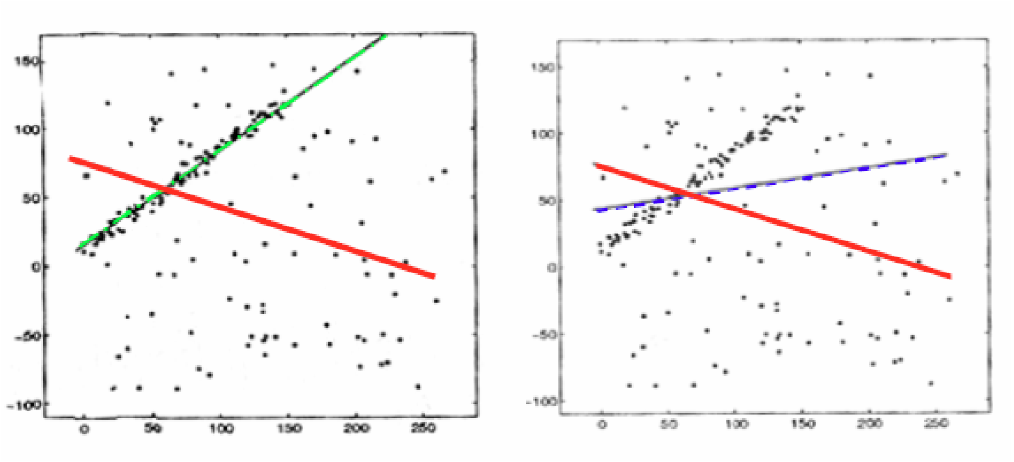
$$w\left(\frac{\varepsilon}{\sigma}\right) = \frac{1}{1 + \left(\frac{\varepsilon}{c \cdot \sigma}\right)^2}.$$

Метод итеративно перевзвешиваемых наименьших квадратов

5. Используя взвешенные наименьшие квадраты получить $\Theta^{(t)}$.
6. Если $\left\|\Theta^{(t)} - \Theta^{(t-1)}\right\| > \varepsilon^*$, то перейти на шаг 3,

где ε^* — максимальное желаемое отклонение.

Метод итеративно перевзвешиваемых наименьших квадратов



Влияние настроечной константы c на построение линии:
красная линия – первый шаг,
зеленая – $c = 1{,}5$,
синяя – $c = 3{,}5$.

Недостатки М-оценок

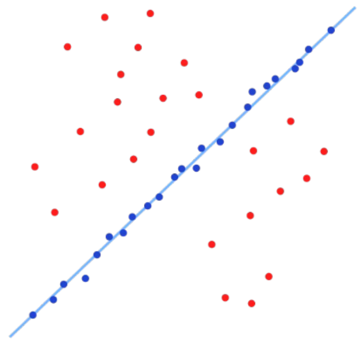
1. Необходимость хорошего первого приближения;
2. Необходимо правильно рассчитывать баланс весов, чтобы выполнялась работа алгоритма с достаточной точностью.

RANdom SAmple Consensus – RANSAC

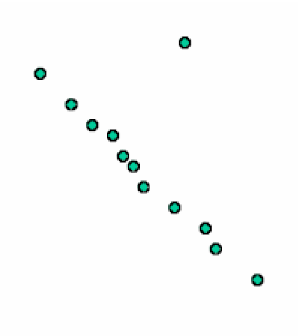
- Метод оценки параметров модели на основе случайных выборок.
- Идея: проведение оценки не по всем данным, а лишь по небольшой выборке, не содержащей выбросов.
- Поскольку заранее неизвестно какие точки являются выбросами, а какие нет, то можно построить сразу много выборок случайным образом.
- Затем по каждой из выборок строим гипотезу. После этого выбираем такую гипотезу из всех, которая наилучшим образом согласуется со всеми данными.

RANSAC

- Основная проблема: число таких выборок огромно, поэтому необходимо строить гипотезы по выборке минимального размера.
- Например, при вписывание прямой в множество точек на плоскости, данный метод берет за основу только две точки необходимые для построения прямой и с их помощью строит модель.
- После этого проверяется, какое количество точек соответствует модели, используя функцию оценки с заданным порогом.

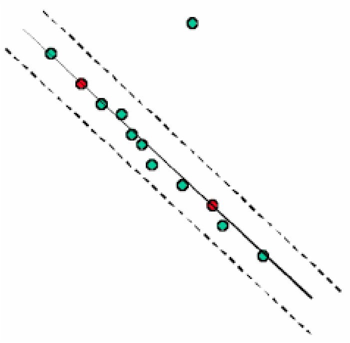


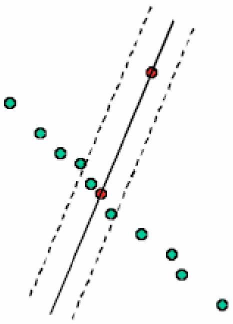
Пример RANSAC



Набор данных, в который необходимо вписать прямую

Пример RANSAC

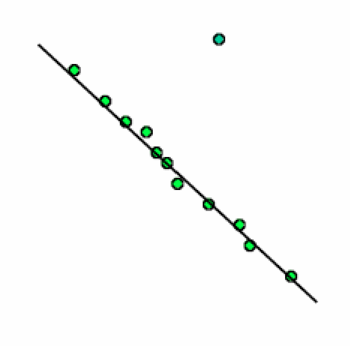




Две минимальные выборки (по две точки) с отсечением по порогу вдоль предложенной прямой

На левом изображении в область попало 11 точек, на правом – 4

Пример RANSAC



Левая выборка более адекватно описывает прямую (получила больше «голосов»), соответственно является верным решением

Базовый алгоритм RANSAC

- Цикл из N-итераций:
    1. Построить выборку $S \subset X$ $(x_i \in X)$. Как правило, минимально возможного размера для оценки параметров.
    2. Выдвинуть гипотезу Θ по выборке S.
    3. Оценить степень согласия гипотезы Θ и набора исходных данных X. Каждая точка помечается «выбросом» или «не выбросом».
    4. После проверки всех точек, проверяется, является ли гипотеза лучшей на данный момент, и если является, то она замещает предыдущую лучшую гипотезу.
- В конце работы цикла оставляется последняя лучшая гипотеза, из которой можно определить параметры модели, а также точки, помеченные как «выбросы» и «не выбросы».

RANSAC

- Для получения модели, построенной без выбросов с заданной вероятностью p, количество итераций N цикла можно рассчитать, если возможно указать заданную долю «выбросов» e.
- Количество выборок N выбирается так, чтобы вероятность выбора хотя бы одной выборки без выбросов была бы не ниже заданной (например, 0,99). Таким образом:

$$(1-(1-e)^S)^N = 1-p,$$

$$N = \frac{\log(1-p)}{\log(1-(1-e)^S)},$$

где N – количество выборок (число итераций),

p – вероятность получить хорошую выборку за N итераций,

S – количество элементов (точек) в выборке,

e – доля «выбросов».

RANSAC

- Для оценки степени согласия гипотез рассмотрим функции оценки гипотез:

$$R(\Theta) = \sum_i p(\varepsilon_i(\Theta)^2), p(\varepsilon_i(\Theta)^2) = \begin{cases} 1, \text{если } |\varepsilon_i| \le T \\ 0, \text{если} |\varepsilon_i| > T \end{cases}, i = \overline{1, n},$$

где $\varepsilon_i(\Theta)$ – невязка i-ой точки и оцениваемой гипотезы;

p – вероятность (1 – «не выброс», 0 – «выброс»);

T – порог, выбираемый из соображений, что величина вероятности «не выброса» (inlier) была $p \approx 0{,}95$.

- Как правило, используется Гауссова модель шума с нулевым математическим ожиданием такая, что $T^2 = 3{,}84\sigma^2$.

Метод медианных квадратов

- LMS (Least Median Squares, метод медианных квадратов)
- Используется функция оценки гипотез:

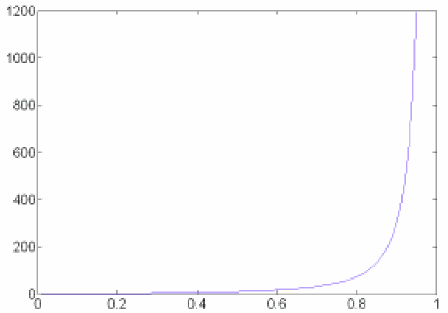
$$R(\Theta) = \mathrm{median}(\varepsilon_i(\Theta)^2), i = \overline{1, n},$$

где $\varepsilon_i(\Theta)$ – невязка i-ой точки и оцениваемой гипотезы.

RANSAC

- Количество выбросов быстро растет с ростом размера выборки и доли «выбросов».

	proportion of outliers e						
s	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



Зависимость количества выборок от размера выборки и доли «выбросов»

- Как минимизировать долю выбросов, которая заранее неизвестна?

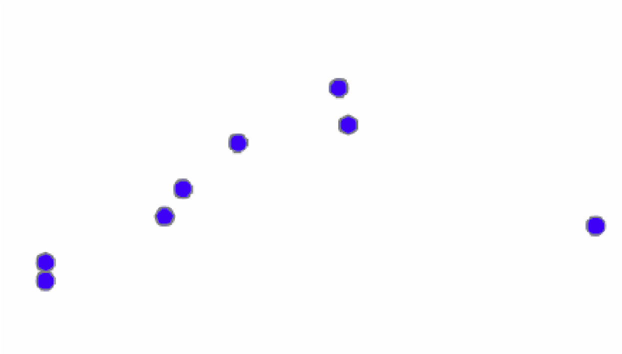
Адаптивное завершение алгоритма

- Можно начать алгоритм с грубой оценки, например, 50%, и затем последовательно уточнять число выборок.
- Адаптивное завершение алгоритма RANSAC:

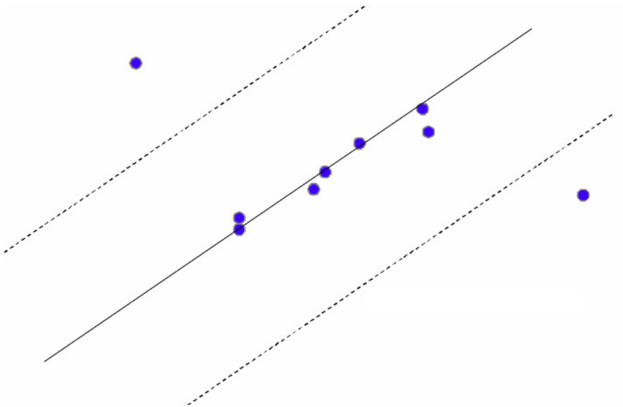
```
N=999999, sample_count = 0, p = 0.99;
while(N > sample_count) {
    //Базовый алгоритм RANSAC:
    //Построение выборки S (количество точек в выборке),
    //гипотезы, оценка «не выбросов» в выборке inliers
    e = 1 - (inliers / S);
    N = log(1 - p) / log(1 - (1 - e)^S );
    sample_count++;
}
```

Проблема выбора порога

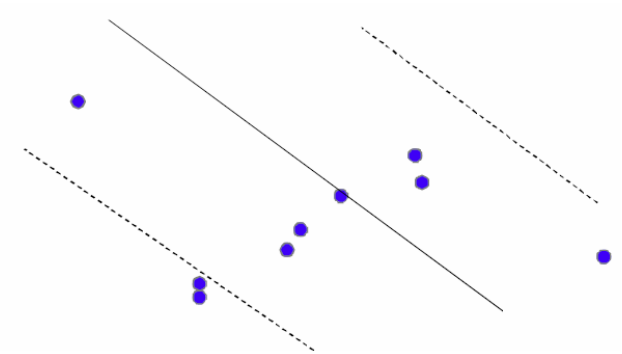
- Одним из недостатков метода RANSAC является неопределенность выбора порога.
- Как большой, так и маленький пороги приводят к неверным результатам.
- Пусть задан набор точек:



Проблема выбора порога

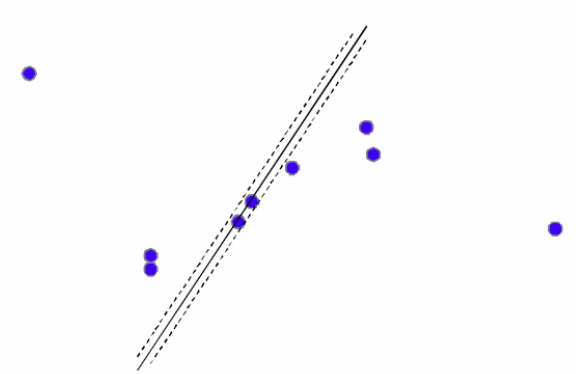


Большой порог:
верное решение

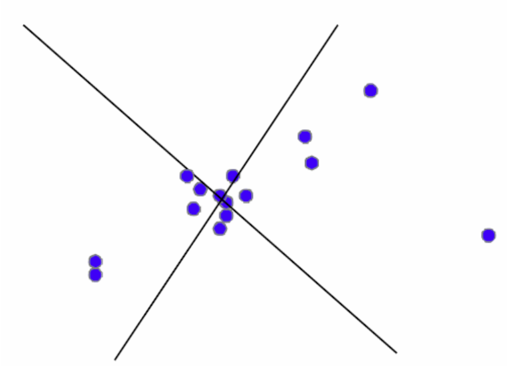


Большой порог: неверное решение,
эквивалентное верному

Проблема выбора порога



Маленький порог:
неверное решение



Проблема LMS: медиана ошибки
одинакова для обоих решений

Алгоритм M-SAC

- Для оценки степени согласия гипотез рассмотрим **функцию оценки гипотез**:

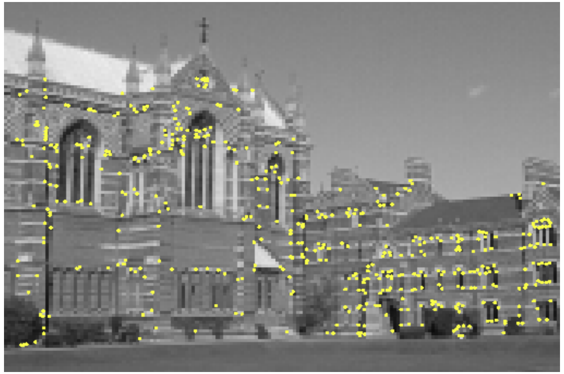
$$R(\Theta) = \sum_i p(\varepsilon_i(\Theta)^2), p(\varepsilon_i(\Theta)^2) = \begin{cases} \varepsilon_i^2, \text{если } \varepsilon_i^2 \le T^2 \\ T^2, \text{если } \varepsilon_i^2 > T^2 \end{cases}, i = \overline{1, n},$$

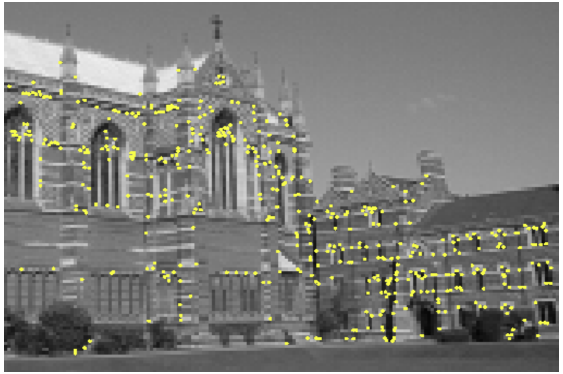
которая аналогична функции RANSAC за исключением модификации функции вероятности.

- Данный метод дает более точную оценку без увеличения вычислительной сложности и гарантирует верное решение.

Задача сопоставления точек

- Пример использования: сопоставление одинаковых характерных точек.



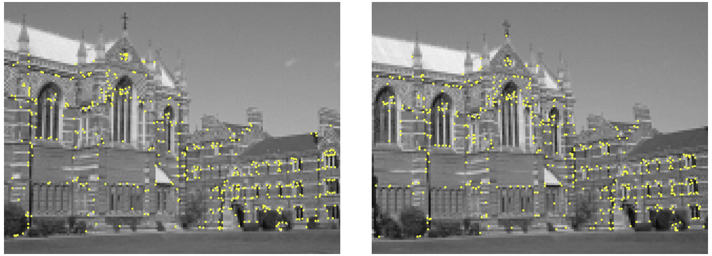


- При сопоставлении особых точек по дескрипторам будет определено довольно много ложных пар.

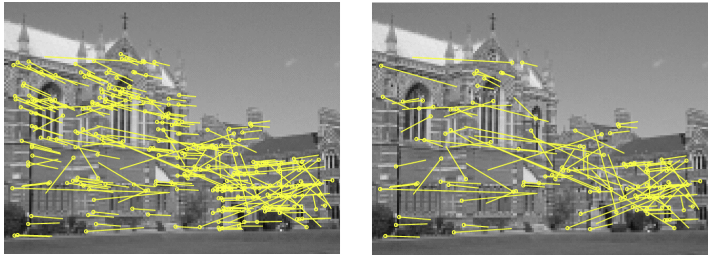
Алгоритм сопоставления

1. На изображениях Im и Im' заданы пары точек $\{x, x'\}$.
2. Вычисление модели преобразования T по ключевым точкам между изображениями Im и Im'.
   - Использование схемы RANSAC для построения модели T в наборах точек.
3. Фильтрация выбросов в $\{x, x'\}$.
4. Уточнение модели по оставшимся точкам.
   - Либо итеративным методом наименьших квадратов;
   - Либо нелинейной минимизацией.

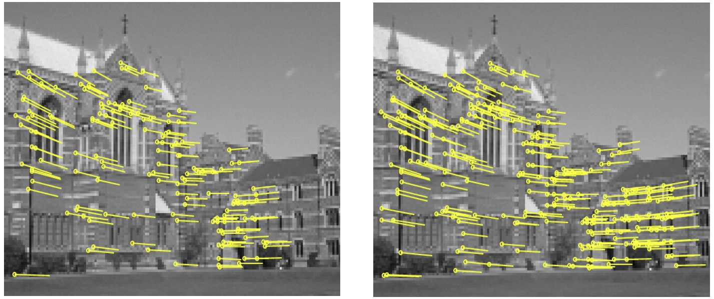
Пример сопоставления точек



На двух изображениях найдено
по 500 характеристических точек



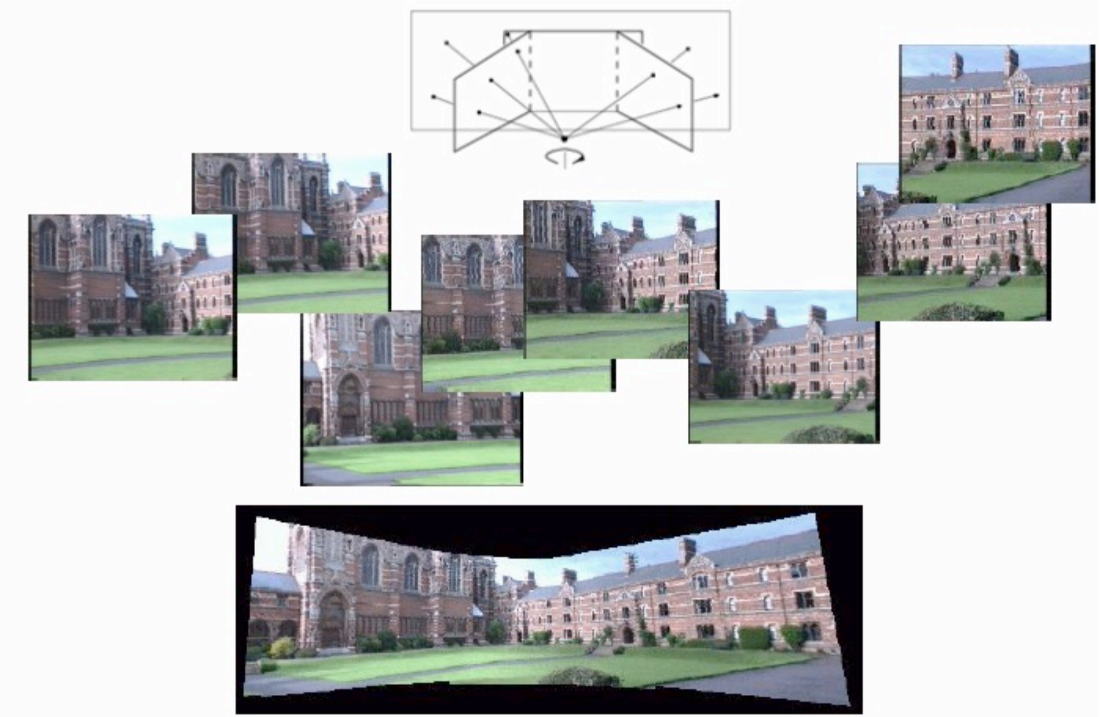
Из них 117 выбросов и 268 соответствий



После фильтрации было отобрано
151 хорошее соответствие

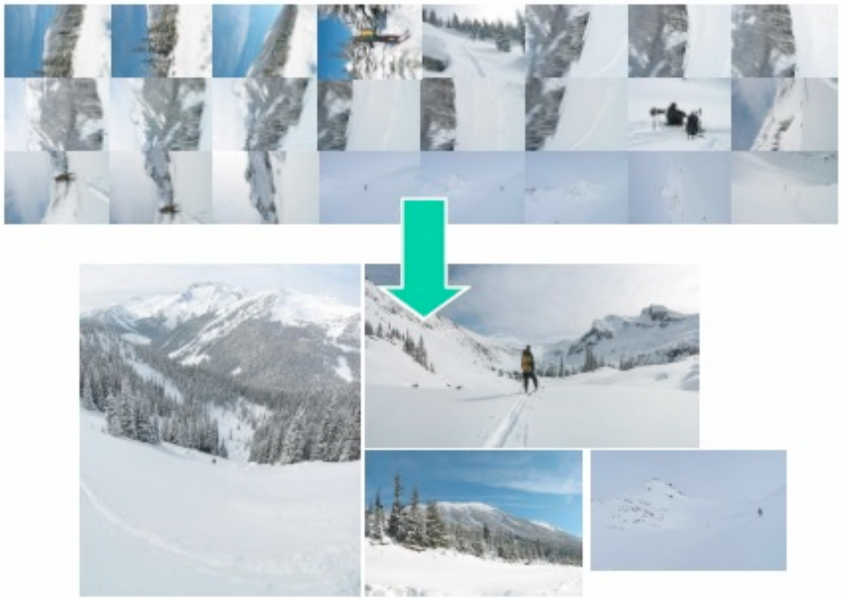
Пример: построение панорамы

- Построение панорамы из упорядоченного набора фотографий.



Пример: построение панорамы

- Построение панорамы из неупорядоченного набора фотографий.
    - Необходимо определить, какие их них относятся к одному изображению, а какие к другому.



Достоинства RANSAC

- Простой и общий метод, применимый для множества задач;
- Хорошо работает на практике;
- Способен дать надёжную оценку параметров модели, то есть оценить параметры модели с высокой точностью, даже если в исходном наборе данных присутствует значительное количество выбросов.

Недостатки RANSAC

- Много настраиваемых параметров;
- Не всегда удается хорошо оценить параметры по минимальной выборке;
- Иногда требуется слишком много итераций;
- Не срабатывает при очень высокой доле выбросов;
- Часто есть лучший способ, нежели равновероятно выбирать точки;
- Отсутствие верхней границы времени, необходимого для вычисления параметров модели;
- Методом RANSAC можно определить только одну модель для определённого набора данных. Как и для любого подхода, предназначенного для одной модели, существует следующая проблема: когда в исходных данных присутствуют две (или более) модели, RANSAC может не найти ни одну.

Вопросы?

ITsMOre than a UNIVERSITY

s.shavetov@itmo.ru