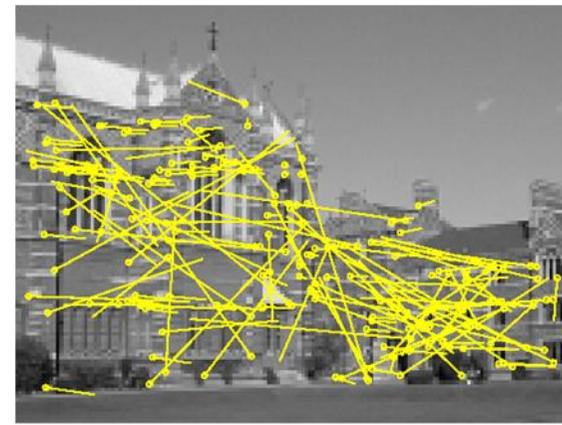
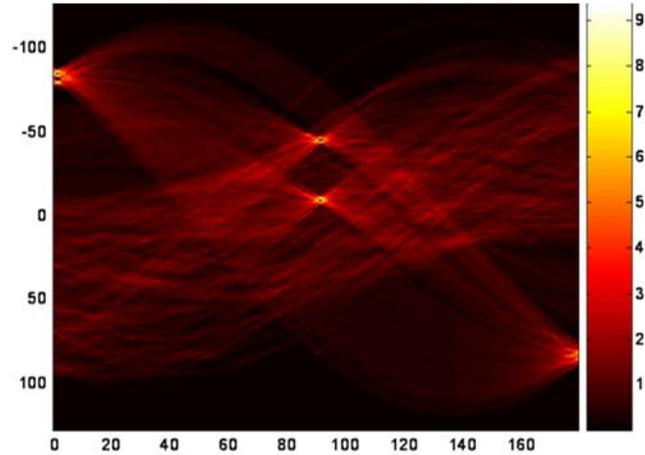
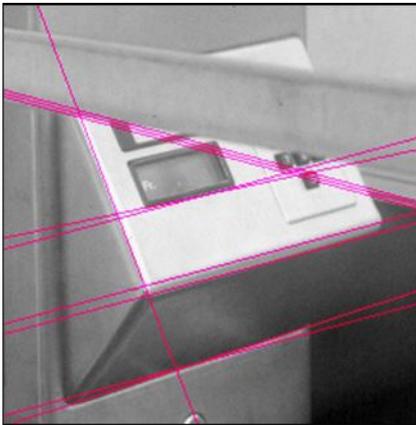
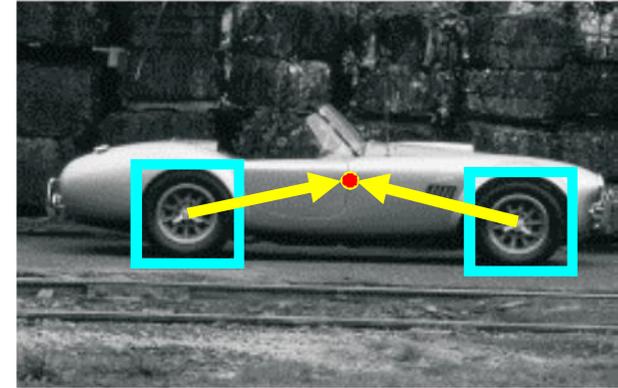
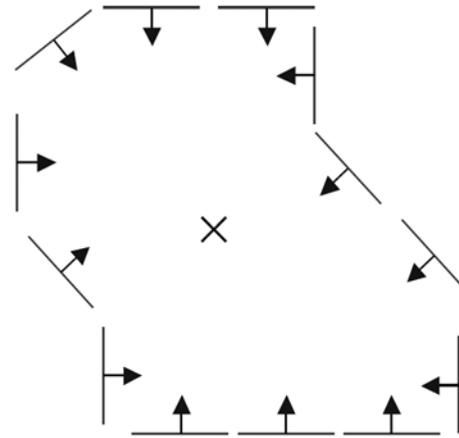
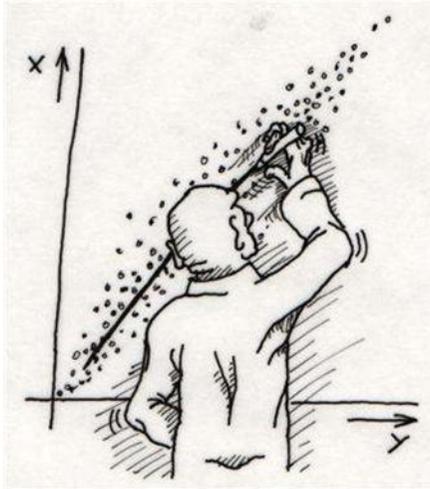




Оценка параметров моделей





Общая информация

Microsoft
Research

Этот курс
подготовлен и
читается при
поддержке Microsoft
Research

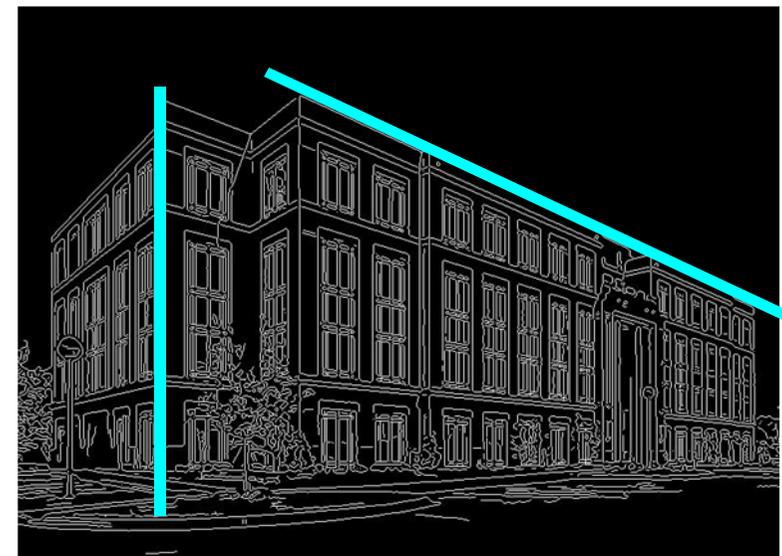
Microsoft
Research

- Страница курса
<http://courses.graphicon.ru/main/vision>



Зачем нужна оценка параметров?

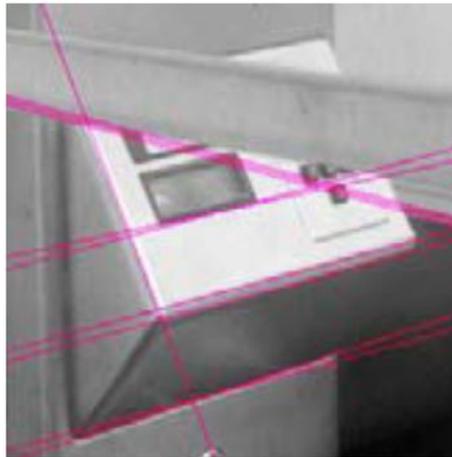
- Умеем находить особенности на изображении - края и особые точки
- Хотим описать множество признаков на более высоком уровне, компактнее
- Сгруппируем точки согласно простым геометрическим моделям





Оценка параметров

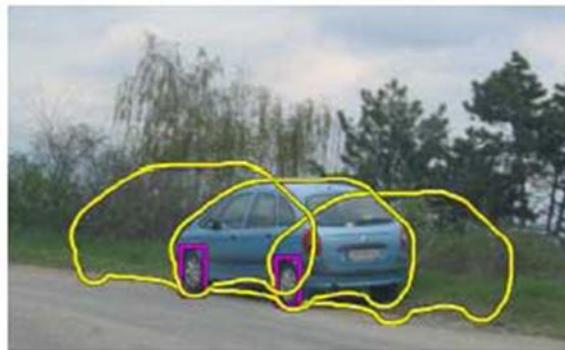
- Выберем параметрическую модель, описывающую геометрическое распределение признаков



Простая модель: линия



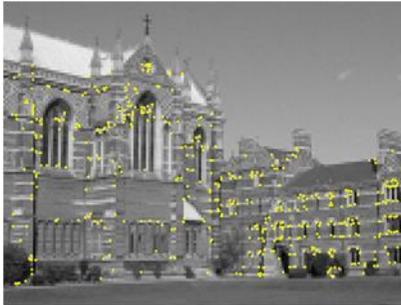
Простая модель: окружность



Сложная модель: машина

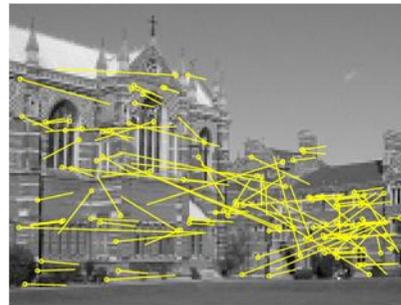
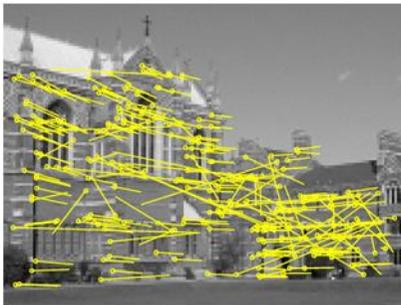


Сопоставление изображений

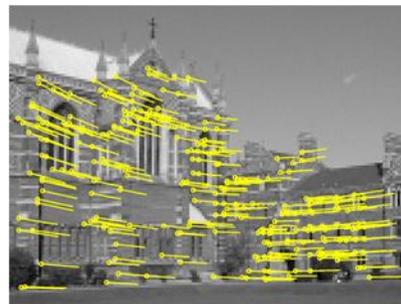
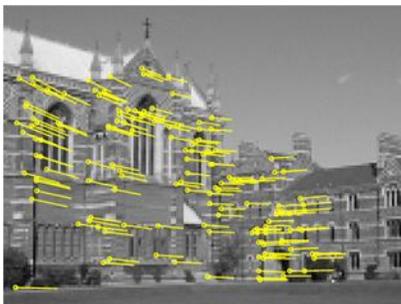


500 особенностей

Лучше картинку про
оценку параметров
преобразования



Соответствия (268)



Выбросы (117)

Хороших соответствий
(151) – благодаря
оценке модели
преобразования!



Оценка параметров

- Выбор модели, описывающей набор признаков
- Критерий принадлежности модели не локальный
 - Только по точке невозможно сказать, удовлетворяет она модели или нет
- Три основных вопроса:
 - Какая модель лучше всего описывает набор признаков?
 - Какому экземпляру (instance) модели принадлежит признак?
 - Сколько экземпляров модели на изображении?
- Вычислительная сложность
 - Невозможно перебрать все возможные экземпляры и параметры моделей



Оценка параметров

Пример: поиск линий



- **Шум** в выделенных признаках
- **Лишние данные:** шум (выбросы), множество линий
- **Пропавшие данные:** перекрытия



Оценка параметров: задачи

- Если мы знаем, какие точки лежат на линии, как нам найти оптимальные параметры линии?
 - Метод наименьших квадратов
- Как быть с выбросами?
 - Робастные оценки, RANSAC
- Как быть, если линий много?
 - Схемы голосования: RANSAC, преобразование Хафа
- Как быть, если мы не уверены, что линия – лучшая модель?
 - Выбор оптимальной модели (Model selection)



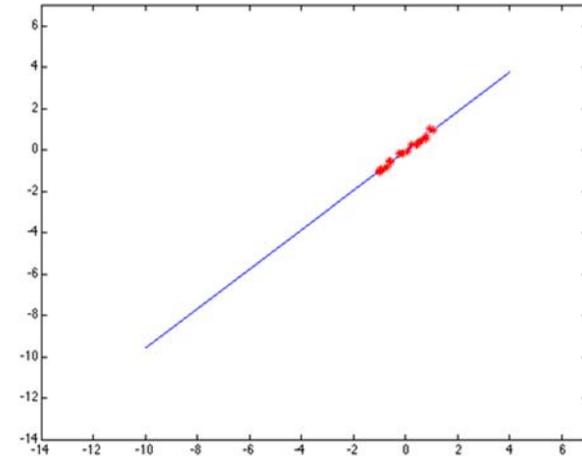
Прямые линии

Дано:

- Набор точек (x_i, y_i) , $i=1..N$

Найти

- Прямую, наилучшим образом их аппроксимирующую

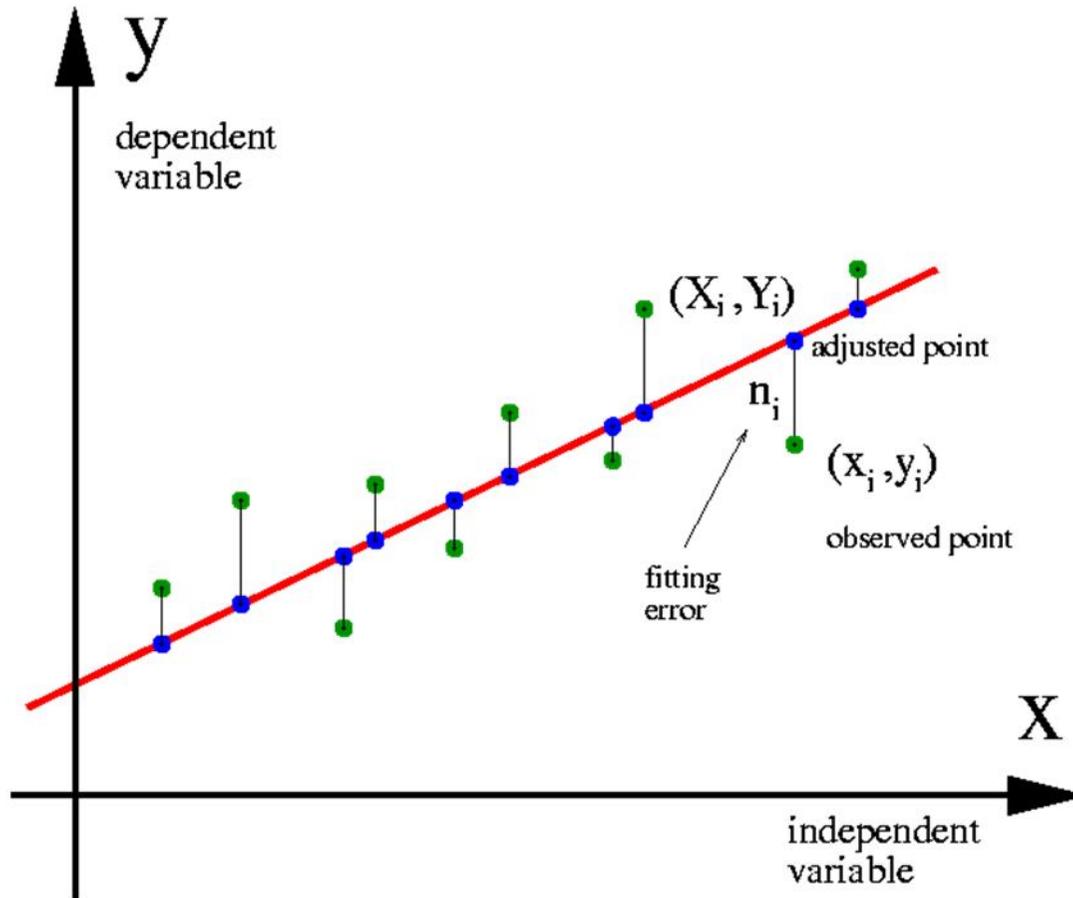


- Здравый смысл:
 - Минимум квадратов расстояний от точек до прямой
- Вероятностная формулировка:
 - Максимум правдоподобия

$$\hat{l} = \arg \max_l P [\{(x_i, y_i)\} | l]$$



Метод наименьших квадратов



Максимум правдоподобия

Нет ошибок в x

Есть ошибки в y

Ошибки:

- независимы
- одинаково распределены
- распределены нормально

$$y = ax + b;$$

$$\forall i \rightarrow x_i = X_i, y_i = Y_i + \epsilon;$$

$$\epsilon \sim N(0, \sigma^2)$$



Метод максимального правдоподобия

Правдоподобие:

$$\arg \max_{(a,b)} P [\{(x_i, y_i)\} | (a, b)]$$

Независимые, нормально
распределенные ошибки:

$$\arg \max_{(a,b)} \prod_i \exp \left[-\frac{(y_i - ax_i - b)^2}{2\sigma_i^2} \right]$$

\log – монотонная функция:

$$\arg \min_{(a,b)} \sum_i \frac{(y_i - ax_i - b)^2}{2\sigma_i^2}$$

Одинаково распределённые
ошибки:

$$\arg \min_{(a,b)} \sum_i (y_i - ax_i - b)^2$$

Получаем метод наименьших квадратов



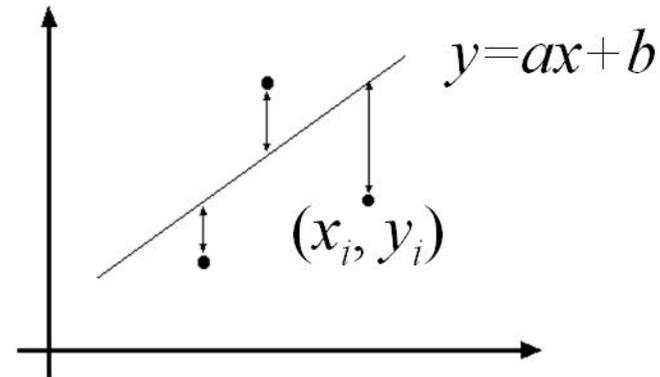
Метод наименьших квадратов

Данные : $(x_1, y_1), \dots, (x_n, y_n)$

Уравнение: $y_i = ax_i + b$

Найти (a, b) минимизирующие

$$E = \sum_{i=1}^n (y_i - ax_i - b)^2$$



This image cannot currently be displayed.

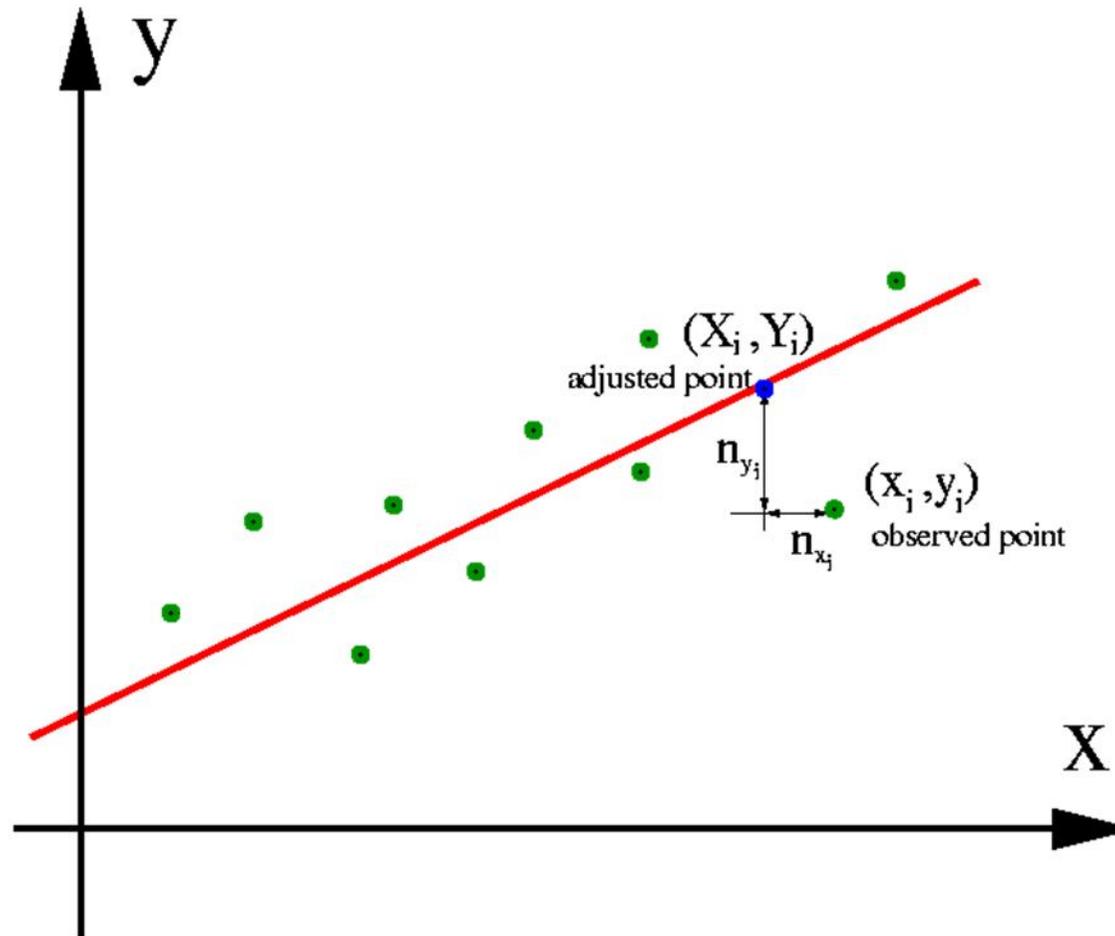
$$\frac{dE}{dB} = 2X^T XB - 2X^T Y = 0$$

This image cannot currently be displayed.

Нормальные уравнения: решение $XB=Y$



Более сложный случай



(ошибки как в x так и в y)



Модель

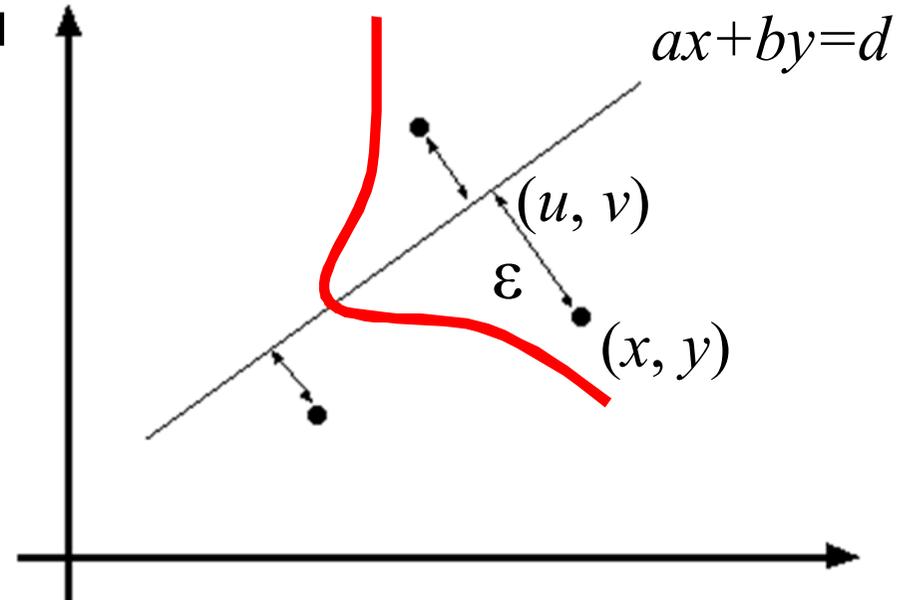
- **Модель:** точки на линии зашумлены гауссовым шумом в направлении перпендикулярном к линии

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

Точка
на
линии

шум:
нормально –
распределенный
с нулевым матожиданием
и стандартным отклонением σ

нормаль

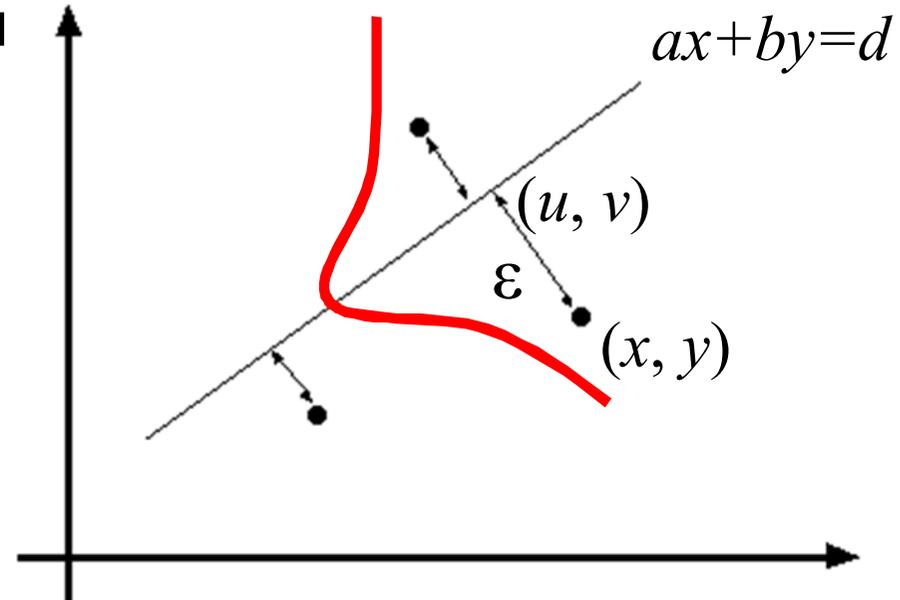




МНК

- **Модель:** точки на линии зашумлены гауссовым шумом в направлении перпендикулярном к линии

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



Правдоподобие точек при параметрах (a, b, d) :

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

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

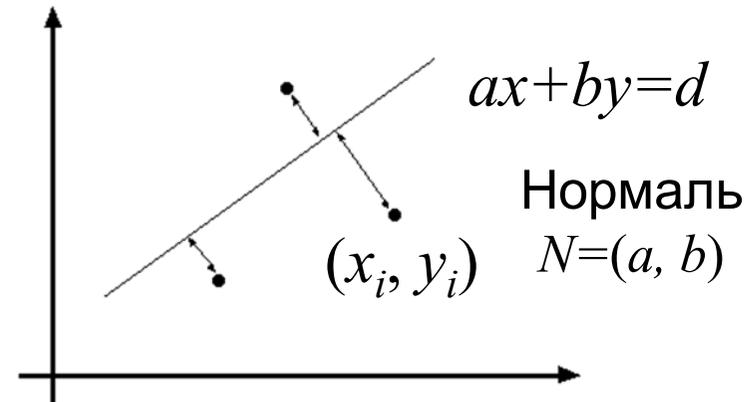


Полные наименьшие квадраты

Расстояние от (x_n, y_n) до линии $ax+by=d$ ($a^2+b^2=1$): $|ax + by - d|$

Найти (a, b, d) минимизирующие:

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



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

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

$$E = \sum_{i=1}^n (a(x_i - \bar{x}) + b(y_i - \bar{y}))^2 = \left\| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = (UN)^T (UN)$$

$$\frac{dE}{dN} = 2(U^T U)N = 0$$

Решение $(U^T U)N = 0$, при условии $\|N\|^2 = 1$: собственный вектор $U^T U$ соответствующий минимальному собственному значению



SVD-разложение

- Singular Value Decomposition: $A = UDV^T$, где
 - U, V - ортогональные
 - D – диагональная (м.б. упорядочены столбцы по убыванию элементов)
 - D – состоит из сингулярных чисел
- Одно из самых важных разложений
- Связь с собственными значениями:

$$A = UDV^T \quad A^T A = VDU^T UDV^T$$
$$A^T A = VDDV^T = VD^2V^T$$

- Сингулярные значения – корни квадратные собственных значений матрицы $A^T A$



Наименьшие квадраты

- Имеем уравнение $Ap = 0$ при $\|p\| = 1$
- Решим с помощью SVD-разложения: $A = UDV^T$
- Нужно минимизировать $\|UDV^T p\|$
- Однако: $\|UDV^T p\| = \|DV^T p\|$ $\|V^T p\| = \|p\|$
- Тогда нам нужно минимизировать $\|DV^T p\|$ при $\|V^T p\| = 1$
- Обозначим $y = V^T p$
- Тогда нам нужно минимизировать $\|Dy\|$ при $\|y\| = 1$
 - D – диагональная, столбцы упорядочены по убыванию!
- $y = (0, \dots, 0, 1)^T$, тогда $p = Vy$ – последний столбец V



Полные наименьшие квадраты

$$U = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \quad U^T U = \begin{bmatrix} \sum_{i=1}^n (x_i - \bar{x})^2 & \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^n (y_i - \bar{y})^2 \end{bmatrix}$$

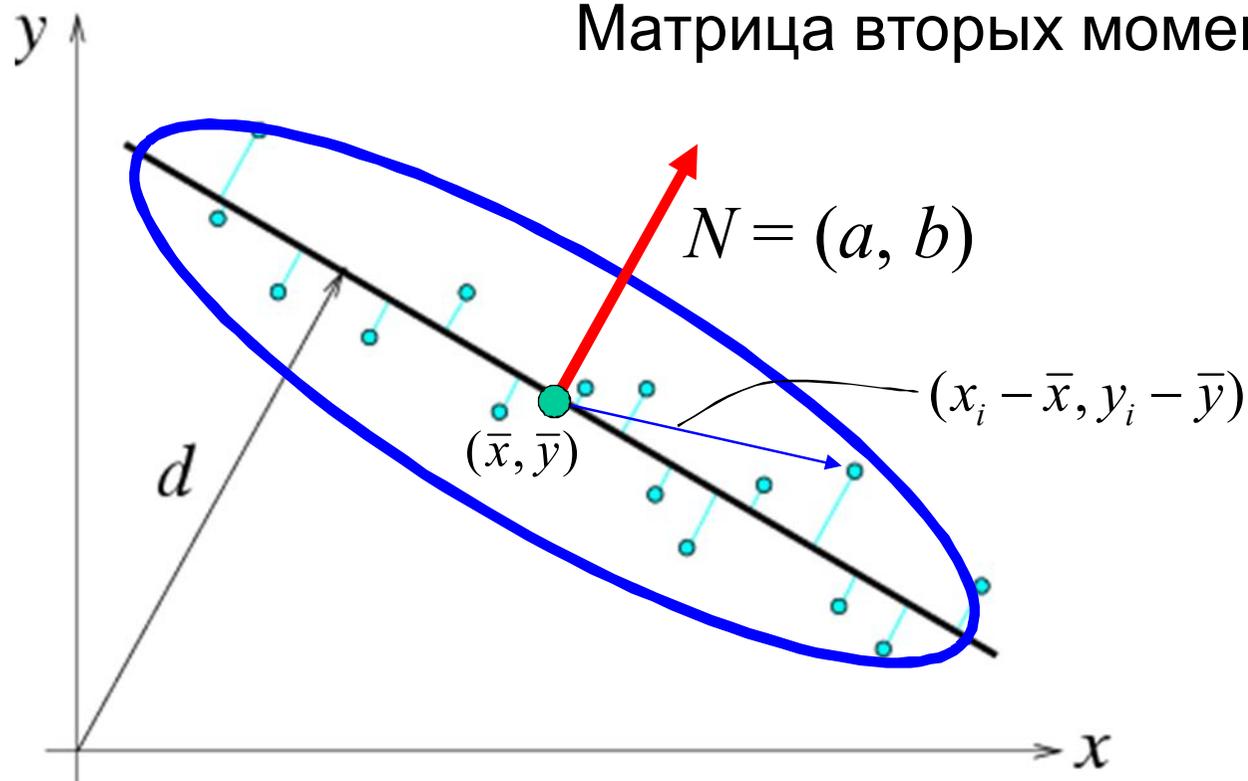
Матрица вторых моментов



Полные наименьшие квадраты

$$U = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \quad U^T U = \begin{bmatrix} \sum_{i=1}^n (x_i - \bar{x})^2 & \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^n (y_i - \bar{y})^2 \end{bmatrix}$$

Матрица вторых моментов





Полные НК

Максимальный собственный вектор матрицы

$$Cov = \begin{bmatrix} \sum_i (x_i - \mu_x)^2 & \sum_i (x_i - \mu_x)(y_i - \mu_y) \\ \sum_i (x_i - \mu_x)(y_i - \mu_y) & \sum_i (y_i - \mu_y)^2 \end{bmatrix}$$

Задаёт направление прямой \vec{l} , минимальный – направление нормали (a, b)

Прямая проходит через среднюю точку (μ_x, μ_y)

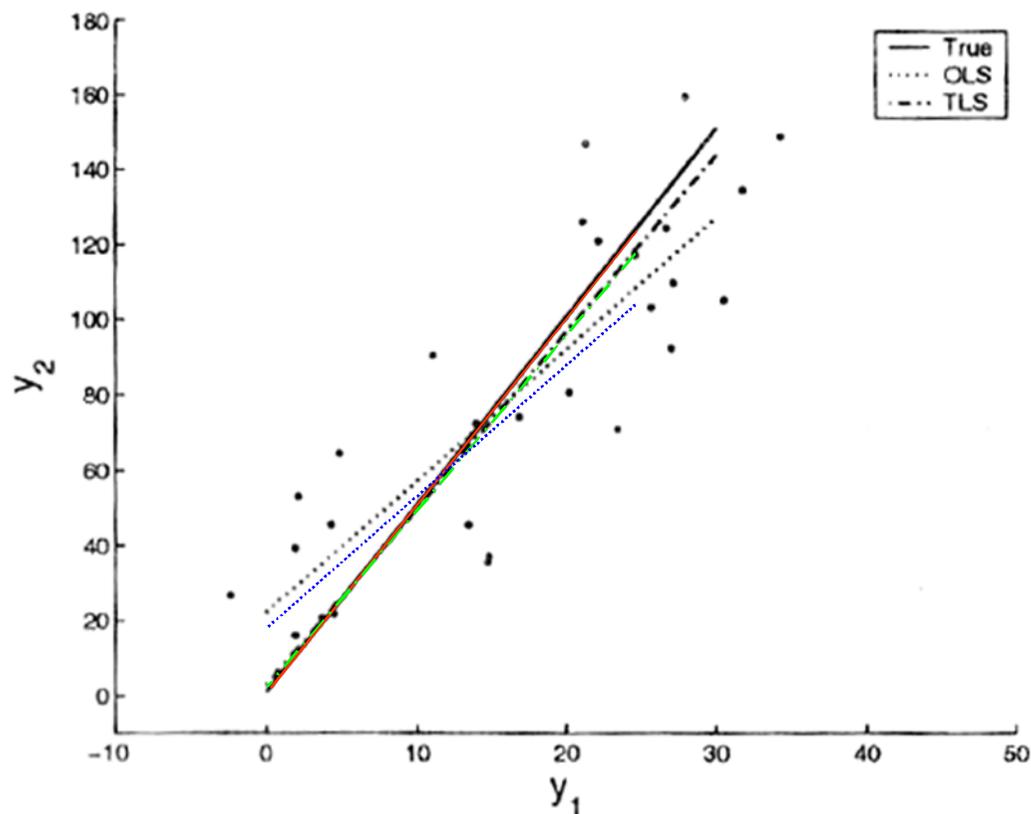
$$\mu_x = \frac{1}{N} \sum_i x_i; \mu_y = \frac{1}{N} \sum_i y_i$$

$$c = -(a\mu_x + b\mu_y)$$



МНК и метод полных НК

Пример отличия:

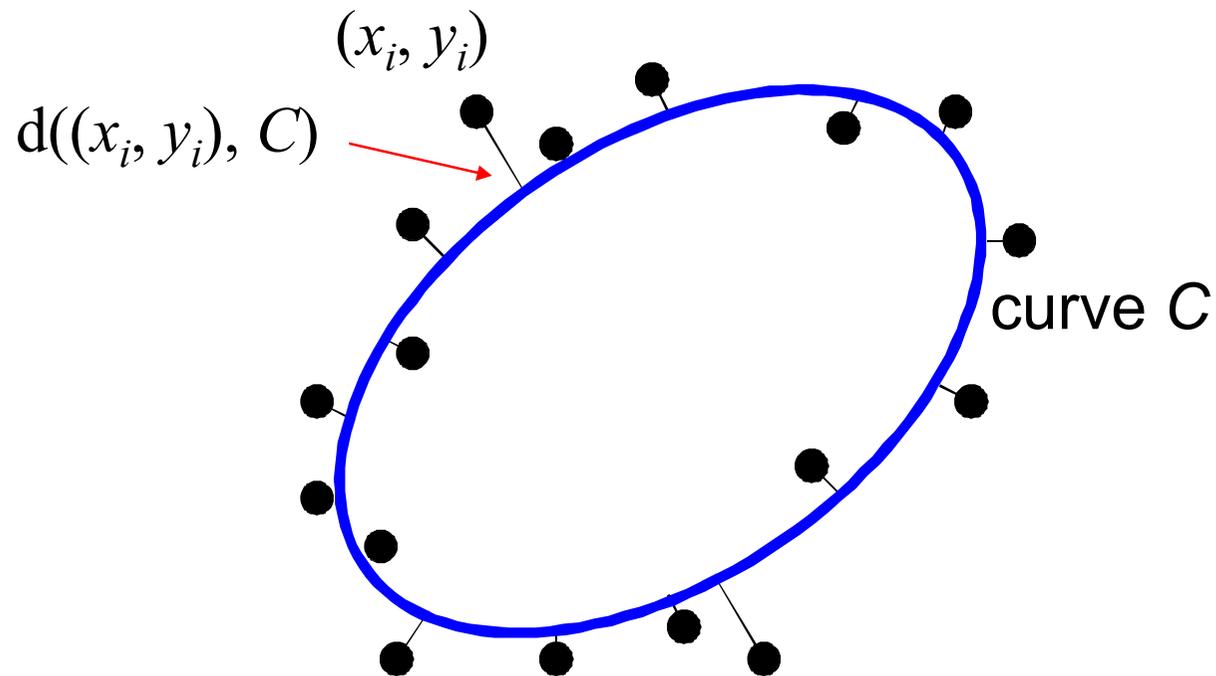


- Исходная прямая
- - - метод полных НК
- ... МНК



Наименьшие квадраты для кривых

- Минимизируем сумму квадратов геометрических расстояний от точек до кривой





Наименьшие квадраты для коники

- Формула коники:

$$C(\mathbf{a}, \mathbf{x}) = \mathbf{a} \cdot \mathbf{x} = ax^2 + bxy + cy^2 + dx + ey + f = 0,$$

$$\mathbf{a} = [a, b, c, d, e, f],$$

$$\mathbf{x} = [x^2, xy, y^2, x, y, 1]$$

- Минимизация геометрических расстояний – нелинейная оптимизация даже для коник
- *Алгебраическое расстояние: $C(\mathbf{a}, \mathbf{x})$*
- Алгебраическое расстояние минимизируется НК:

$$\begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & x_2 y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & & & & \ddots & \vdots \\ x_n^2 & x_n y_n & y_n^2 & x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = 0$$



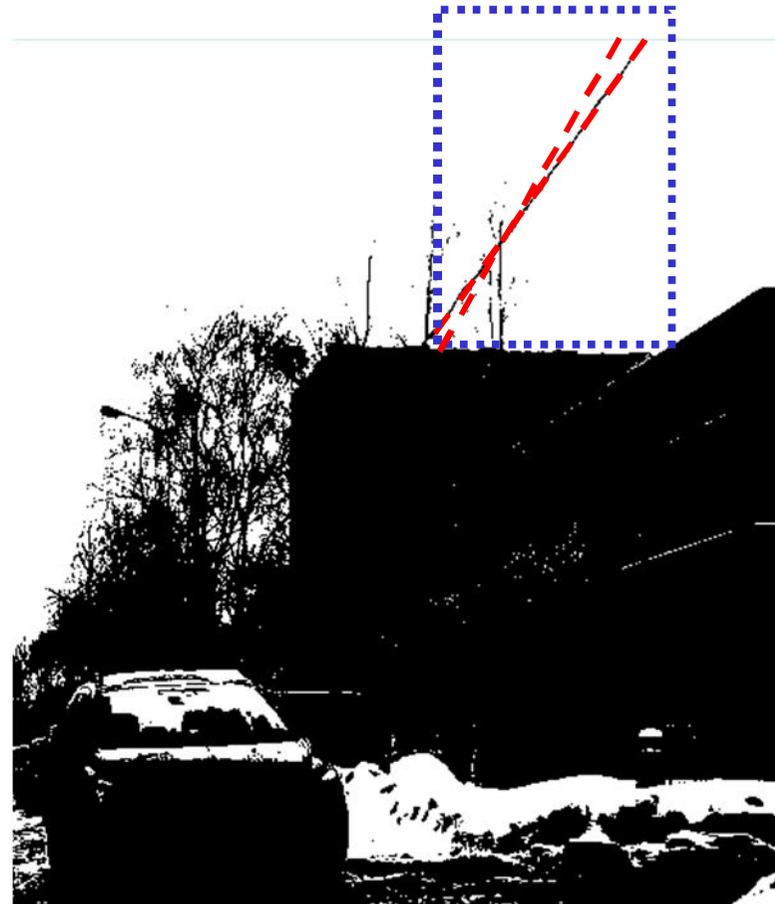
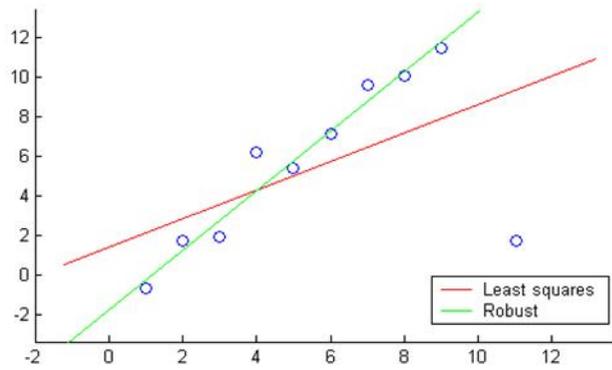
МНК для коник

- СЛАУ: $Da = 0$
- Нужно ограничение для исключения тривиального решения
- Дискриминант: $b^2 - 4ac$
 - Отрицательный: эллипс
 - Ноль: парабола
 - Положительный: гипербола
- Минимизация квадрата алгебраического расстояния с ограничениями ведет к обобщенной задаче на собственные значения (есть ряд методов)
- Подробнее тут:
 - A. Fitzgibbon, M. Pilu, and R. Fisher, [*Direct least-squares fitting of ellipses*](#), IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(5), 476--480, May 1999



Проблема выбросов

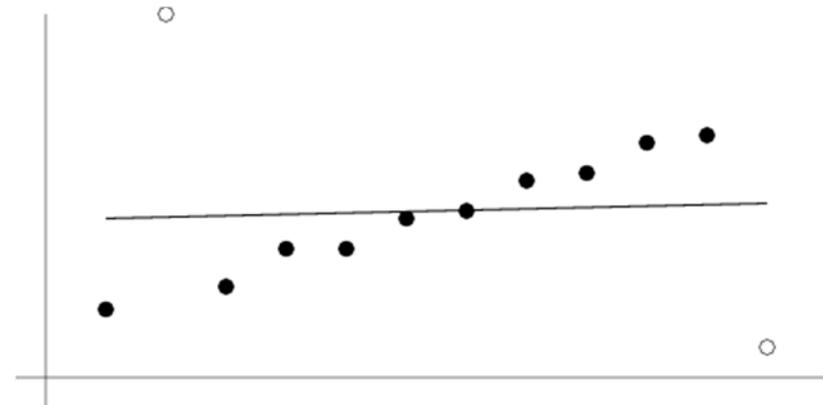
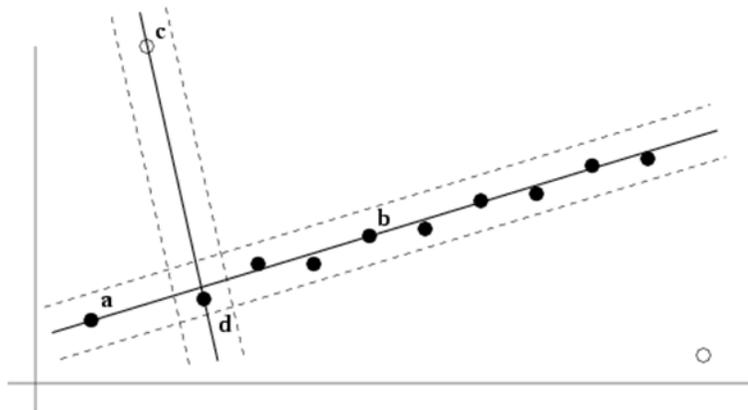
Что, если предположение о характере шума нарушается?





Выбросы

- Если часть данных x не порождена моделью F , то при оценке методом максимального правдоподобия результат может быть сколь угодно далек от истинного
- Такие данные называются выбросы (outliers)





M-оценки

Идея – уменьшить влияние “далеких” точек

Возьмем параметризацию:

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

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

Модифицируем целевую функцию:

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

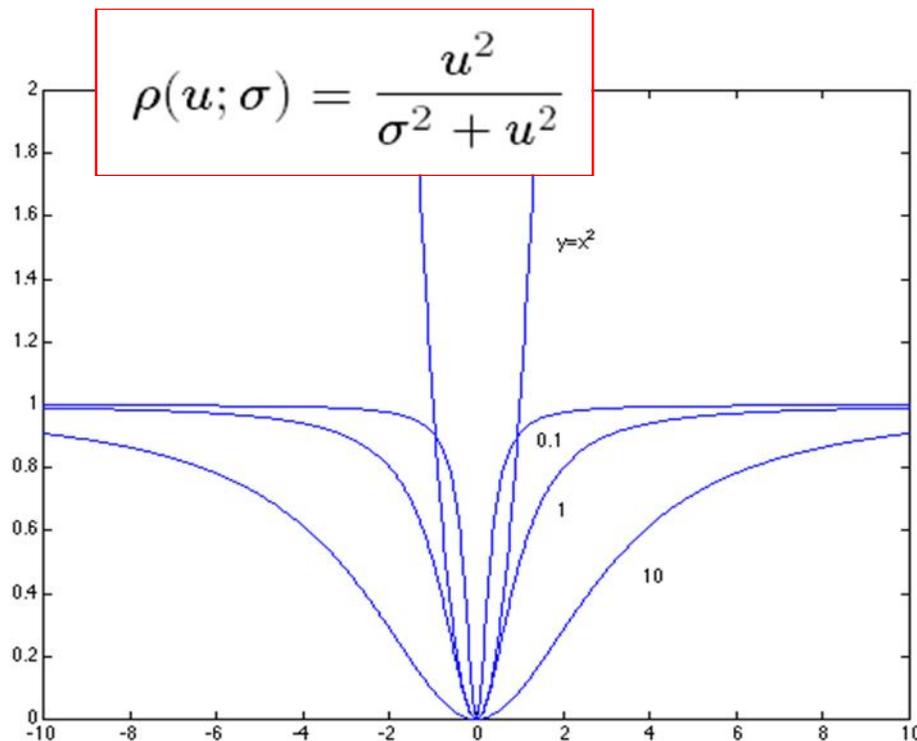
$$\varepsilon_i = x_i \cos \theta + y_i \sin \theta - R$$

Если $\rho(\varepsilon) = \varepsilon^2$ - получаем МНК



Робастные оценки

- Обычно: минимизируем $\sum \rho(r_i(x_i, \theta); \sigma)$
- $r_i(x_i, \theta)$ – невязка i -ой точки при условии параметров модели θ
 ρ – робастная функция с параметром масштаба σ

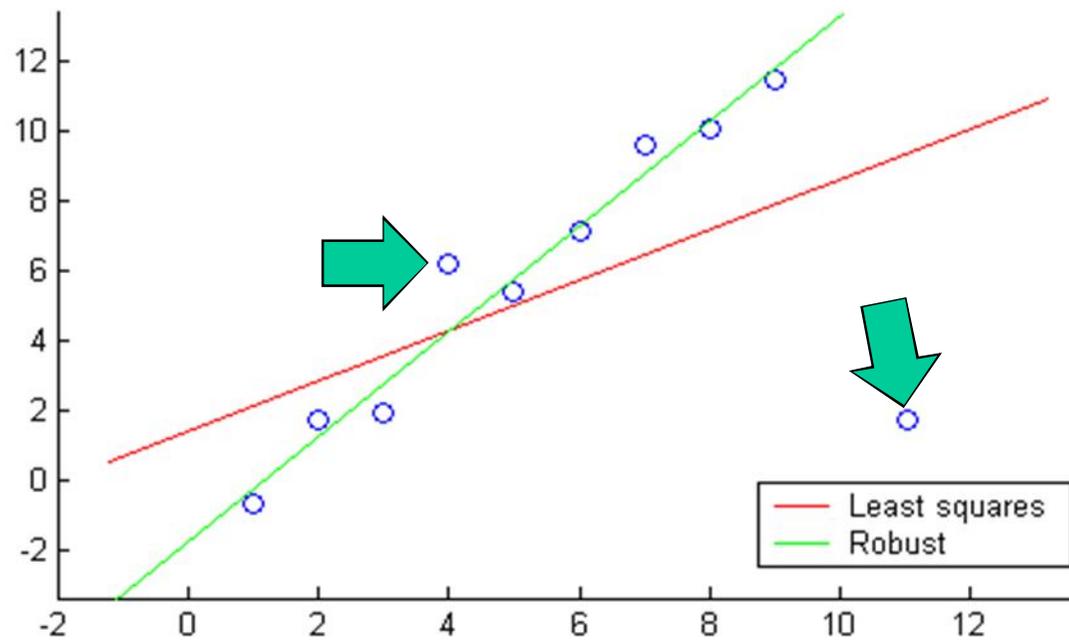


Робастная функция ρ ведёт себя как квадрат расстояния при малых значениях u но выравнивается при больших значениях u



M-оценки

Результат:



Равный вклад в целевую функцию



M-оценки

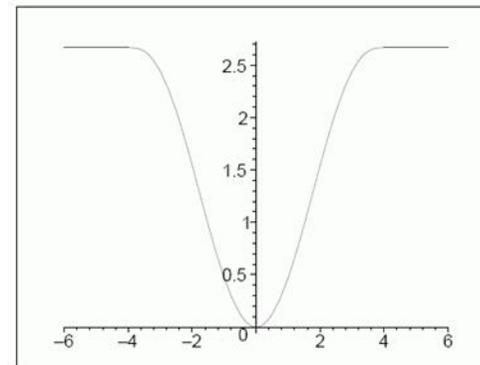
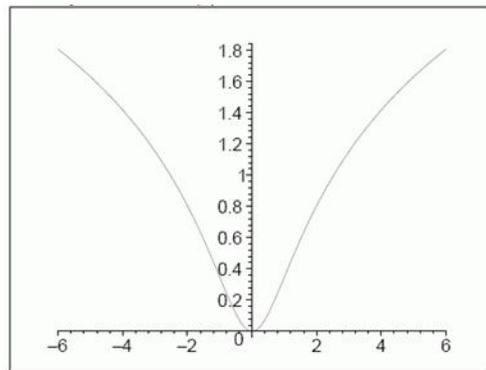
Часто используемые варианты функции ρ :

Тьюки
$$\rho(\varepsilon) = \begin{cases} K^2/6 \left[1 - \left(1 - \left(\frac{\varepsilon}{K} \right)^2 \right)^3 \right] & |\varepsilon| \leq K \\ K^2/6 & |\varepsilon| > K \end{cases}$$

Коши
$$\rho(\varepsilon) = \frac{c^2}{2} \log \left[1 + \left(\frac{\varepsilon}{c} \right)^2 \right];$$

Настройочные константы:

$$K = 4.685; c = 2.385;$$





M-оценки

Вопрос:

- Как найти минимум целевой функции?
 - Это может быть весьма непростой задачей при определенных функциях ρ

Используется метод последовательных приближений – итеративно перевзвешиваемые наименьшие квадраты



Взвешенные наименьшие квадраты

Обычные наименьшие квадраты (ordinary LS):

$$(a, b) = \arg \min_{(a, b)} \sum_i w_i (y_i - ax_i - b)^2$$

Полные наименьшие квадраты (Total LS):

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



Взвешенные НК

Максимальный собственный вектор матрицы

$$Cov = \begin{bmatrix} \sum_i w_i (x_i - \mu_x)^2 & \sum_i w_i (x_i - \mu_x)(y_i - \mu_y) \\ \sum_i w_i (x_i - \mu_x)(y_i - \mu_y) & \sum_i w_i (y_i - \mu_y)^2 \end{bmatrix}$$

Задаёт направление прямой \vec{l} , минимальный – направление нормали (a, b)

Прямая проходит через среднюю точку (μ_x, μ_y)

$$\mu_x = \frac{\sum_i w_i \cdot x_i}{\sum_i w_i}; \quad \mu_y = \frac{\sum_i w_i \cdot y_i}{\sum_i w_i}; \quad c = -(a\mu_x + b\mu_y)$$



Итеративный МНК

Итеративно перевзвешиваемые наименьшие квадраты:

1. Получить начальное приближение методом наименьших квадратов – $\Theta^{(0)} = (\rho^{(0)}, \theta^{(0)})$
2. Номер итерации $t = 1$
3. Для $\Theta^{(t-1)}$ посчитать текущую оценку шума $\sigma^{(t)}$
4. Рассчитать веса точек $w_i^{(t)}$ с учетом функции ρ
5. Используя взвешенные МНК получить Θ_i
6. Если вектор параметров существенно изменяется $\|\Theta^{(t)} - \Theta^{(t-1)}\| > \varepsilon$ - перейти на шаг 3



Оценка масштаба σ (шума)

- Как рассчитать уровень шума σ ?
 - Нужна устойчивая к выбросам оценка

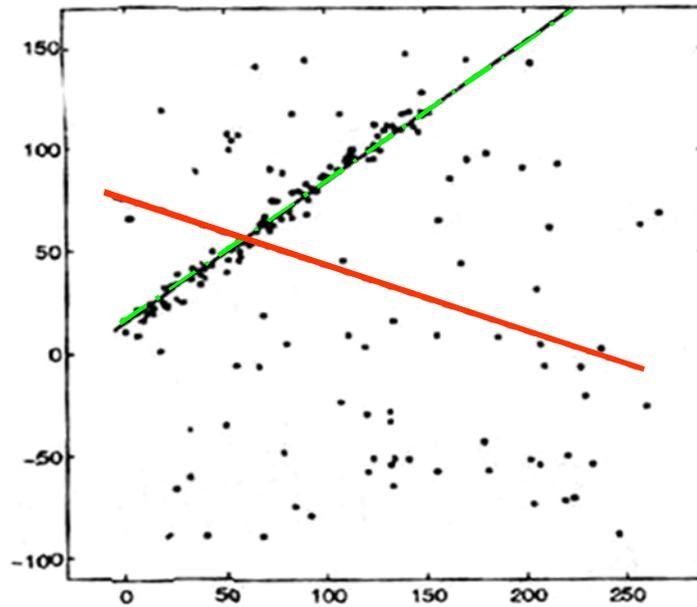
$$\sigma^{(n)} = 1.4826 \operatorname{median}_i |r_i^{(n)}(x_i; \theta^{(n-1)})|$$



Несмещенная (для нормального распределения) робастная оценка средней ошибки

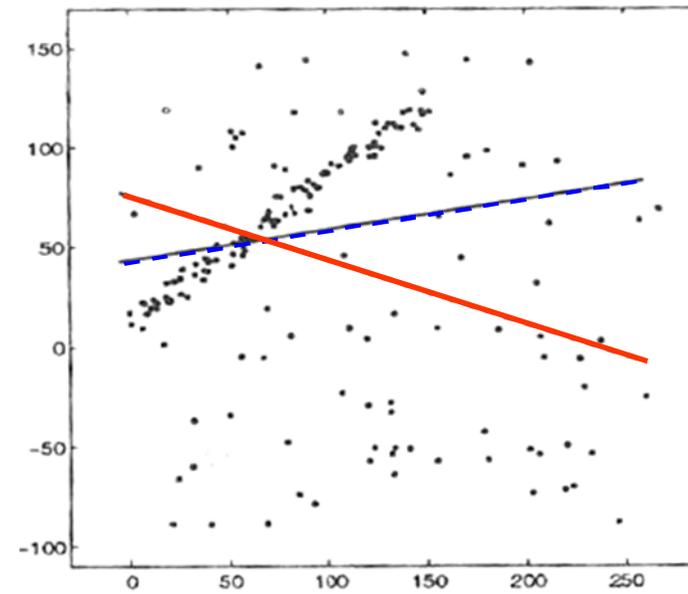


Влияние константы



Первый шаг

$C = 1.5$



$C = 3.5$





Веса

- Как связаны веса точек и функция ρ ?

$$w_i = \frac{\rho'(\varepsilon_i/\sigma)}{\varepsilon_i/\sigma}$$

ТЮКИ

$$w(\varepsilon/\sigma) = \begin{cases} \left[1 - \left(\frac{\varepsilon}{\sigma \cdot K} \right)^2 \right] & |\varepsilon/\sigma| \leq K \\ 0 & |\varepsilon/\sigma| > K \end{cases}$$

КОШИ

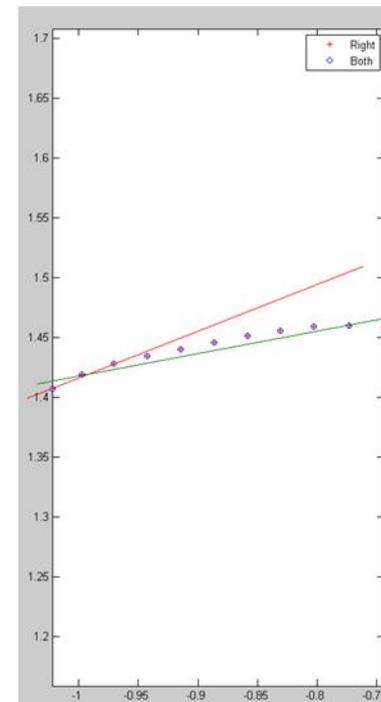
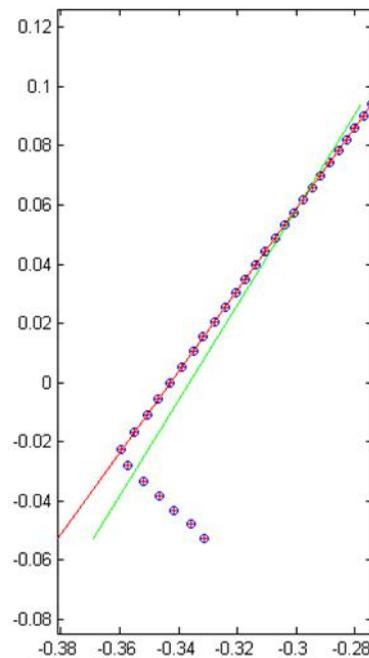
$$w(\varepsilon/\sigma) = \frac{1}{1 + \left(\frac{x}{c \cdot \sigma} \right)^2};$$



M-оценки

Недостатки:

- Необходимость хорошего первого приближения
- Баланс весов – точность vs устойчивость





RANSAC

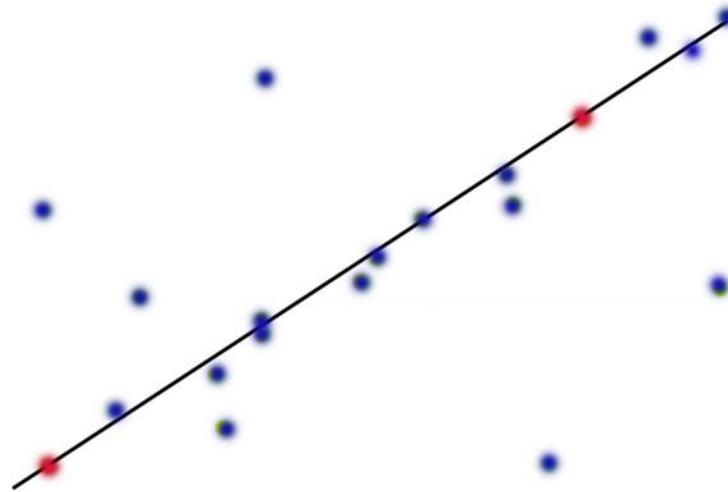
- *Идея* – проведение оценки не по всем данным, а по выборке, не содержащей выбросов
- Поскольку выбросы заранее неизвестны, такую выборку пытаемся получить случайным образом за несколько попыток
- Для снижения вероятности включения выбросов, выборка содержит минимальное количество точек, необходимых для оценки параметров модели
- 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 рассматриваемых схем является оценка модели по небольшой выборке $S \subset X$
- Проблема- количество таких выборок огромно





Базовая схема RANSAC

Повторяем N раз:

- (i) Построение выборки $S \subset X$
- (ii) Построение гипотезы Θ по выборке S
- (iii) Оценка степени согласия гипотезы Θ и набора исходных данных X
- (iv) Выбор наилучшей гипотезы
- (v) Уточнение гипотезы по всем «хорошим» данным

Для заданной доли выбросов γ можно рассчитать количество итераций N , необходимое для получения модели построенной без выбросов с заданной вероятностью P



Функции оценки гипотезы (1)

RANSAC

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

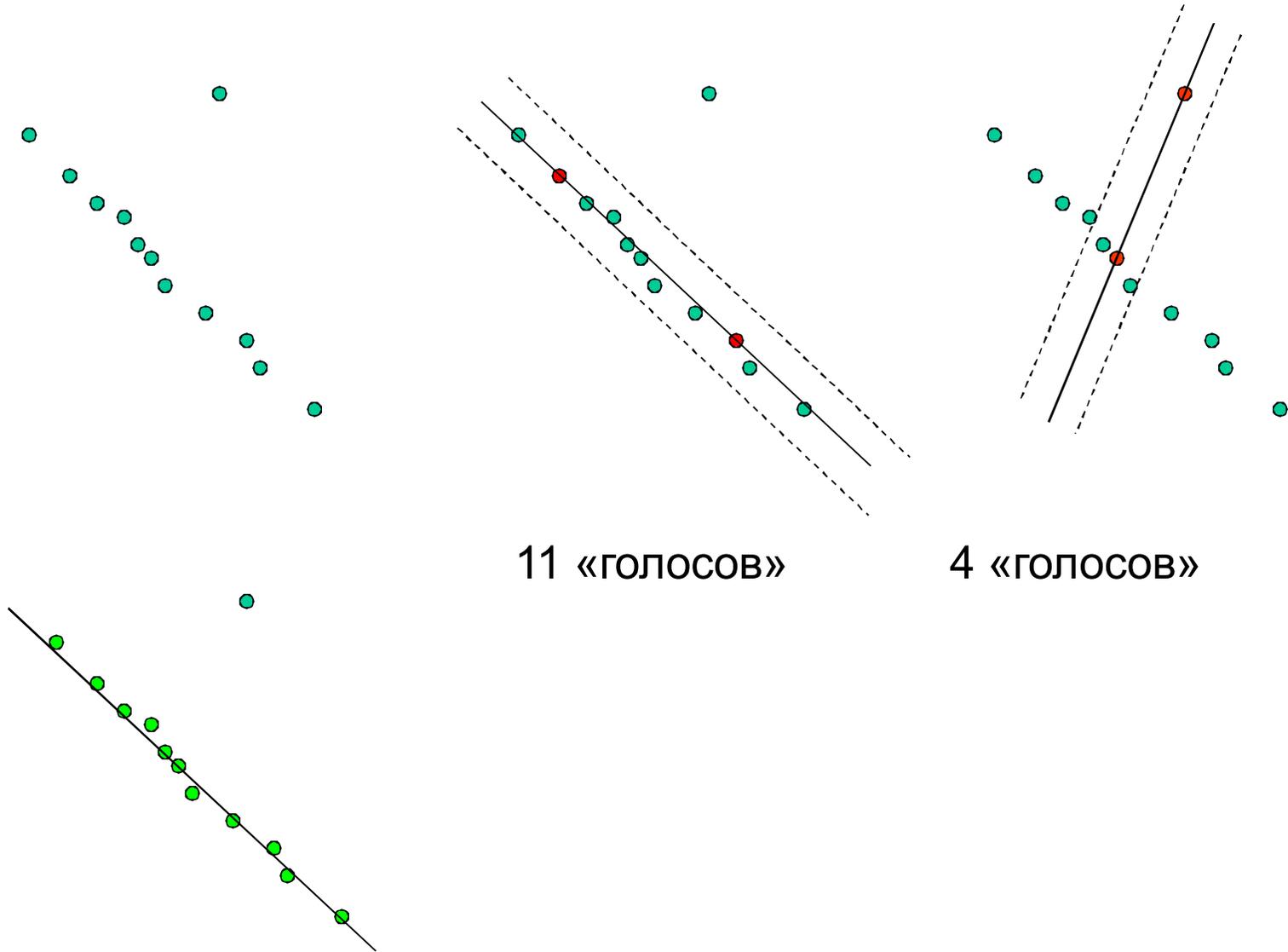
$\varepsilon_i(\theta)$ - невязка i -ой точки и оцениваемой гипотезы

LMS (Least median squares)

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



Пример RANSAC



11 «ГОЛОСОВ»

4 «ГОЛОСОВ»



Выбор параметров

- Размер выборки s
 - Обычно, минимальное количество, достаточное для оценки параметров
- Порог T
 - Выбираем T так, чтобы вероятность инлиера (inlier) была p (пр. 0.95)
 - Обычно гауссова модель шума с нулевым матожиданием и σ : $t^2 = 3.84\sigma^2$
- Количество выборок N
 - Выбираем N так, чтобы вероятность выбрать хотя бы одну выборку без выбросов была не ниже заданной (например, 0.99)
 - Зависит от доли выбросов в данных (доля выбросов: ϵ)



Сколько гипотез нужно проверить?

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

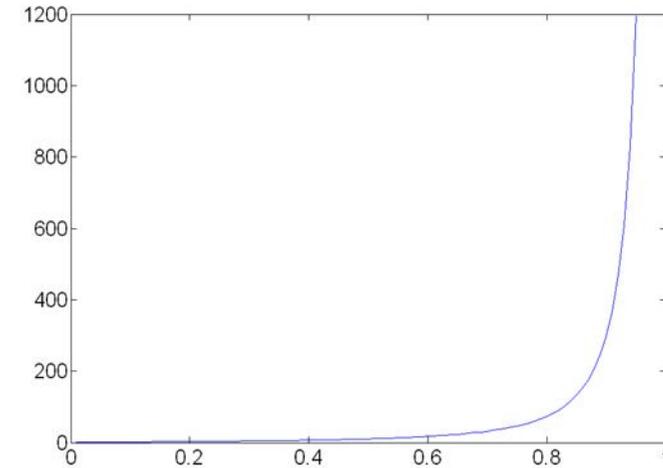
$$N = \log(1 - p) / \log\left(1 - (1 - e)^s\right)$$

- 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



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
 - Set $e = 1 - (\text{number of inliers}) / (\text{total number of points})$
 - Recompute N from e :

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

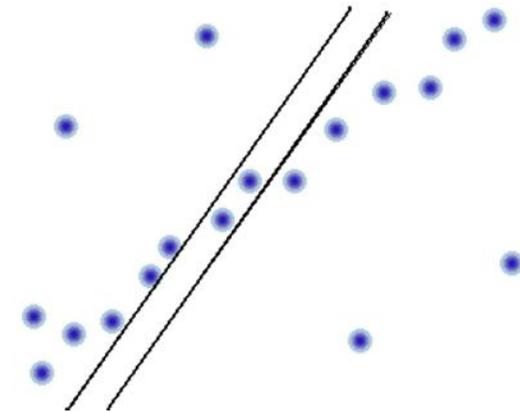
- Increment the $sample_count$ by 1



Функции оценки гипотезы (2)

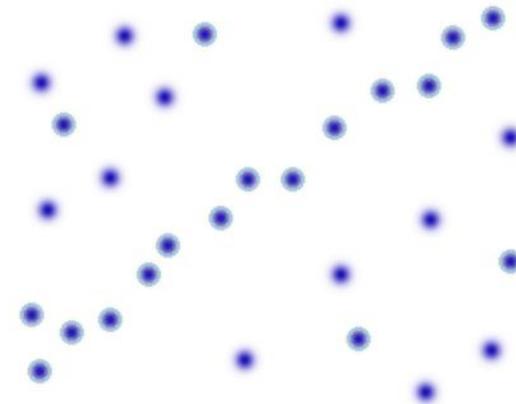
RANSAC

- сильно зависит от выбора порога T



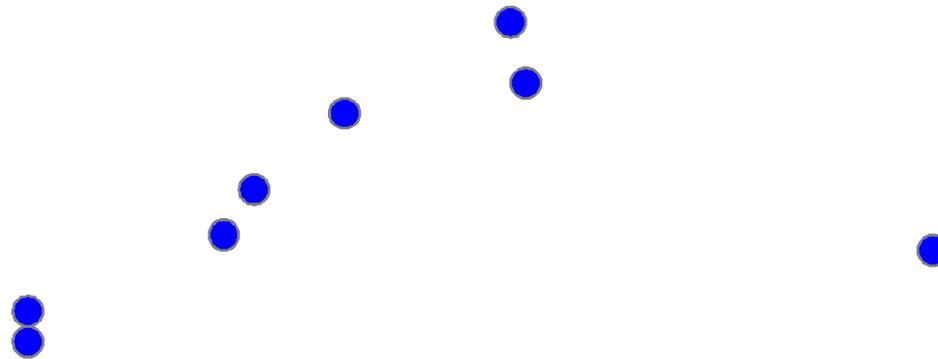
LMS

- Не работает при $>50\%$ выбросов
- Может выбрать ложную гипотезу



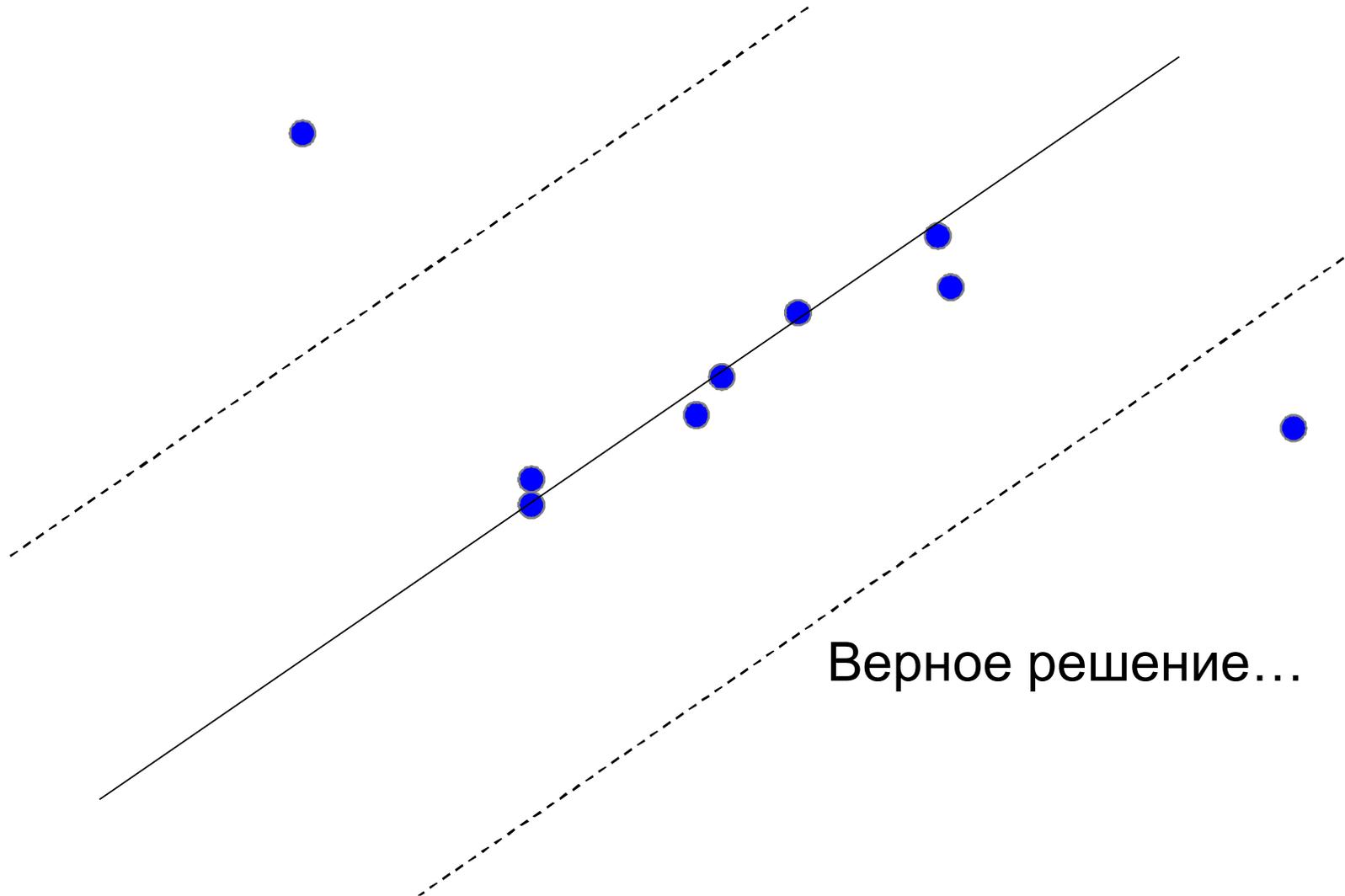


Большой порог – проблема!



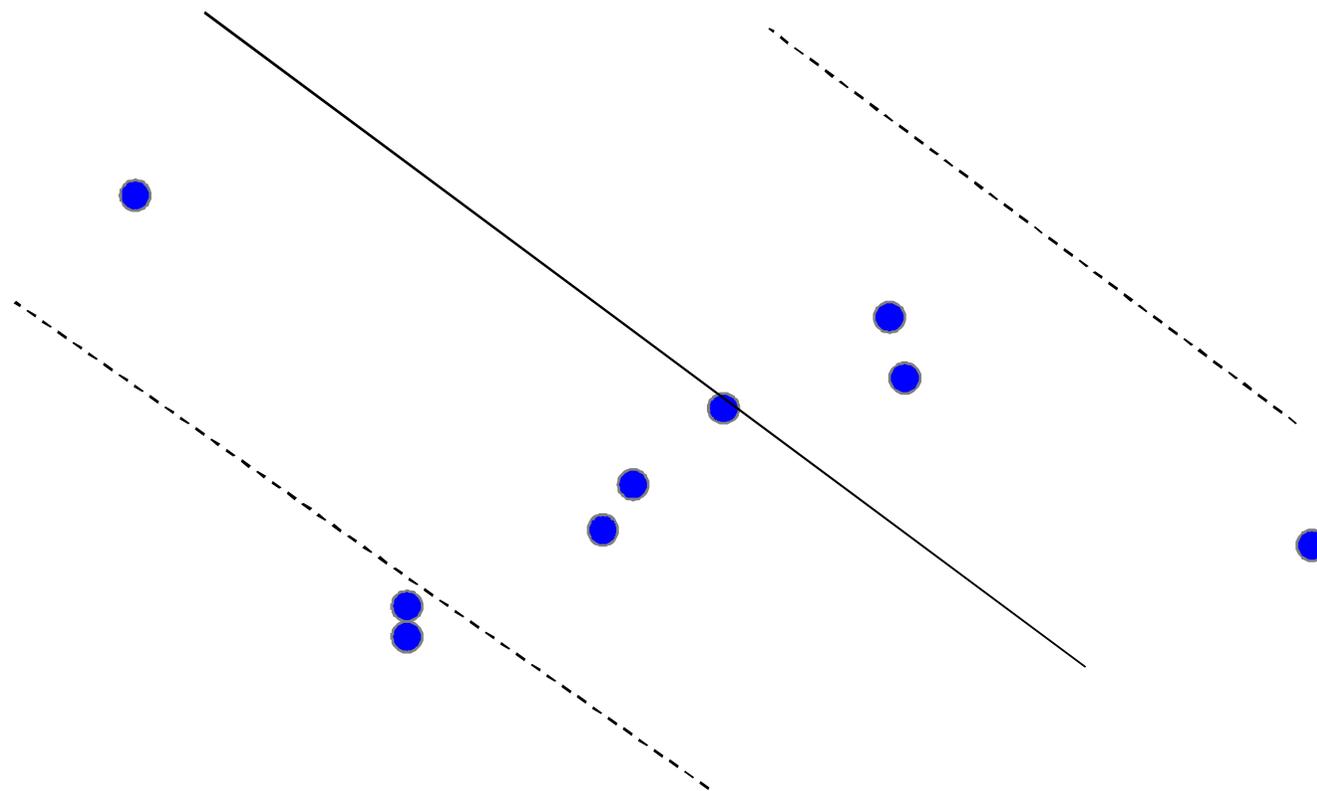


Большой порог – проблема!





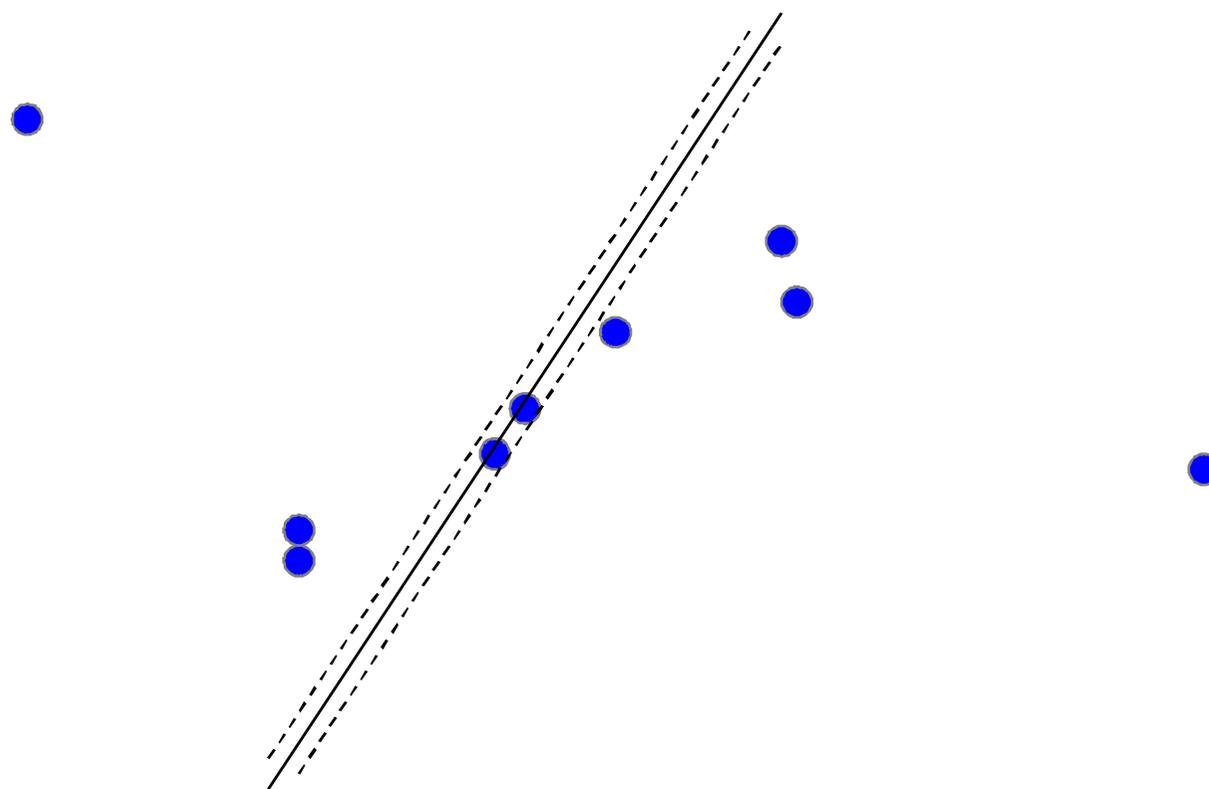
Большой порог – проблема!



...ЭКВИВАЛЕНТНО
НЕВЕРНОМУ

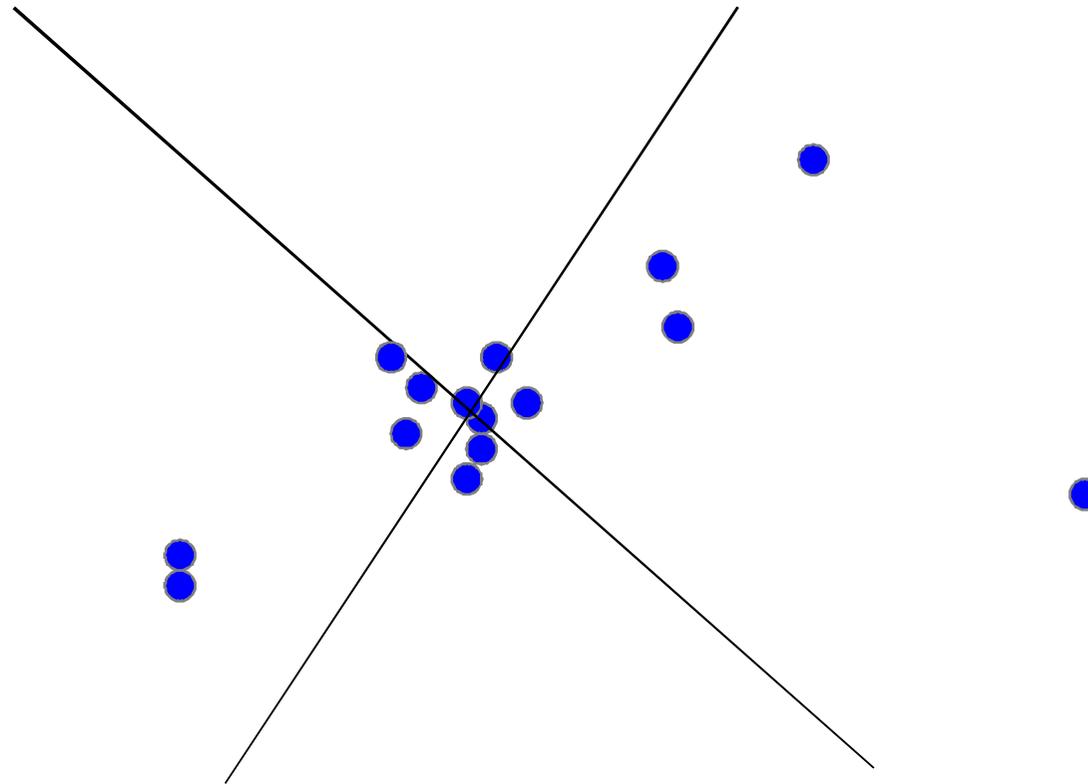


Маленький порог – проблема!





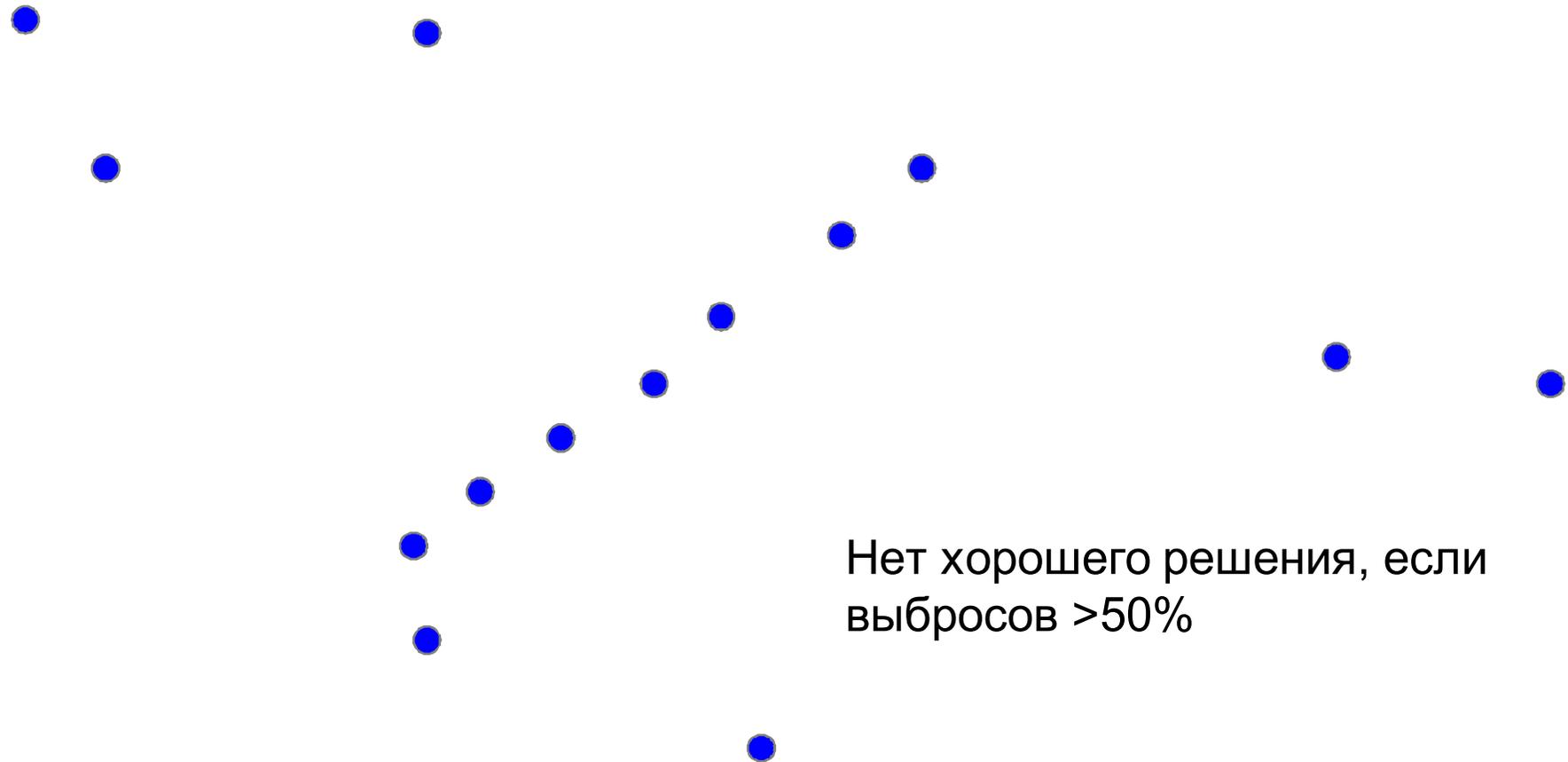
Проблема LMS



Медиана ошибки одинакова
для обоих решений



Con: LMS, subspace problem



Нет хорошего решения, если
выбросов >50%



Функции оценки гипотезы (3)

• M-SAC

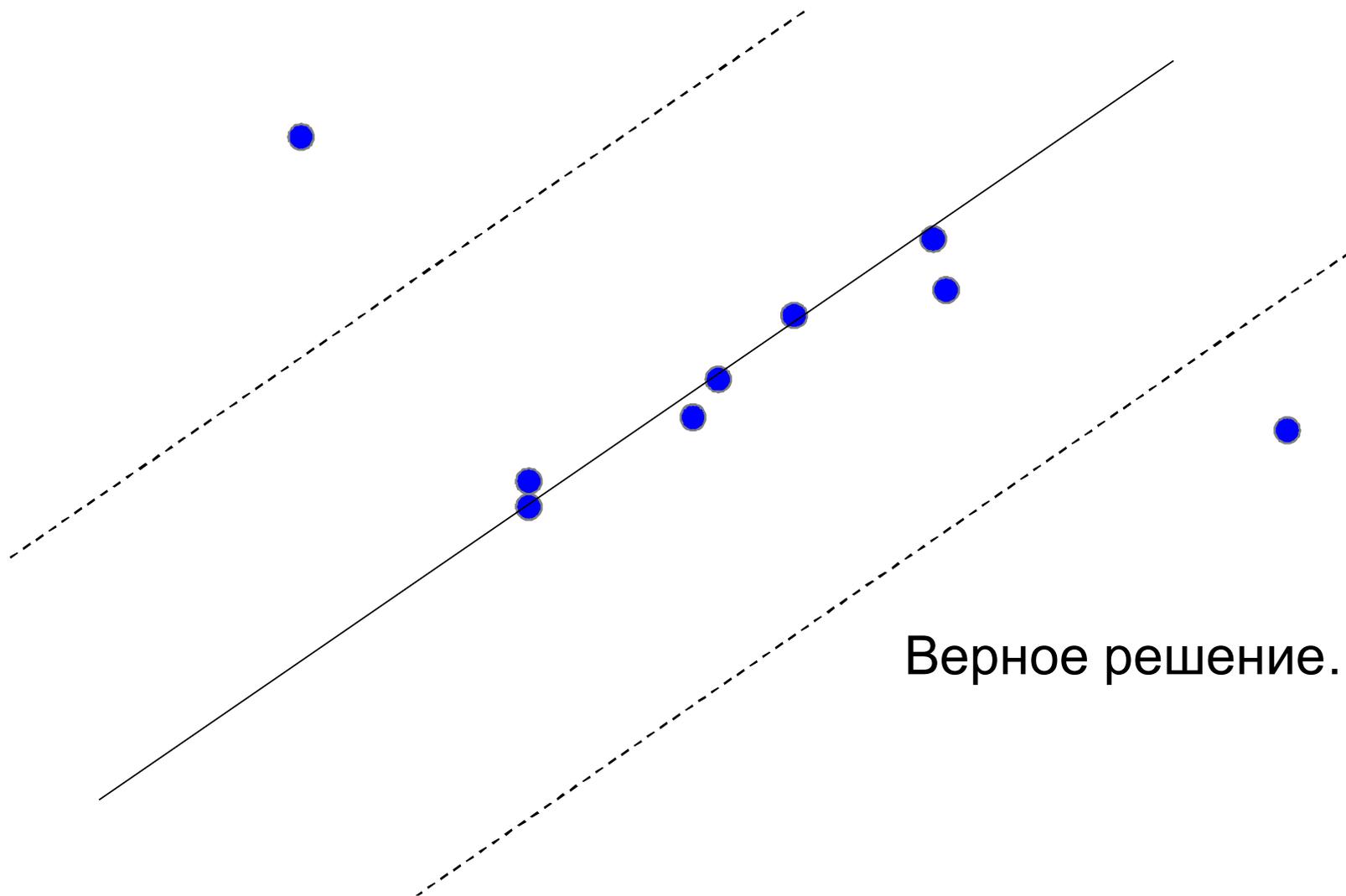
- Возьмём M-оценку в качестве целевой функции:

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

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

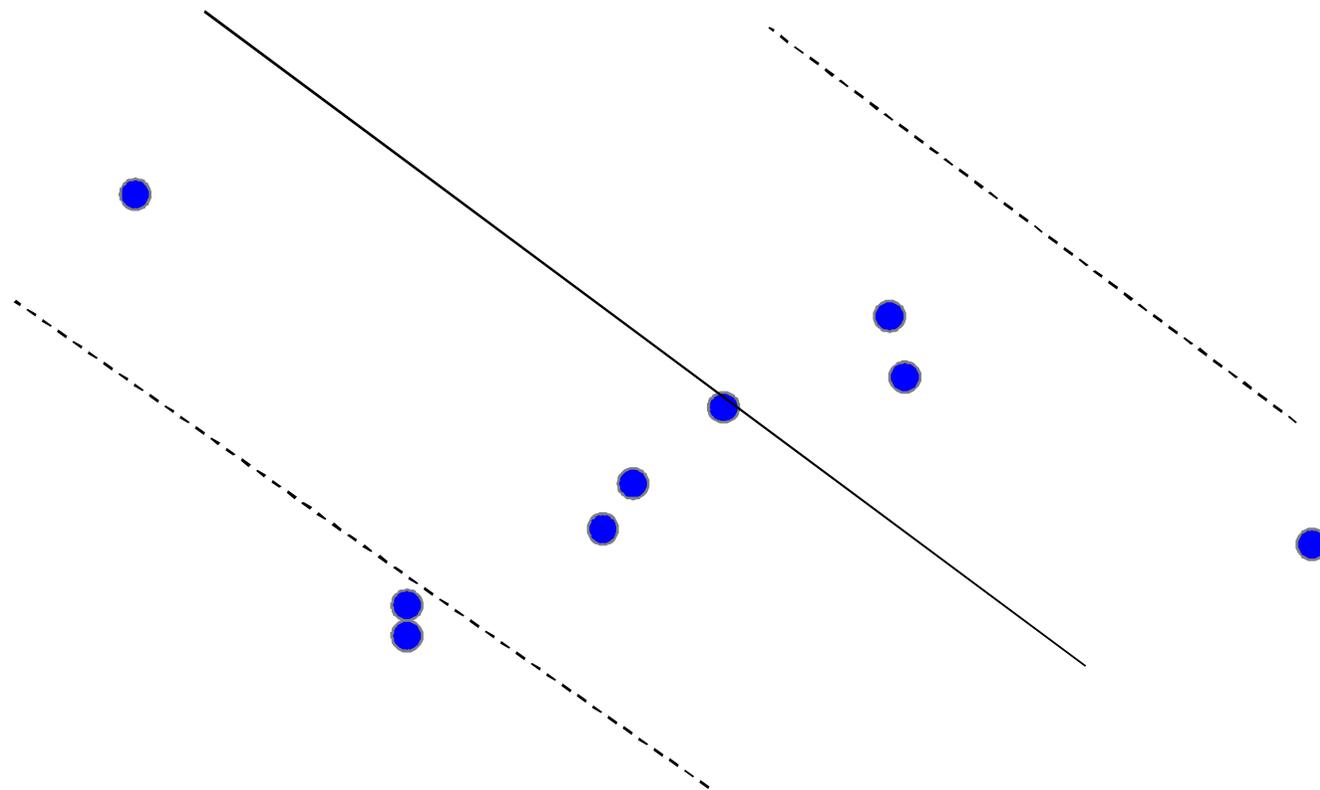


M-SAC





M-SAC



Лучше неверного

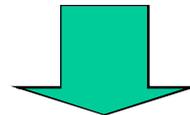
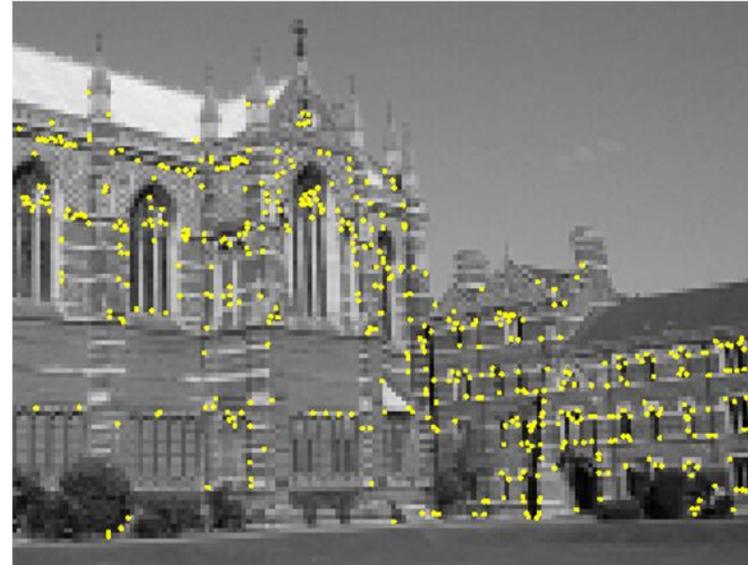
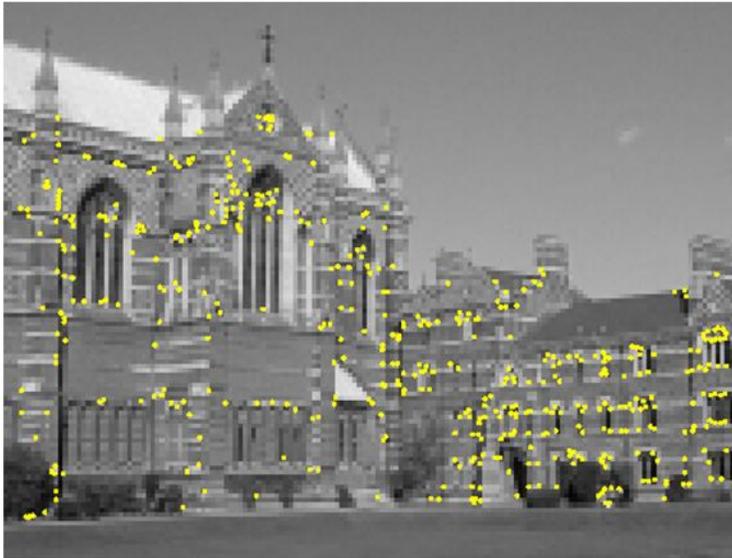


Обобщение на окружности

- Берем три точки
 - Проводим через них окружность
 - Все остальное – точно так же!
-
- Аналогично применяется для всех других моделей. Отличия только в функции оценки параметров модели, размере выборки



Сопоставление изображений



Сопоставление по
дескрипторам –
МНОГО ЛОЖНЫХ

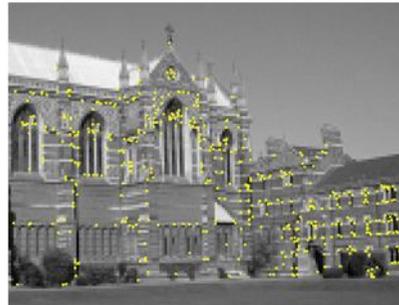
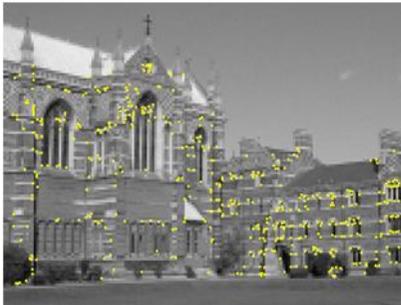


Фильтрация ложных соответствий

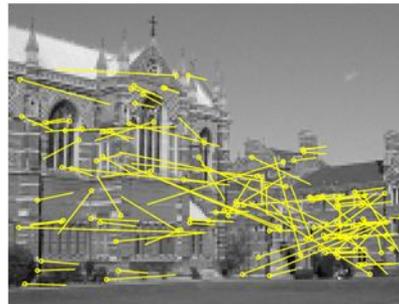
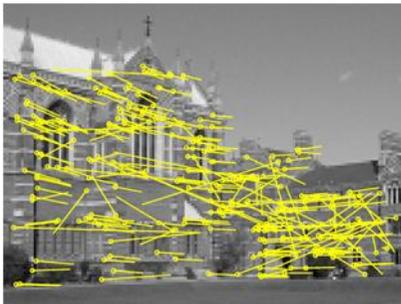
- (m, m') – пары соответствующих точек
- Как отделить правильные соответствия от неправильных?
- Используем модель преобразования H из M в M' для фильтрации точек
- Оценим модель с помощью RANSAC
- Отфильтруем выбросы по модели
- Уточним модель по всем оставшимся точкам
 - Iterative least-squares
 - Non-linear minimization



Ложные соответствия

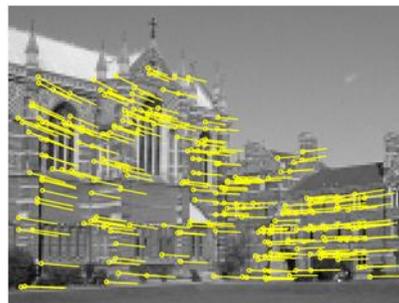


500 особенностей



Соответствия (268)

Выбросы (117)

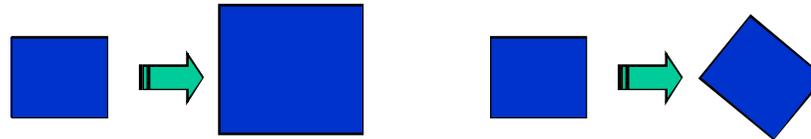


**Хороших
соответствий (151)**

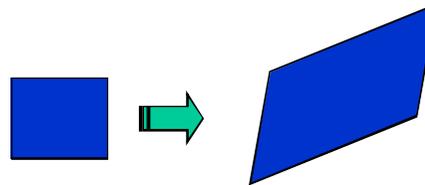


Модели преобразований в 2D

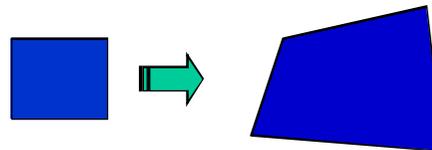
- Подобие
(перенос, масштаб, поворот)



- Аффинное



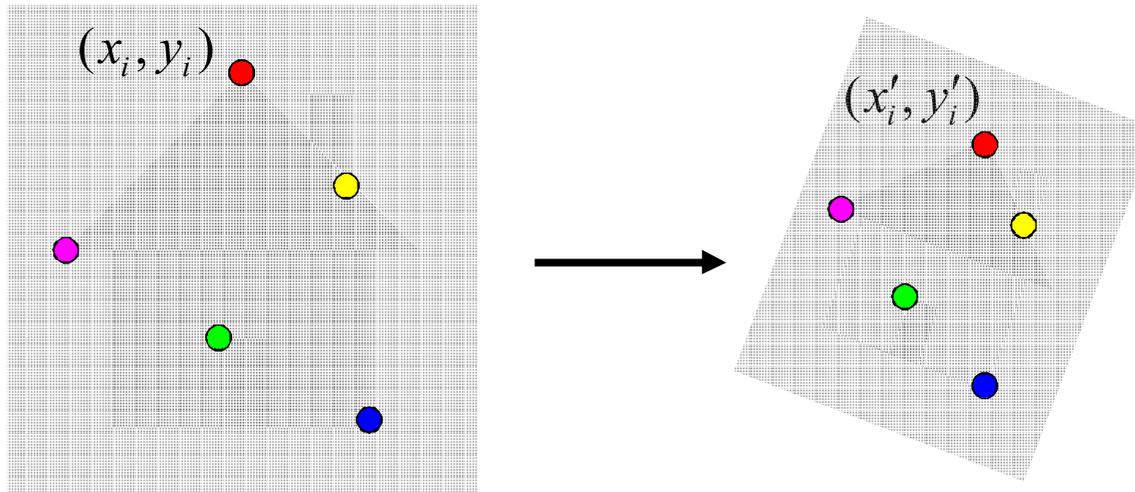
- Проективное
(гомография)





Оценка аффинного преобразования

- Как вычислить преобразование по известным соответствиям?



$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

$$\begin{bmatrix} x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ \dots & & & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \dots \\ x'_i \\ y'_i \\ \dots \end{bmatrix}$$



Оценка аффинного преобразования

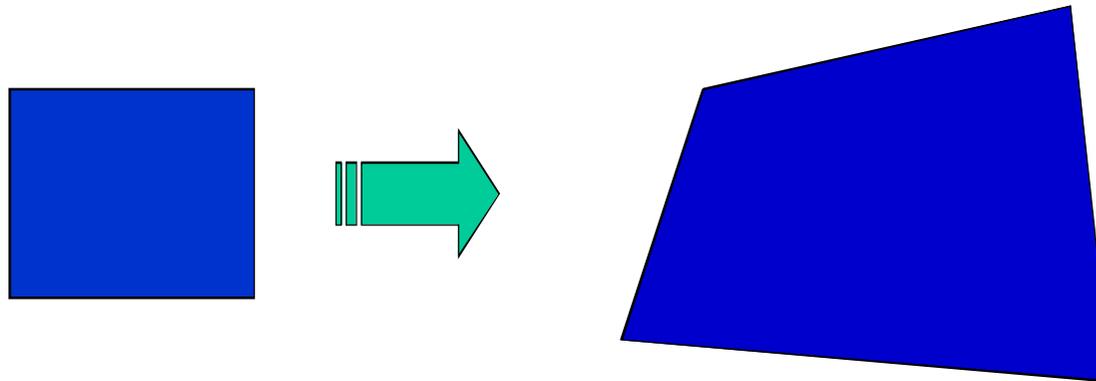
$$\begin{bmatrix} & & \dots & & & & \\ x_i & y_i & 0 & 0 & 1 & 0 & \\ 0 & 0 & x_i & y_i & 0 & 1 & \\ & & \dots & & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \dots \\ x'_i \\ y'_i \\ \dots \end{bmatrix}$$

- Линейная система с 6ю неизвестными
- Каждое соответствие дает нам 2 линейно-независимых уравнения
- Нужно 3+ соответствий для оценки преобразования



Гомография

- Перспективное преобразование плоскости
- Переводит четырехугольник в другой произвольный четырехугольник





Гомография

- Преобразование между 2мя разными видами одной и той же плоскости



- Преобразование между видами с повернутой камеры (центр проекции общий)





Однородные координаты

- Однородные координаты

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

Перевод в однородные
координаты

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

Перевод из однородных
координат

Удобнее представлять себе так:

$$[wx \quad wy \quad w]^T \cong [x \quad y \quad 1]^T \cong [x \quad y]^T$$

Подробнее будет в лекциях про геометрические свойства камер!



Оценка гомографии

$$\lambda \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad \lambda x' = Hy$$

Уравнение гомографии

- Каждая пара соответствующих точек $\{(x,y),(x',y')\}$ даёт 2 линейно-независимых уравнения на параметры H
- H определена с точностью до масштаба
 - Можем поставить ограничение $\|H\|=1$ или $h_{33}=1$
- 4х пар соответствующих точек достаточно для составления системы уравнений
- Задача сводится к задаче наименьших квадратов



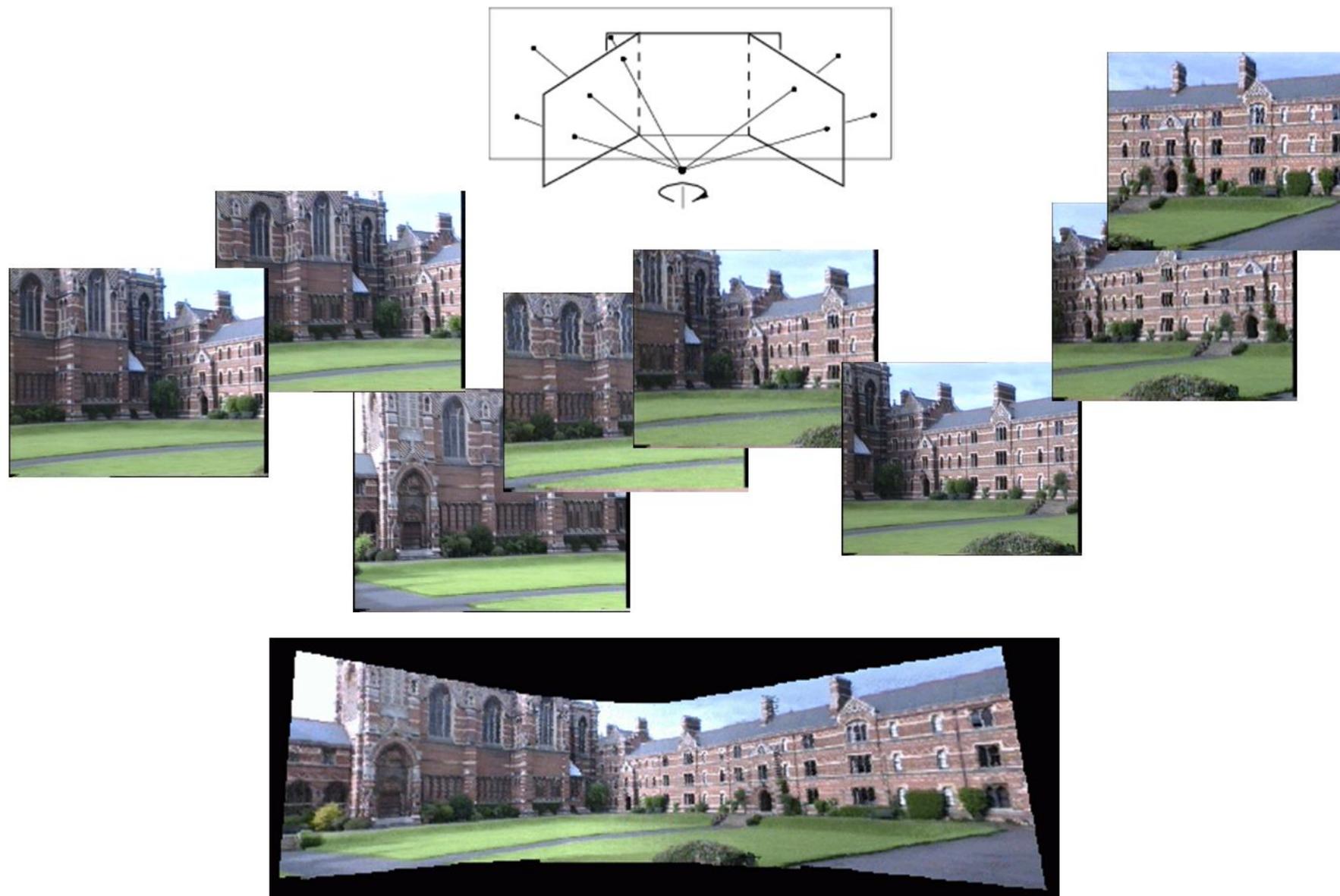
Ректификация изображения



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



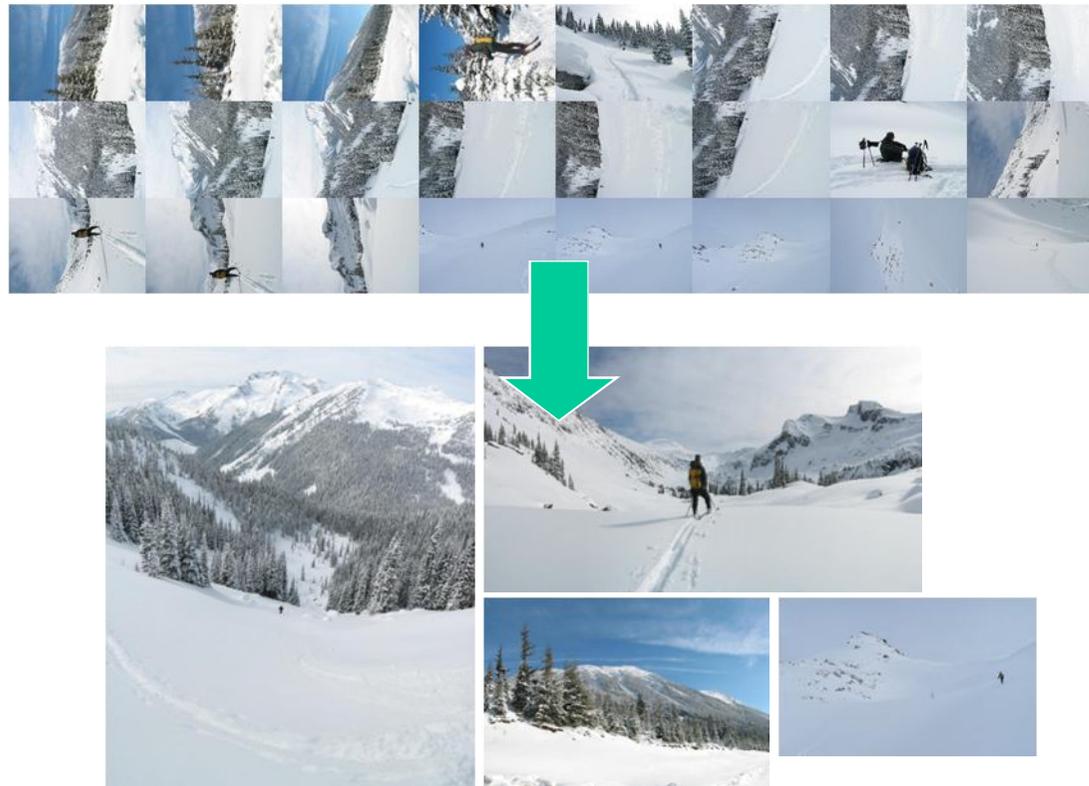
Приложение: склейка панорам





Распознавание панорам

- Из неупорядоченного набора фотографий определить, какие относятся к какой панораме и сшить их



M. Brown and D. Lowe, ["Recognizing Panoramas,"](http://www.cs.ubc.ca/~mbrown/panorama/panorama.html) ICCV 2003.
<http://www.cs.ubc.ca/~mbrown/panorama/panorama.html>



RANSAC pros and cons

- Плюсы
 - Простой и общий метод
 - Применим для множества задач
 - Хорошо работает на практике
- Минусы
 - 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



Схемы голосования

- Пусть каждый признак голосует за все модели, которым он удовлетворяет
- Модели, за которые проголосовало наибольшее количество признаков, считаем результатом
- Надеемся, выбросы/шум не будут голосовать согласованно за определенные модели
- Пропущенные данные не играют большой роли до тех пор, пока оставшихся данных достаточно для голосования за правильные модели



Преобразование Хафа

- Hough transform:
 - Дискретизируем пространство параметров (разделим его на ячейки)
 - Для каждого признака (точки) на изображении, голосуем за все ячейки в пространстве параметров, которые могут породить этот признак (точку)
 - Найдем ячейки с максимум голосов

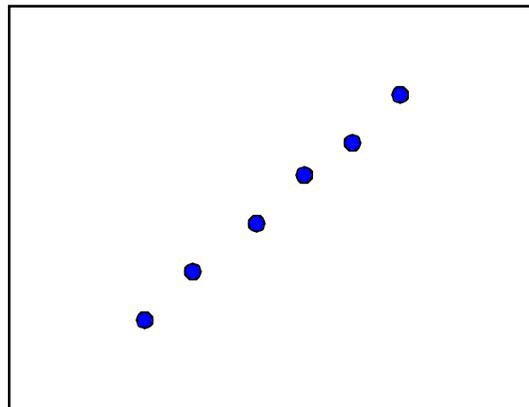
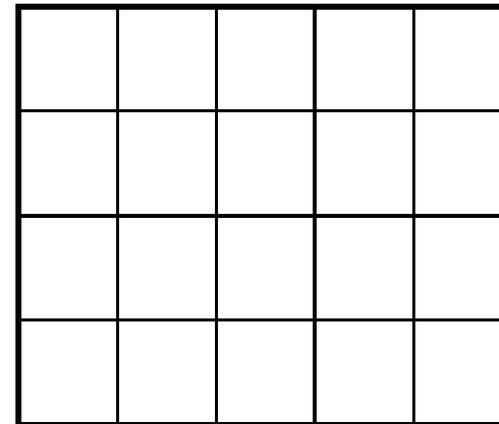
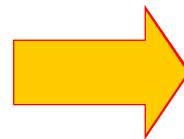


Image space



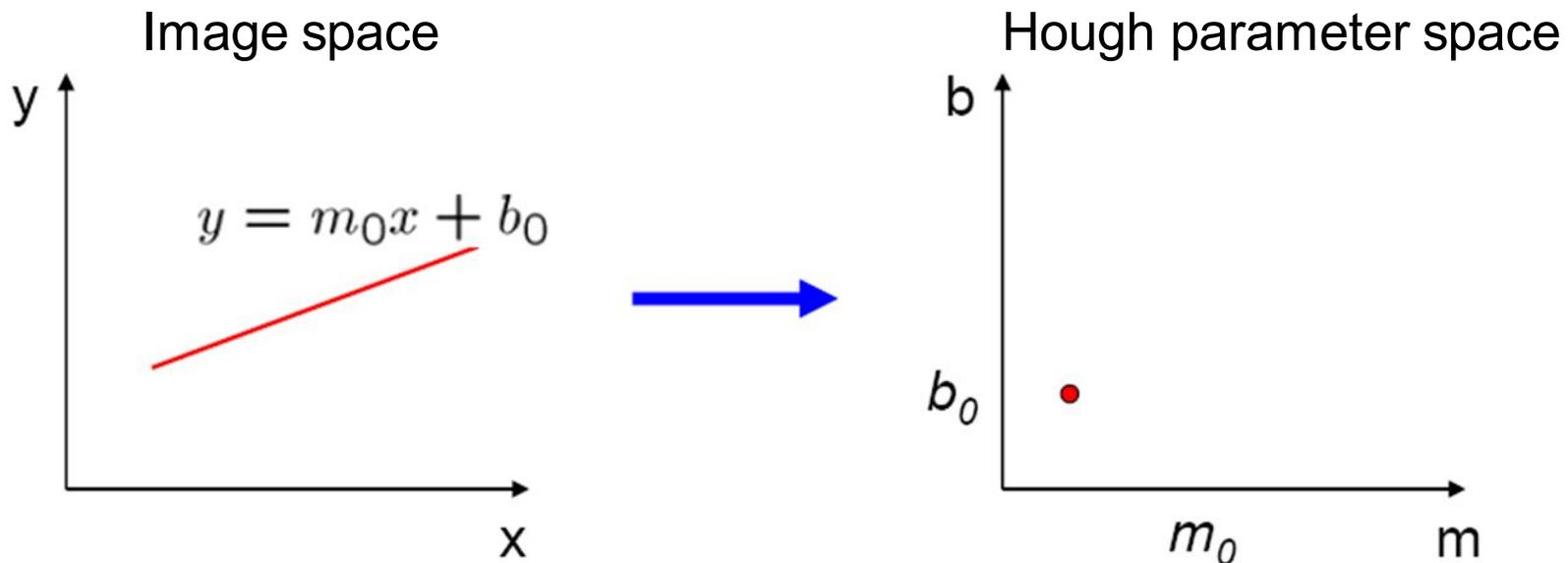
Hough parameter space

P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures*, Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959



Фазовое пространство

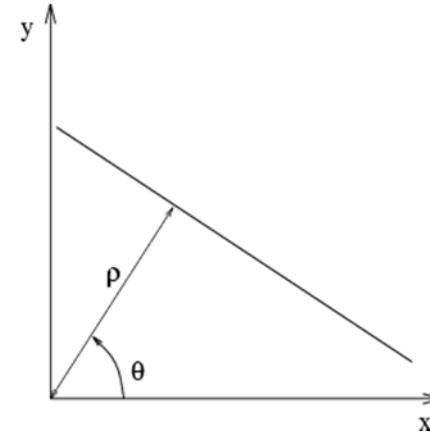
- Пример для линий
- Каждая линия на изображении соответствует точке в пространстве параметров (фазовом пространстве)
- Необходимо, что всевозможные линии на изображении соответствовали ограниченной области в фазовом пространстве





Параметризация линий

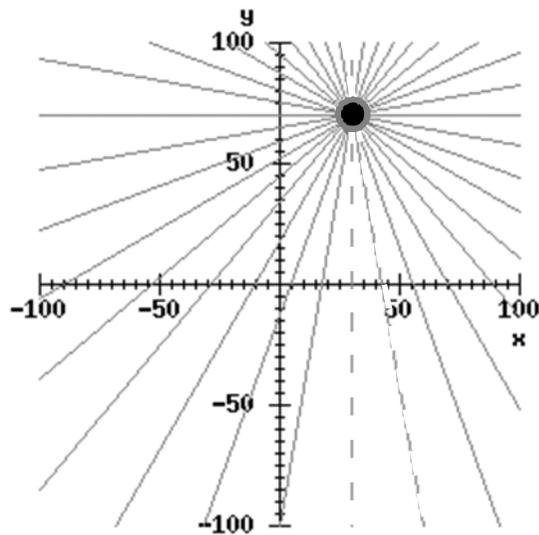
Прямую на плоскости можно задать следующим образом:
 $x \cos\theta + y \sin\theta = R,$



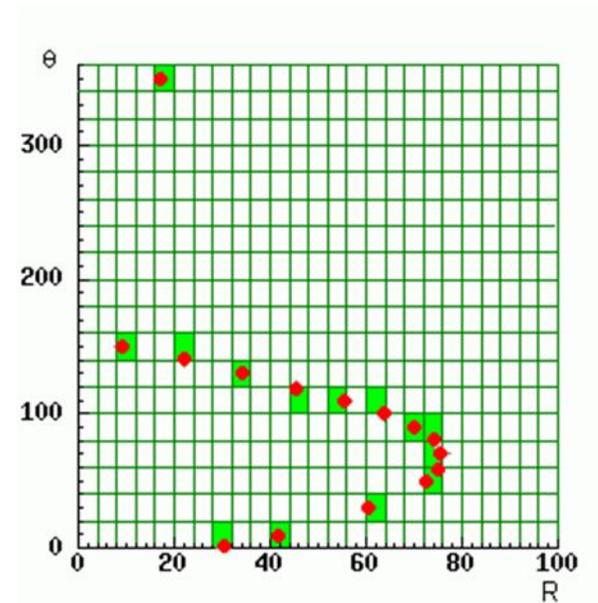
- R - длина перпендикуляра опущенного на прямую из начала координат
 - R ограничено размерами входного изображения.
- θ - угол между перпендикуляром к прямой и осью OX
 - θ изменяется в пределах от 0 до 2π



Фазовое пространство



Через одну точку можно провести несколько прямых. Учитывая дискретность их будет конечное число.



Каждой прямой пространства (x, y) соответствует точка фазового пространства (R, θ) . Прямые с левого рисунка образуют синусоиду.



Дискретизация

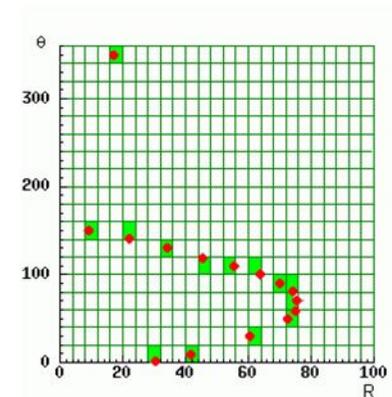
Переводим непрерывное фазовое пространство в дискретное:

- введем сетку на пространстве (R, θ)

Счетчик ставится в соответствие каждой ячейке сетки $[R_i, R_{i+1}] \times [\theta_i, \theta_{i+1}]$

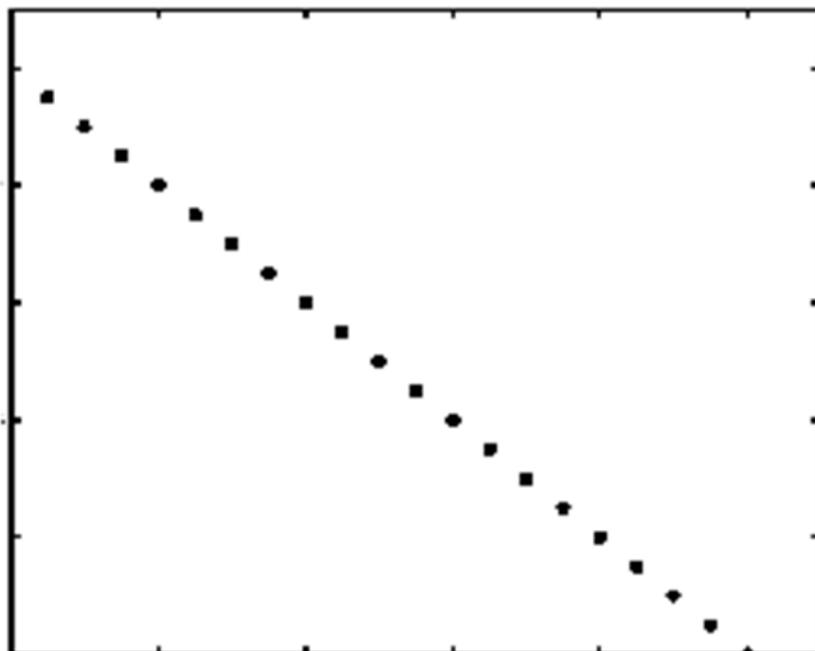
За эту ячейку «голосуют» точки (x, y) , удовлетворяющие:

$$x \cos \theta + y \sin \theta = R, \text{ где } \theta_i \leq \theta \leq \theta_{i+1}, R_i \leq R \leq R_{i+1}$$

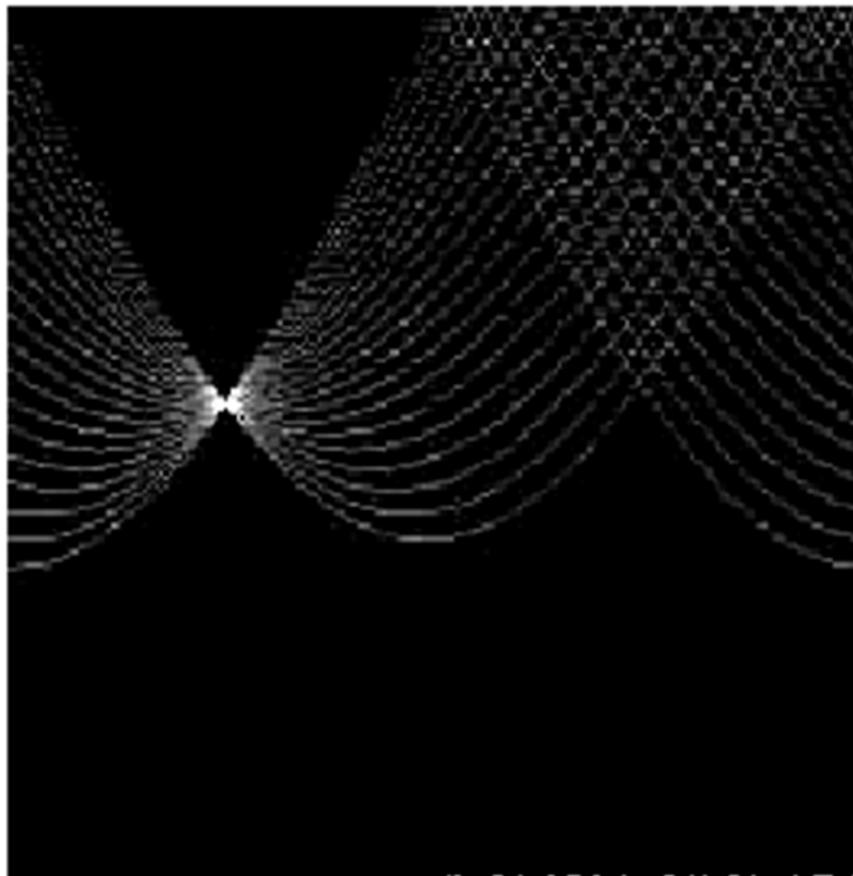




Иллюстрация



features

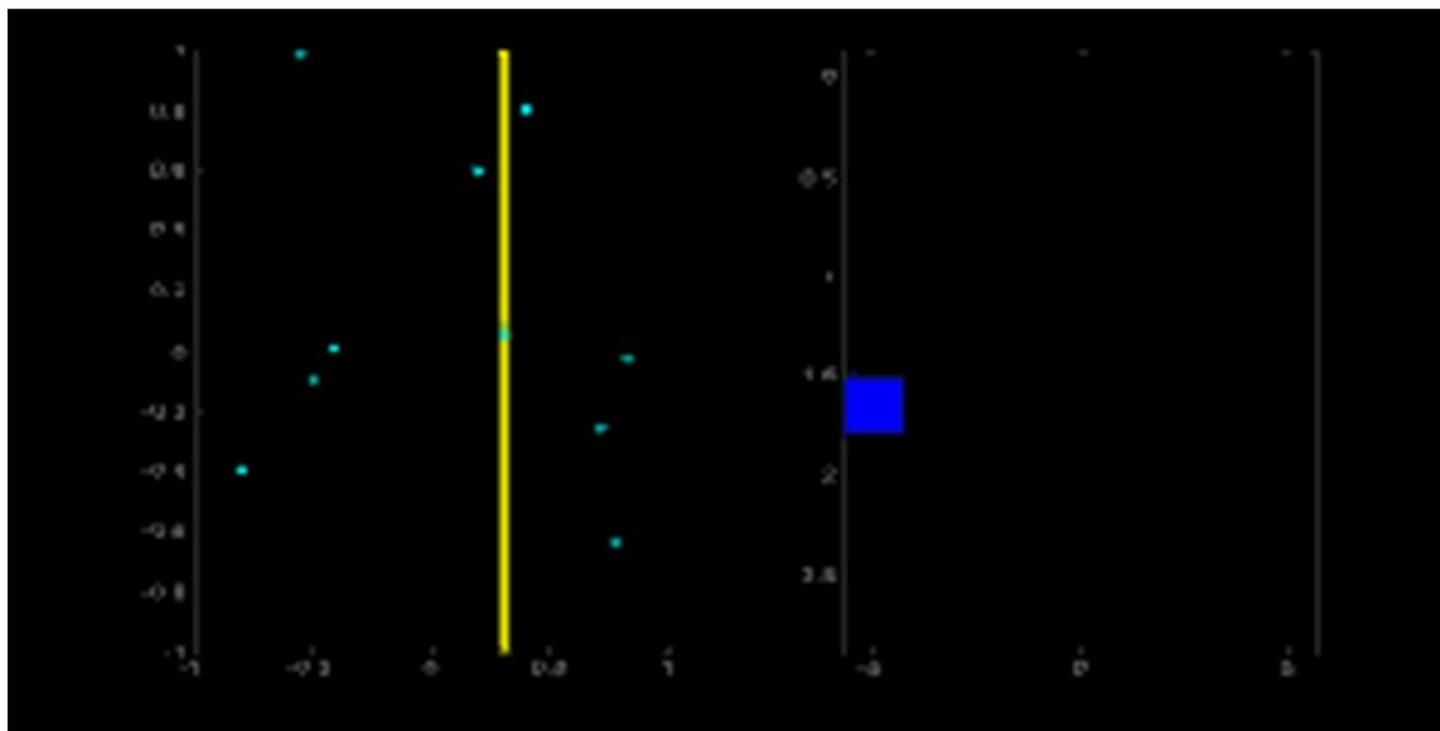


votes



Демонстрация

Как заполняется аккумулятор



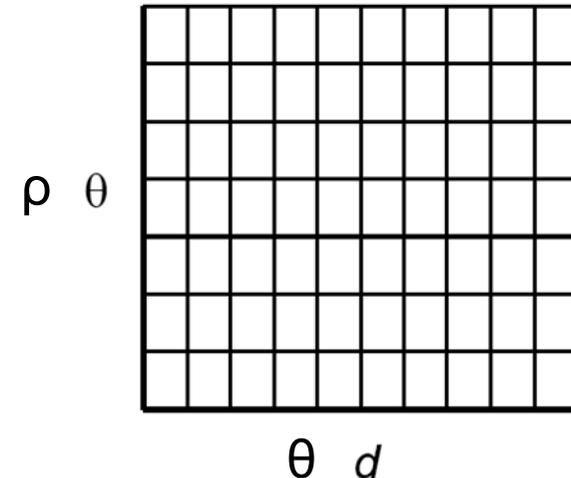
Анимация со страницы Amos Storkey



Схема алгоритма

- Инициализируем аккумулятор H нулями
- Для каждой точки края (x, y) на изображении
 - For $\theta = 0$ to 180
 - $\rho = x \cos \theta + y \sin \theta$
 - $H(\theta, \rho) = H(\theta, \rho) + 1$
 - end
- end
- Найти значения (θ, ρ) где $H(\theta, \rho)$ достигает локального максимума
 - Найденные линии задаются формулой
 - $\rho = x \cos \theta + y \sin \theta$

H: accumulator array (votes)

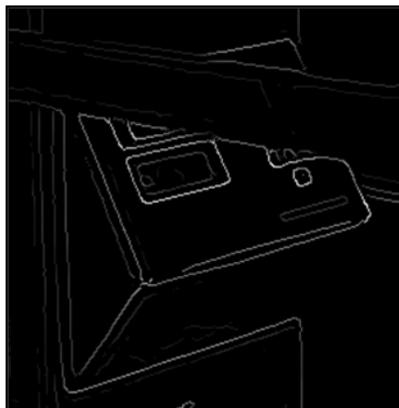




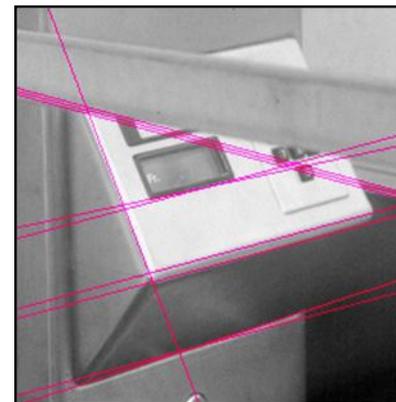
Пример работы метода



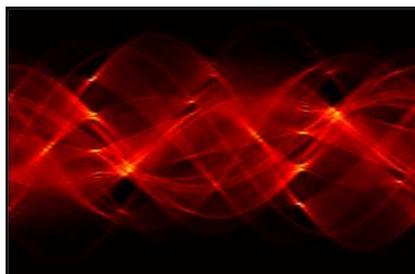
Исходное изображение



Выделенные края



Найденные прямые

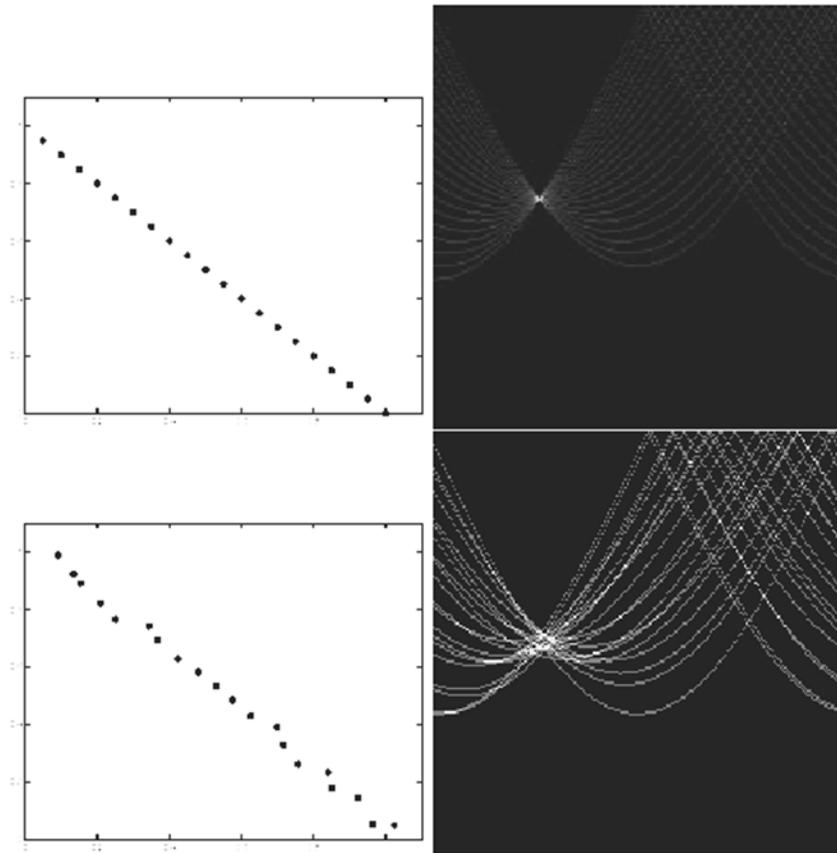


Фазовое пространство



Недостатки метода Хафа

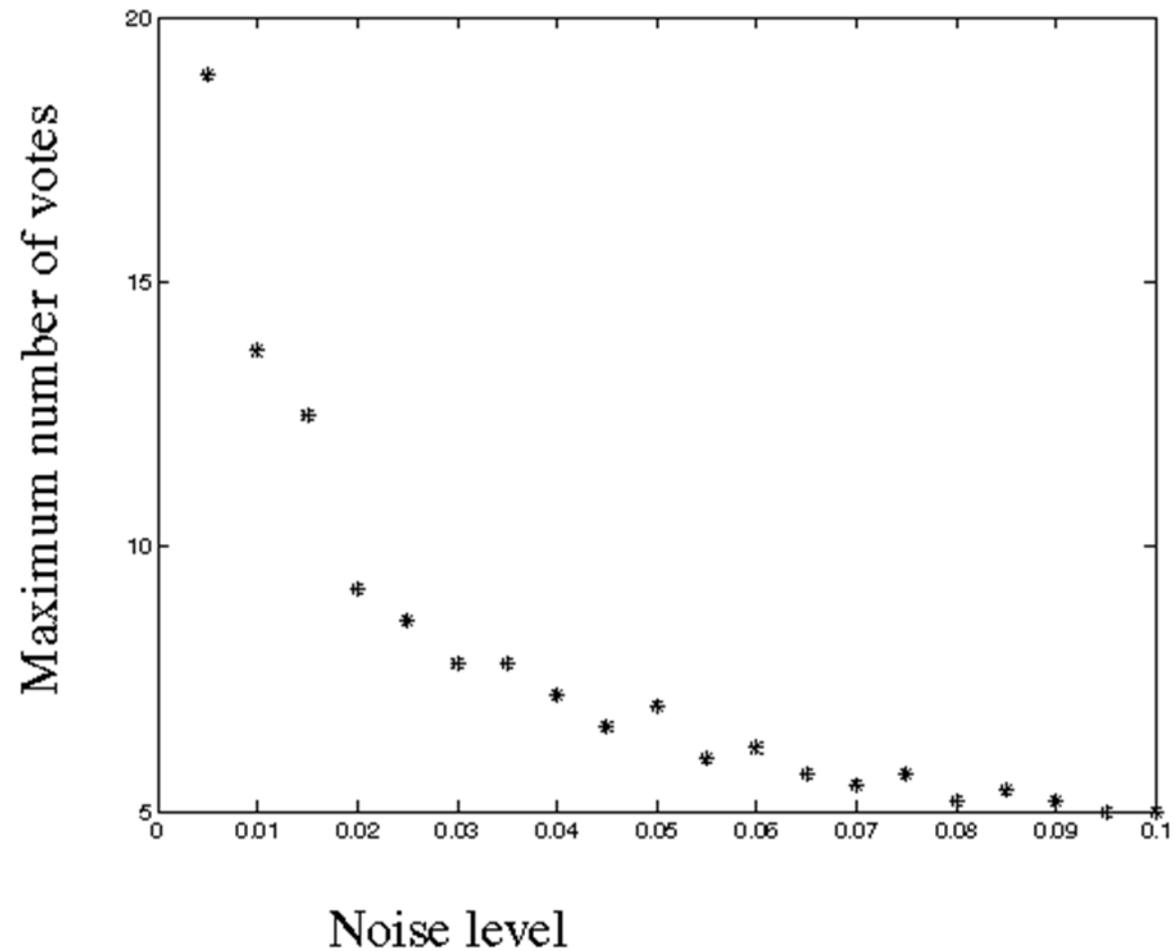
- Ручной выбор дискретизации фазового пространства
 - Шум приводит к «размытию» максимумов





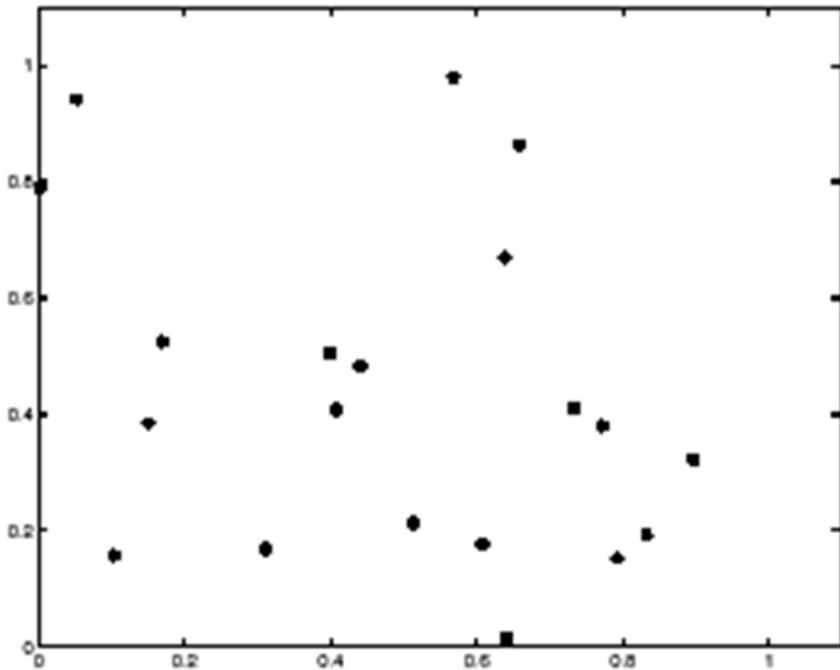
Эффект шума

- Количество голосов за линию из 20 точек с увеличением шума:

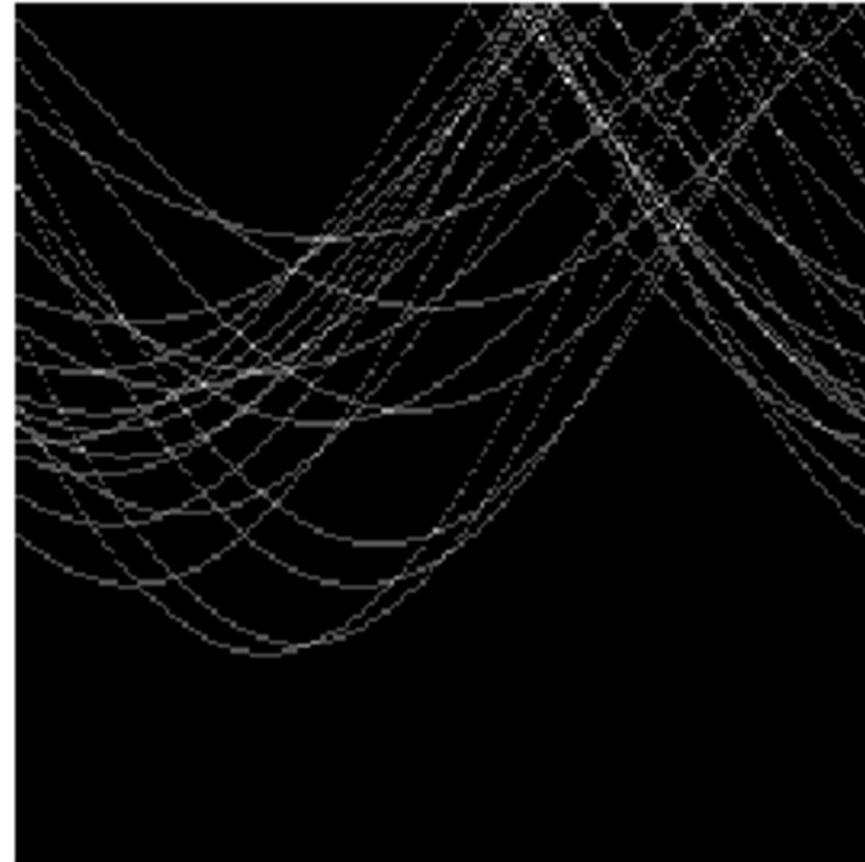




Случайные точки (выбросы)



features

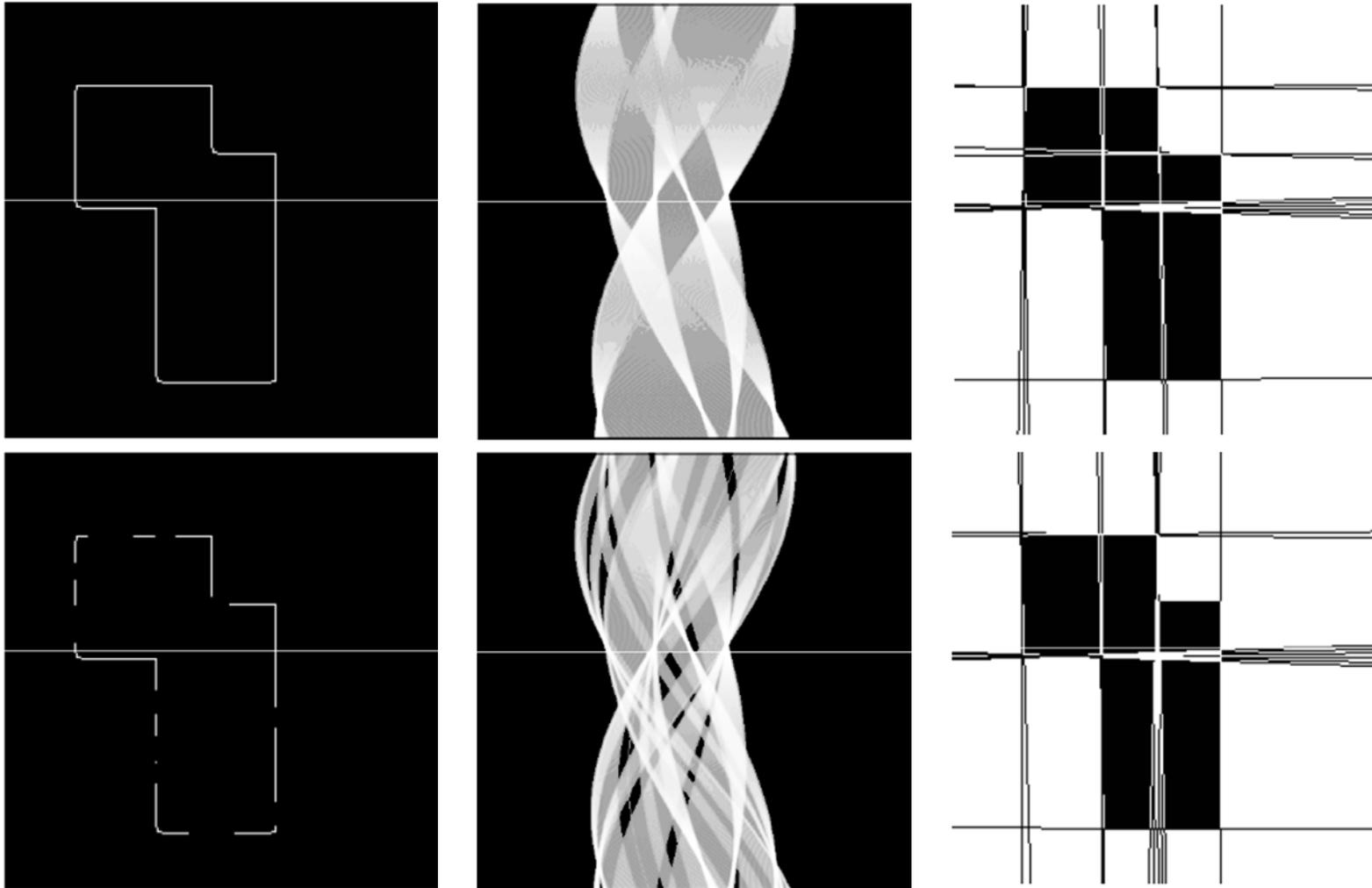


votes

Равномерно распределенные точки могут приводить к случайным пикам в аккумуляторе



Неполнота данных



Пропущенные данные также приводят к размытию значений в аккумуляторе



Преобразование Радона

Преобразование Хафа (Hough transform)

- Перебрать все точки на изображении
- Для каждой выбрать все прямые, через нее проходящие
 - Увеличить счетчик точек для каждой из этих прямых

Преобразование Радона (Radon transform)

- Перебрать все возможные прямые на изображении
- Для каждой посчитать кол-во точек, на ней лежащих

Доказано, что эти подходы эквивалентны в дискретном случае



Преобразование Радона

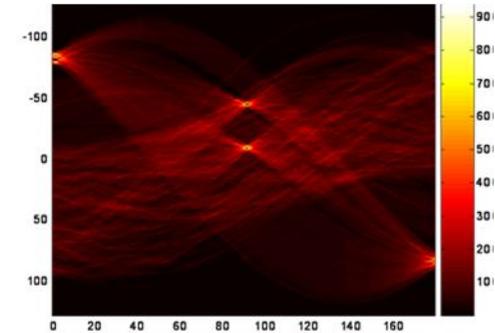
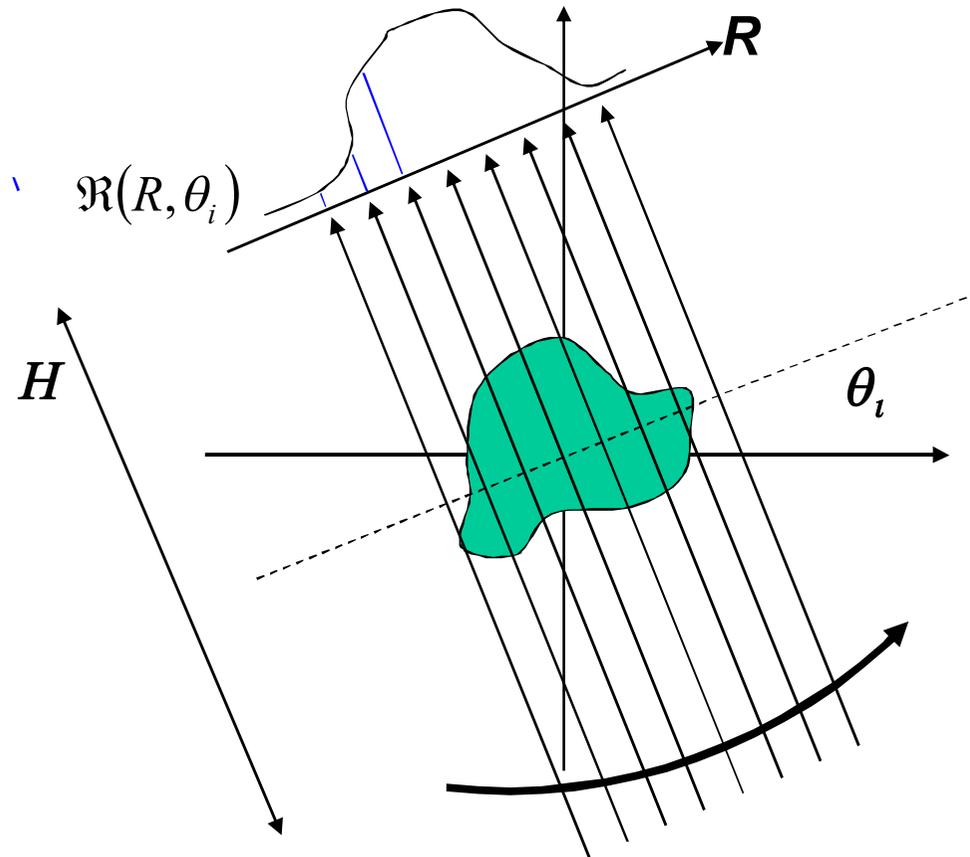
Используется та же параметризация, что и для Хафа

- R - определяется размерами изображения,
 $0 \leq \theta \leq 2\pi$
- Рассчитывается следующая величина:

$$\mathfrak{R}(R, \theta) = \int_H I(R \cos \theta - s \sin \theta, R \sin \theta + s \cos \theta) ds$$



Преобразование Радона

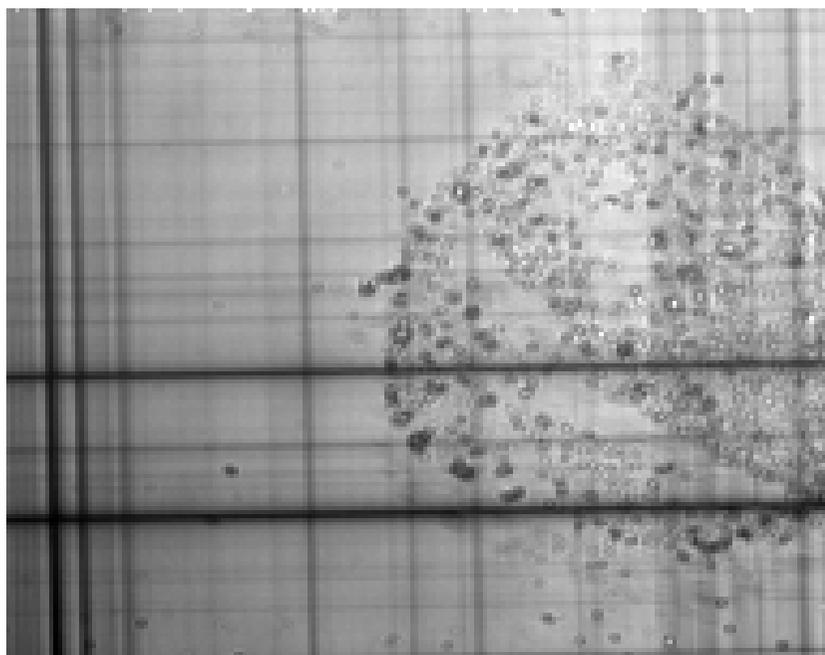


$$\mathfrak{R}(R, \theta) = \int_H I(R \cos \theta - s \sin \theta, R \sin \theta + s \cos \theta) ds$$

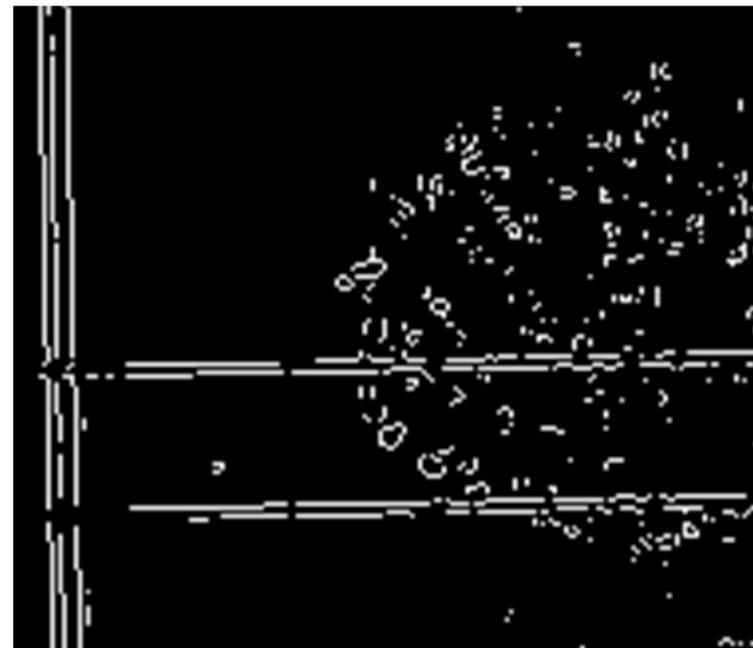


Пример поиска прямых

Пример взят из помощи к MatLab



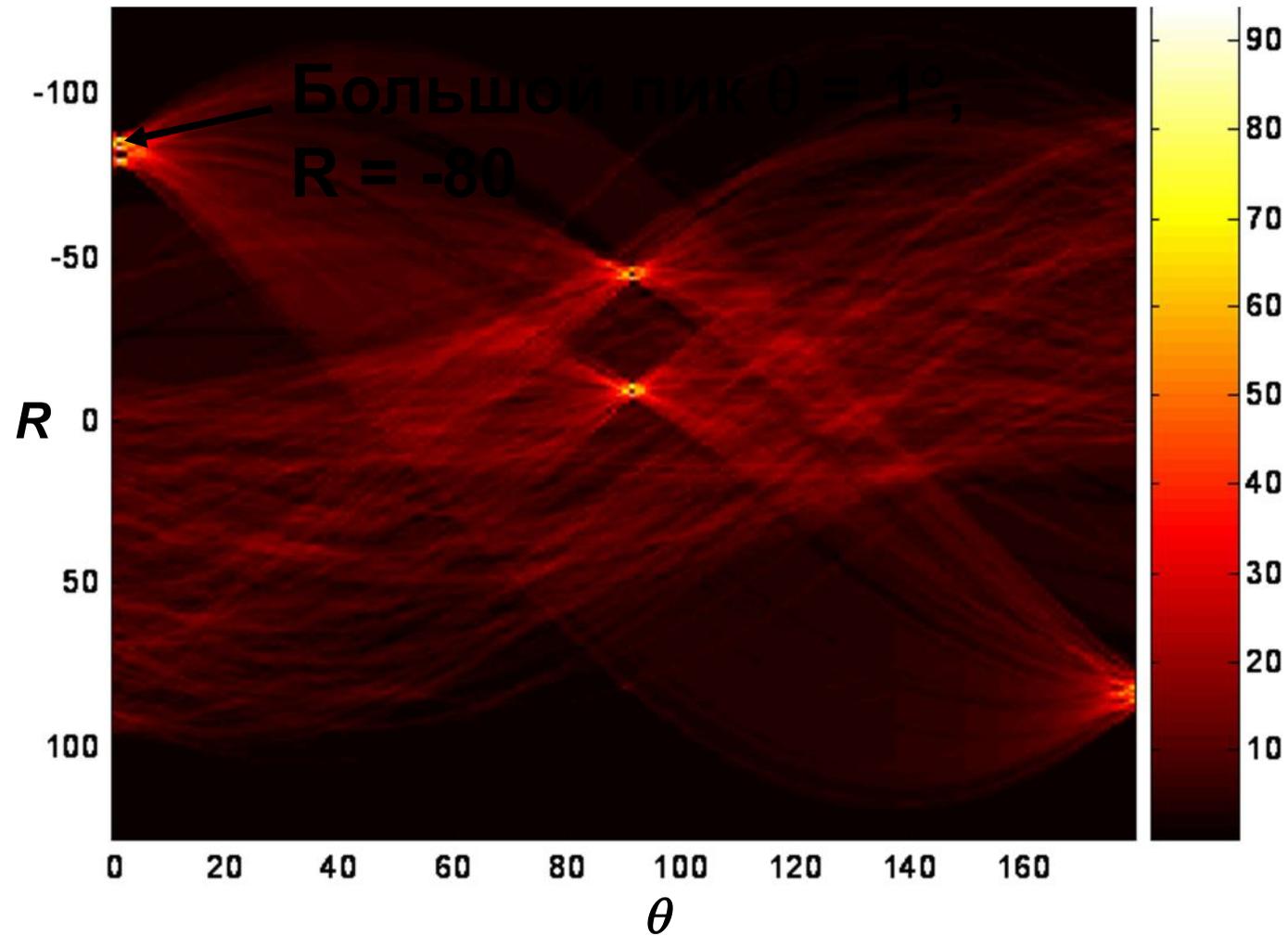
**Исходное
изображение**



**Найденны
е края**

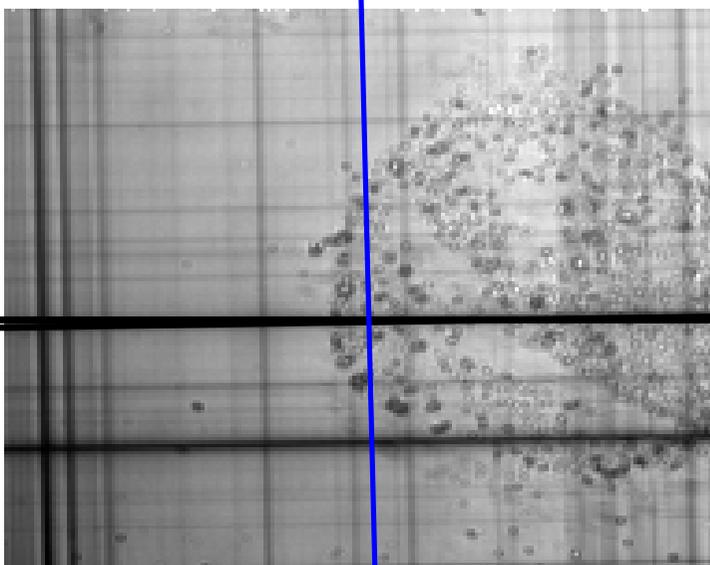


Пример фазового пространства



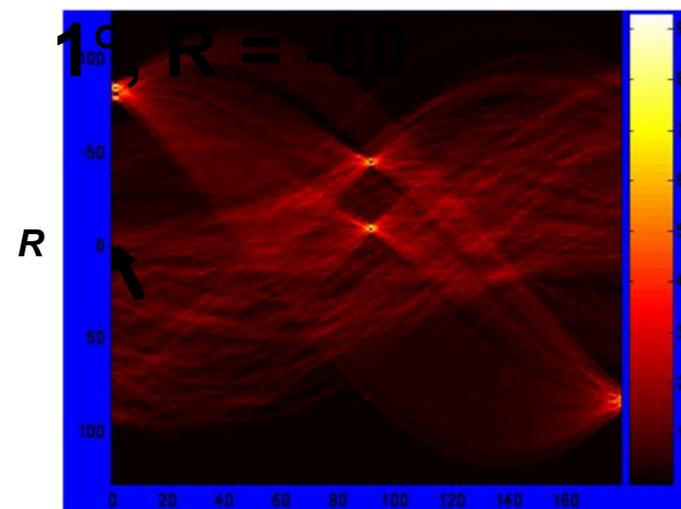


Соответствующая прямая



Исходное
изображение

Большой пик $\theta =$



θ
 $\mathfrak{R}(R, \theta_i)$



Радон и Хаф

Доказано, что эти преобразования эквивалентны для дискретного случая

- Преобразование Радона пришло из томографии, изначально формулируется непрерывно
- Преобразование Хафа придумано специально для поиска фигур на изображениях, дискретно изначально

Когда что лучше применять?

- Хаф эффективнее когда на изображении мало точек
- Радон эффективнее, когда возможных конфигураций прямых немного



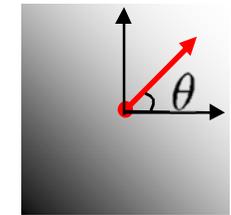
Практические вопросы

- Отфильтровать лишние признаки
 - Для линий, стоит брать точки на краях только с большим градиентом
- Выбор правильной сетки
 - Слишком грубая: несколько близких линий будут голосовать за одну ячейку
 - Слишком мелкая: можно пропустить линии, т.к. зашумленные точки будут голосовать за разные ячейки
- Для поиска максимумов можно сглаживать значения в аккумуляторе
- Какая точка соответствует какой линии?
 - Помечаем голоса



Расширение: учет градиента

- Когда мы находим края, мы знаем также градиент
- Это значит, что направление линии уже определено!
- Усовершенствованный метод:


$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$
$$\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$$

```
For each edge point (x,y)
  theta = gradient orientation at (x,y)
  rho = x cos theta + y sin theta
  H(theta, rho) = H(theta, rho) + 1
end
```



Обобщение на другие кривые

Рассмотрим семейство кривых на плоскости, заданное параметрическим уравнением:

$$F(a_1, a_2, \dots, a_n, x, y) = 0;$$

где F - некоторая функция, a_1, a_2, \dots, a_n - параметры семейства кривых, x, y - координаты на плоскости.

Параметры семейства кривых образуют фазовое пространство, каждая точка которого (конкретные значения параметров a_1, a_2, \dots, a_n) соответствует некоторой кривой.



Преобразование Хафа: За и Против

- За
 - Может работать в условиях перекрытий
 - Поиск нескольких экземпляров объектов за один проход
 - Некоторая устойчивость к шуму: выбросы обычно голосуют за разные ячейки
- Против
 - Сложность алгоритма резко растет с увеличением количества параметров
 - Сложно выбрать правильный шаг сетки



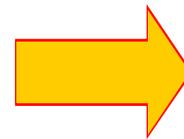
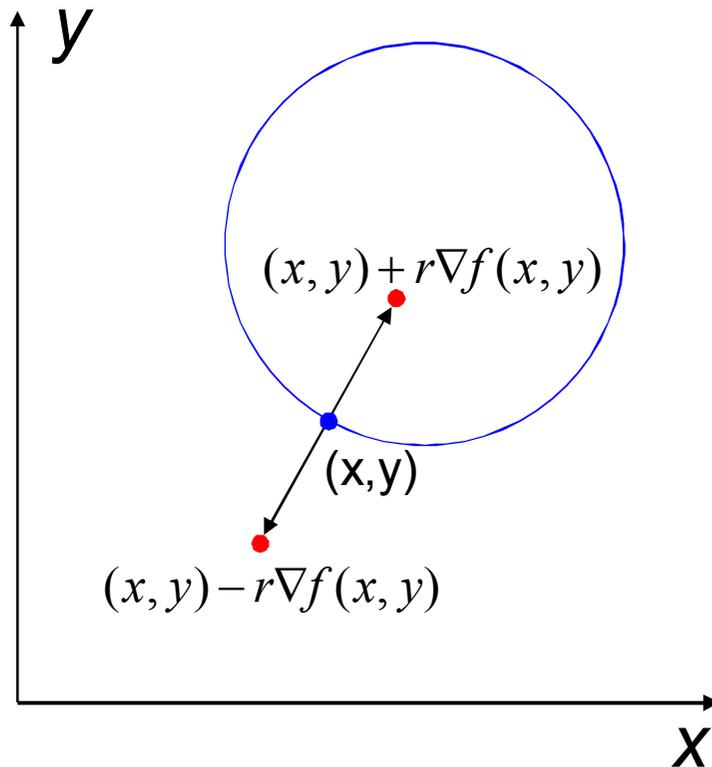
Поиск окружностей

- Как применить преобразование Хафа для окружностей?
- Размерность нашего фазового пространства?
- Если дана точка ориентированного края, за какие точки в фазовом пространстве она будет голосовать?

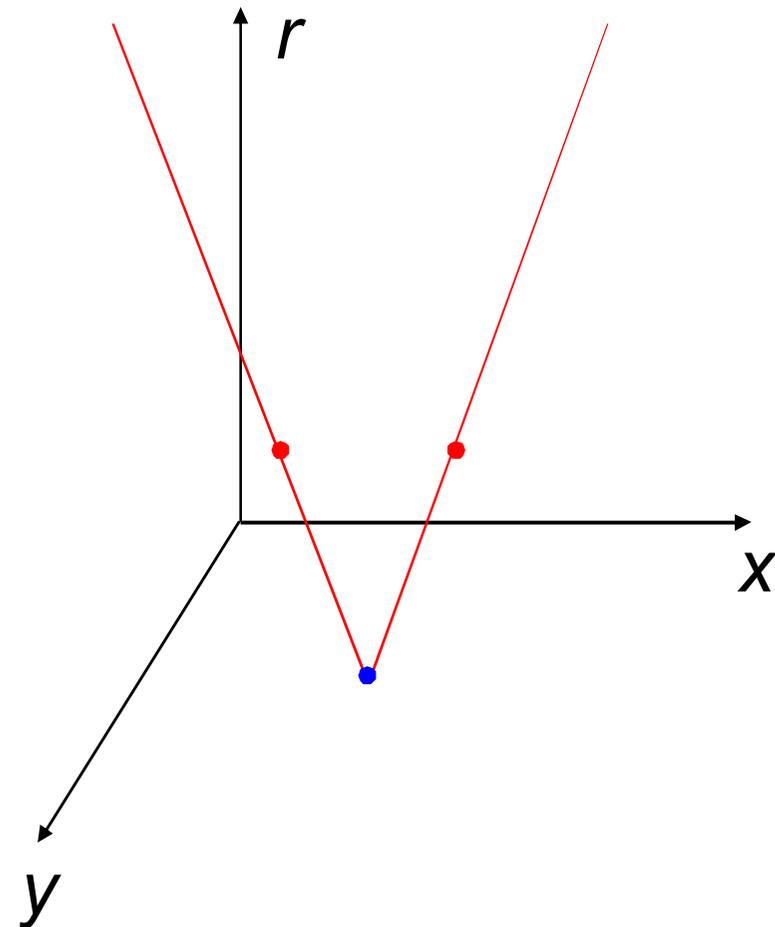


Поиск окружностей

image space



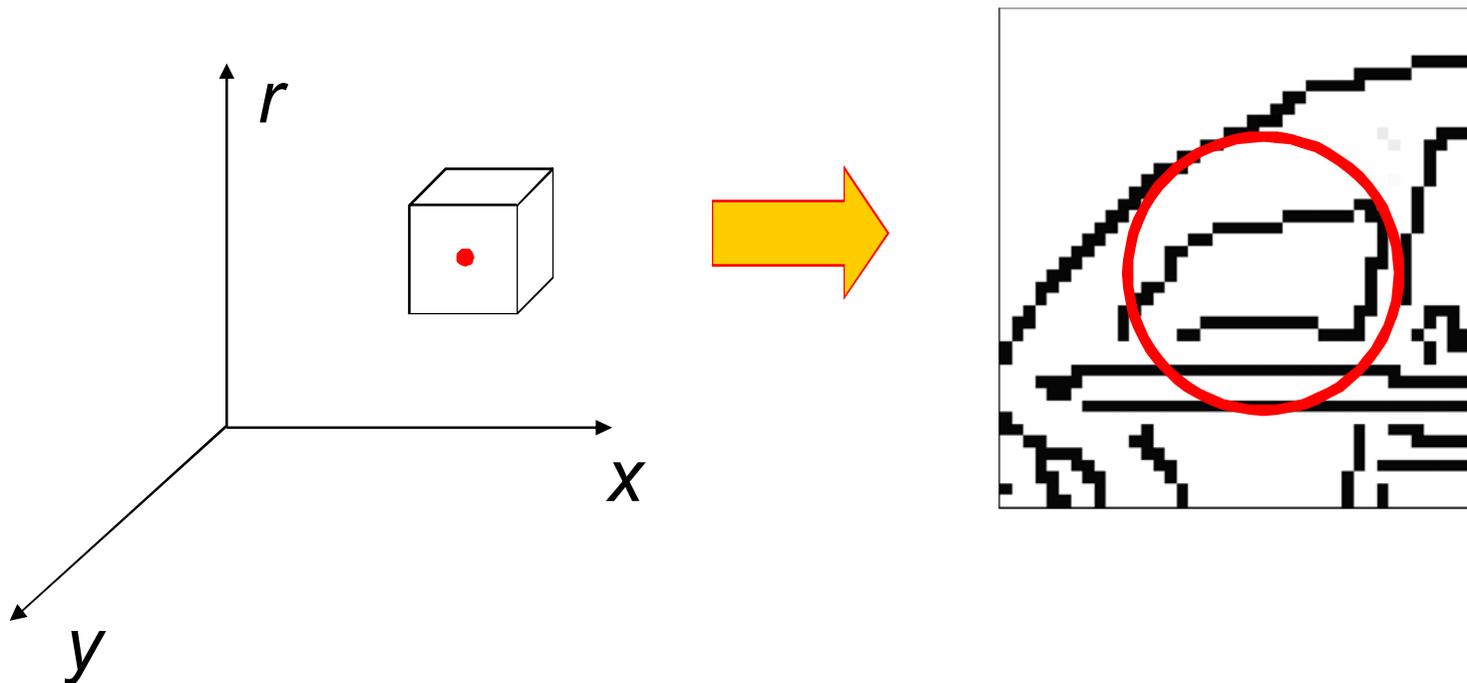
Hough parameter space





Поиск окружностей

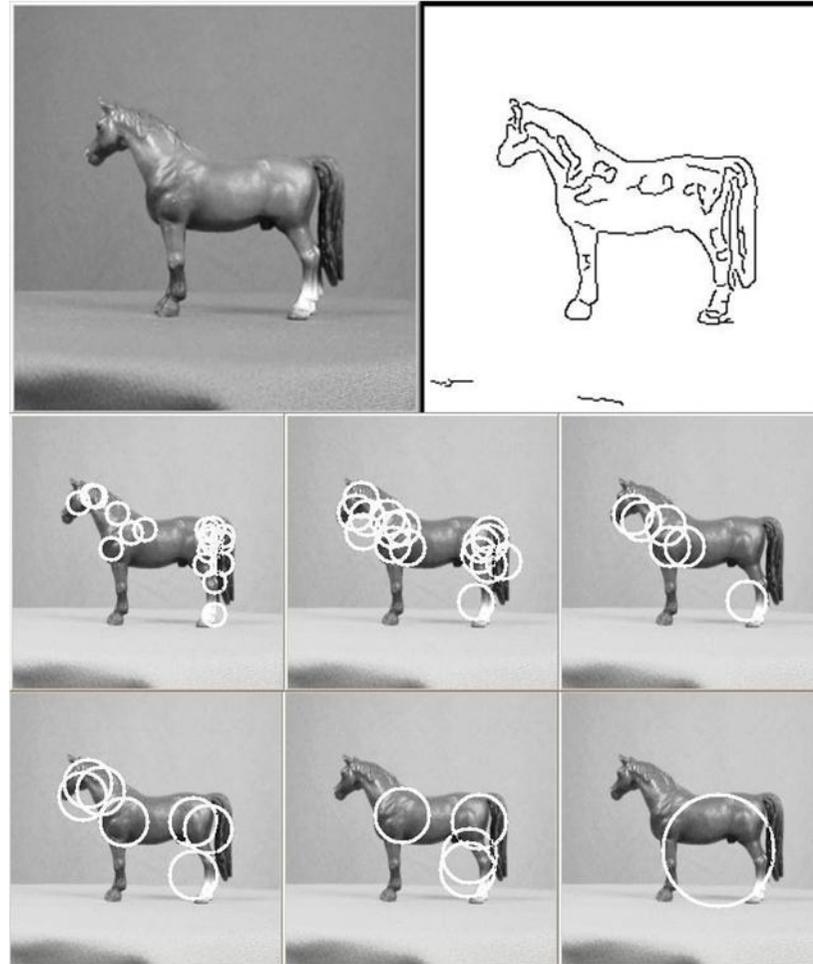
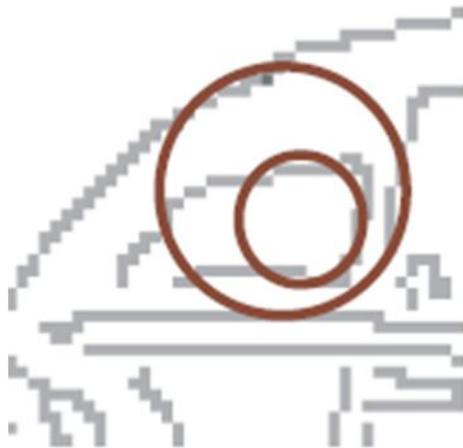
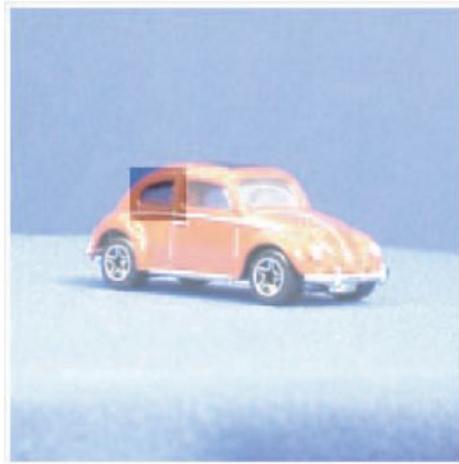
- Преобразование Радона для окружностей
- Для каждого (x, y, r) , рисуем окружность на изображении и вычисляем «поддержку»



Какое преобразование эффективнее?



Применение для распознавания

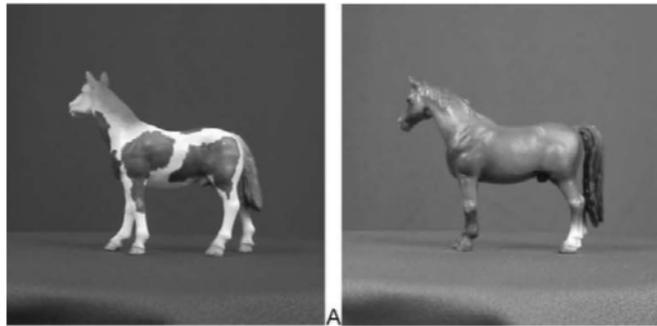


F. Jurie and C. Schmid, [Scale-invariant shape features for recognition of object categories](#), CVPR 2004

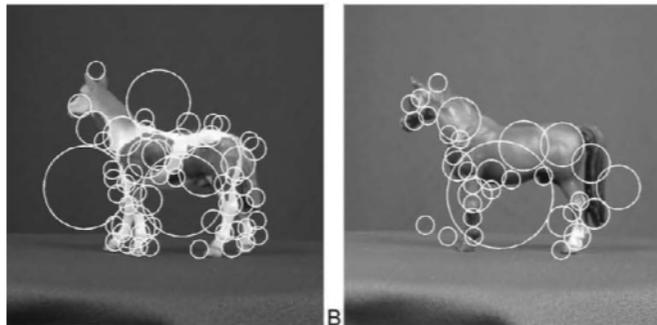


Hough circles vs. Laplacian blobs

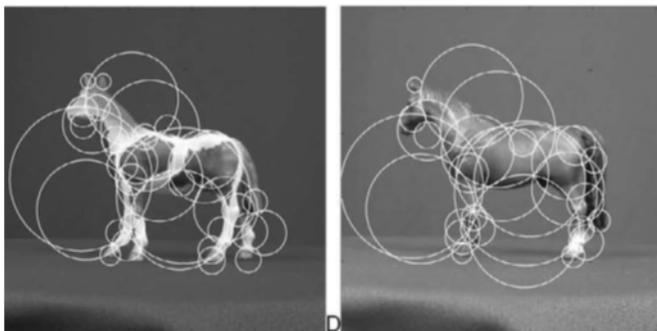
Original images



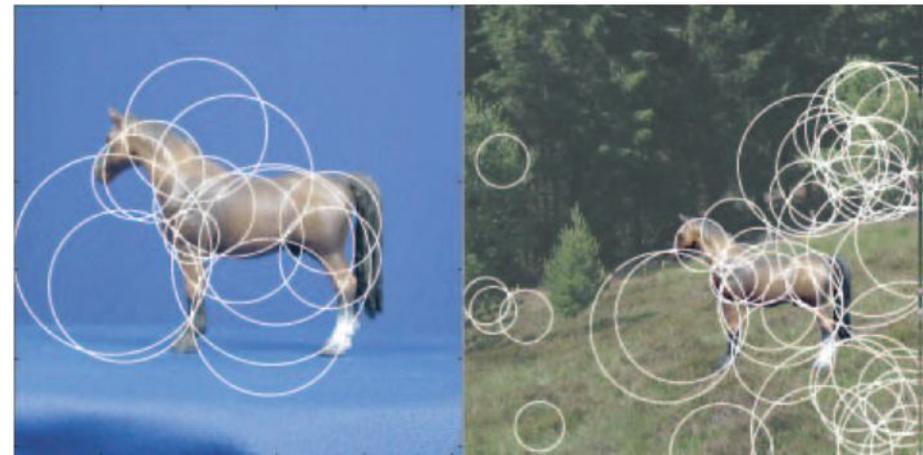
Laplacian circles



Hough-like circles



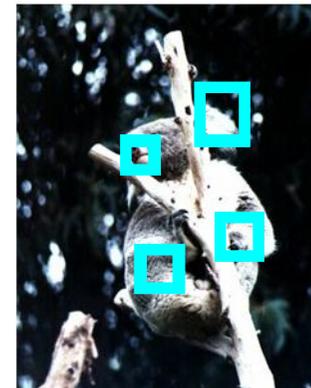
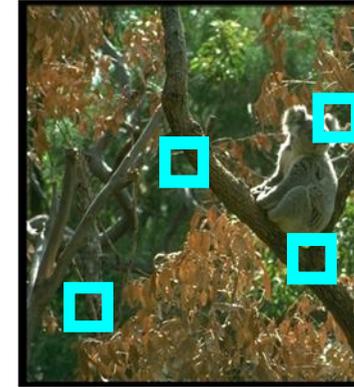
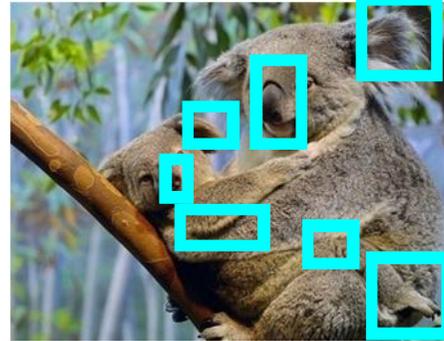
Устойчивость к шуму и фону



F. Jurie and C. Schmid, [Scale-invariant shape features for recognition of object categories](#), CVPR 2004



Применение: распознавание





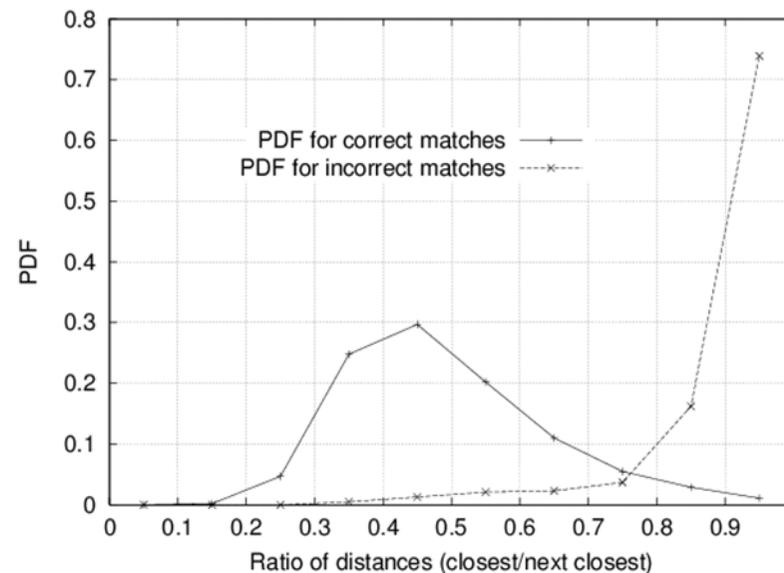
Сопоставление по SIFT

- Идея:
 - Возьмем несколько изображений объекта с разных ракурсов
 - Найдем на нём особые точки, вычислим дескриптор
 - То же самое сделаем с тестовым изображением
 - Будем сопоставлять все изображения из базы с тестовым изображением
 - Ищем «группы» совпадающих особенностей
 - Достаточно хорошо работает при 3х особенностях



Шаг 1

- Найдем ближайшие по дескриптору пары (м.б. тысячи)
- Отфильтруем ложные по эвристике:
 - Сравним расстояния до ближайшей особенности с второй ближайшей по метрики
 - Отношения будет большим для не очень «выделенных» особенностей
 - Порог 0.8 дает неплохое разделение



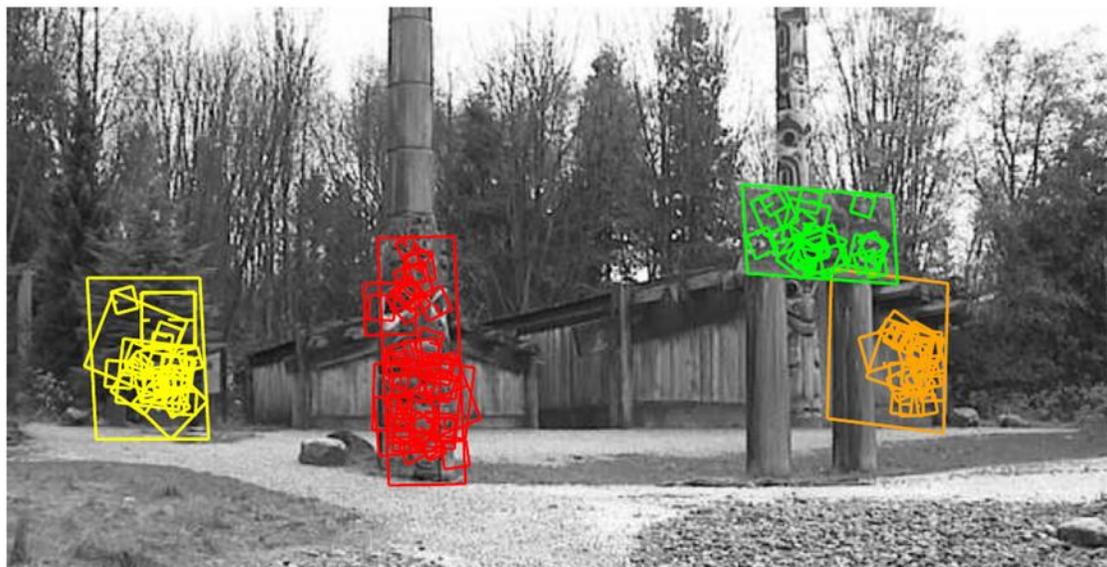


Шаг 2

- Используем схему голосования (Хаф)
 - Каждое соответствие голосует за преобразование
 - X , Y , Масштаб, ориентация
 - Находим ячейки не менее чем с $3m$ голосами
- Уточняем преобразование
 - Считаем аффинное преобразование по $3m$ соответствиям
 - Используем для поиска соответствий по другим точкам, фильтруем ложные, итеративно уточняем преобразование
- Классифицируем объект
 - Если не менее $3x$ соответствий
 - Используем Байесовский классификатор



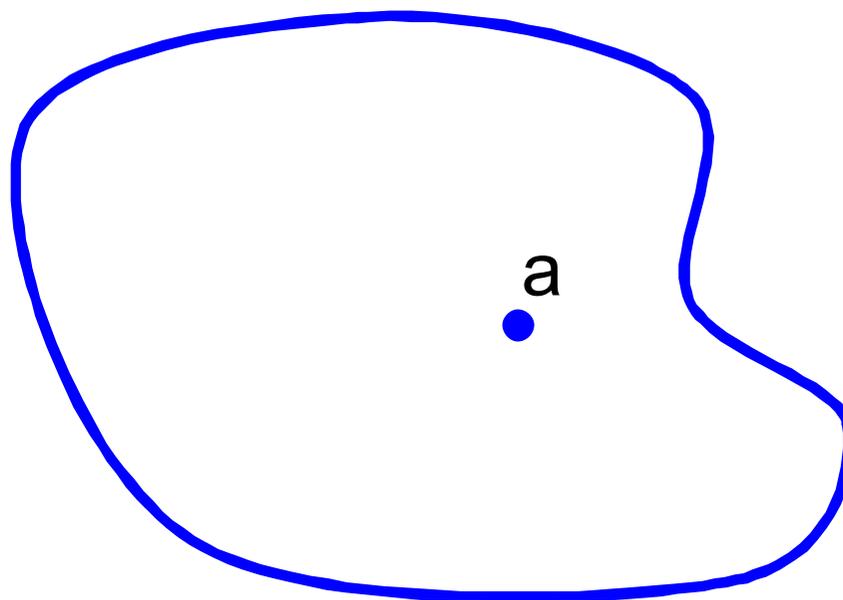
Пример





Обобщенное преобразование Хафа

- Мы хотим найти форму, заданную своим контуром и центром

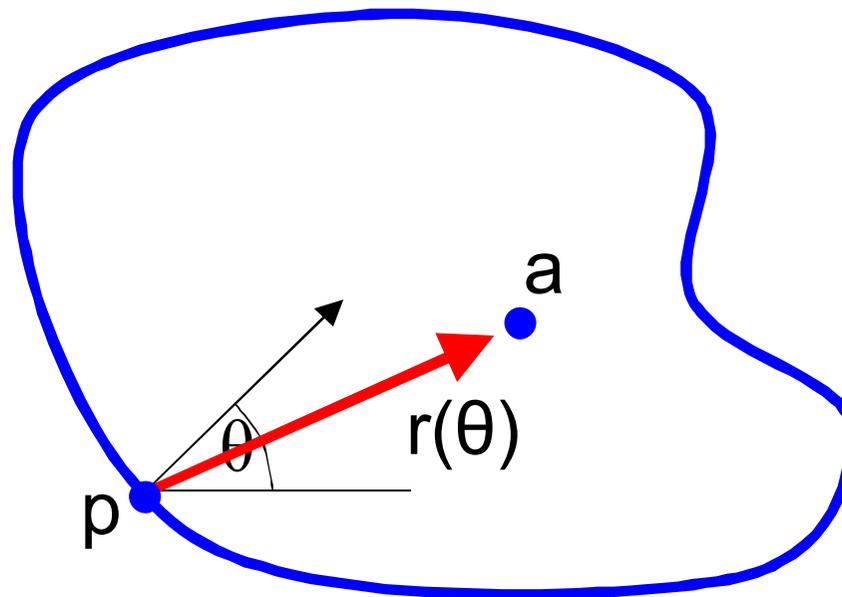


D. Ballard, [Generalizing the Hough Transform to Detect Arbitrary Shapes](#),
Pattern Recognition 13(2), 1981, pp. 111-122.



Обобщенное преобразование Хафа

- Для каждой точки контура p , вычислим вектор смещения $r = a - p$ как функцию ориентации градиента θ



D. Ballard, [Generalizing the Hough Transform to Detect Arbitrary Shapes](#),
Pattern Recognition 13(2), 1981, pp. 111-122.

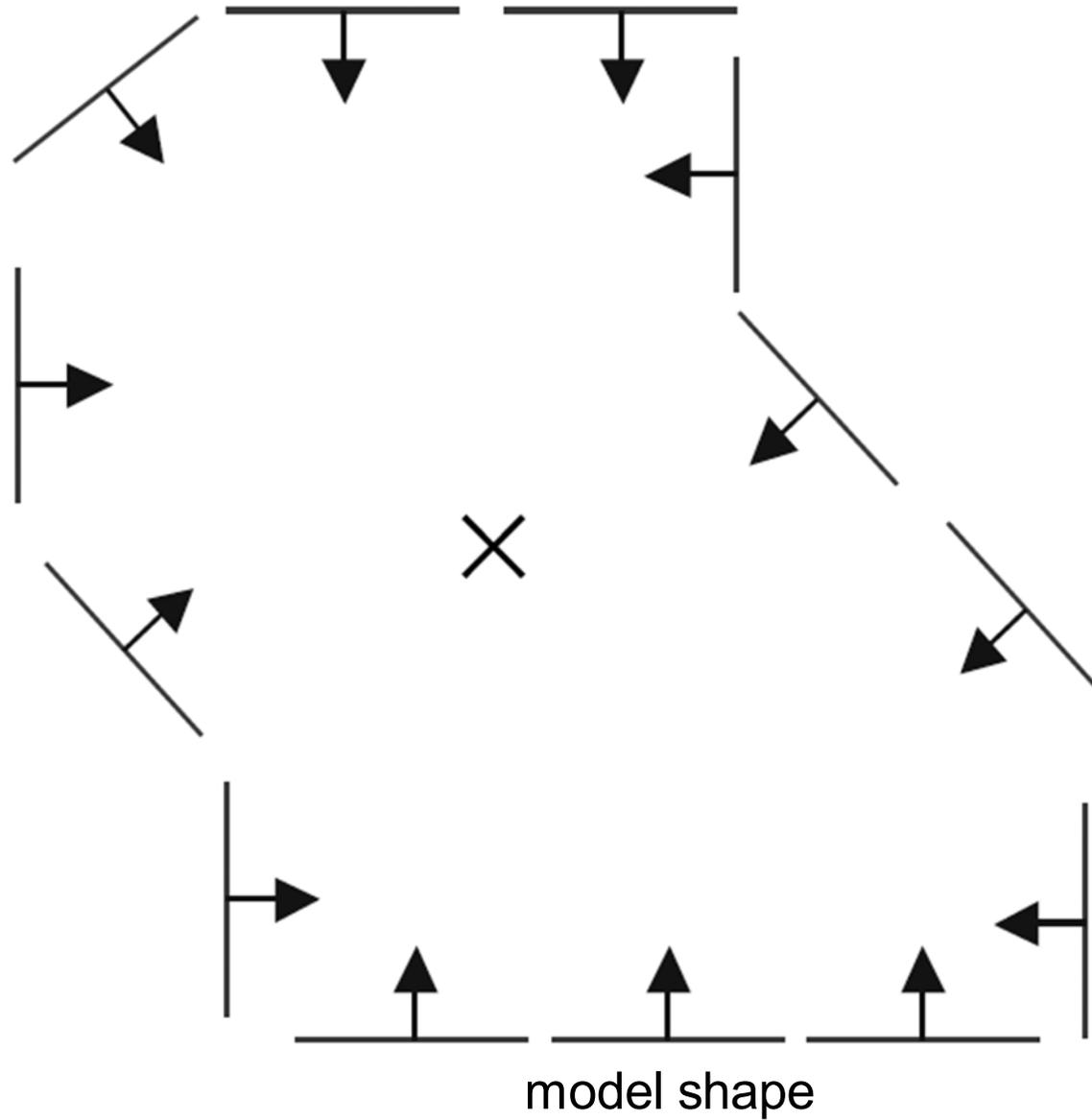


Обобщенное преобразование Хафа

- Для модели формы: построим таблицу, индексированную по θ , хранящую вектора смещения r как функцию от ориентации градиента
- Поиск: Для каждой точки края p с ориентацией градиента θ :
 - Извлечь все r индексированные по θ
 - Для каждого $r(\theta)$, проголосовать в аккумуляторе за точку $p + r(\theta)$
- Пики в фазовом пространстве соответствуют опорным точкам с наибольшей поддержкой
- Используется предположение о фиксированном размере и ориентации!

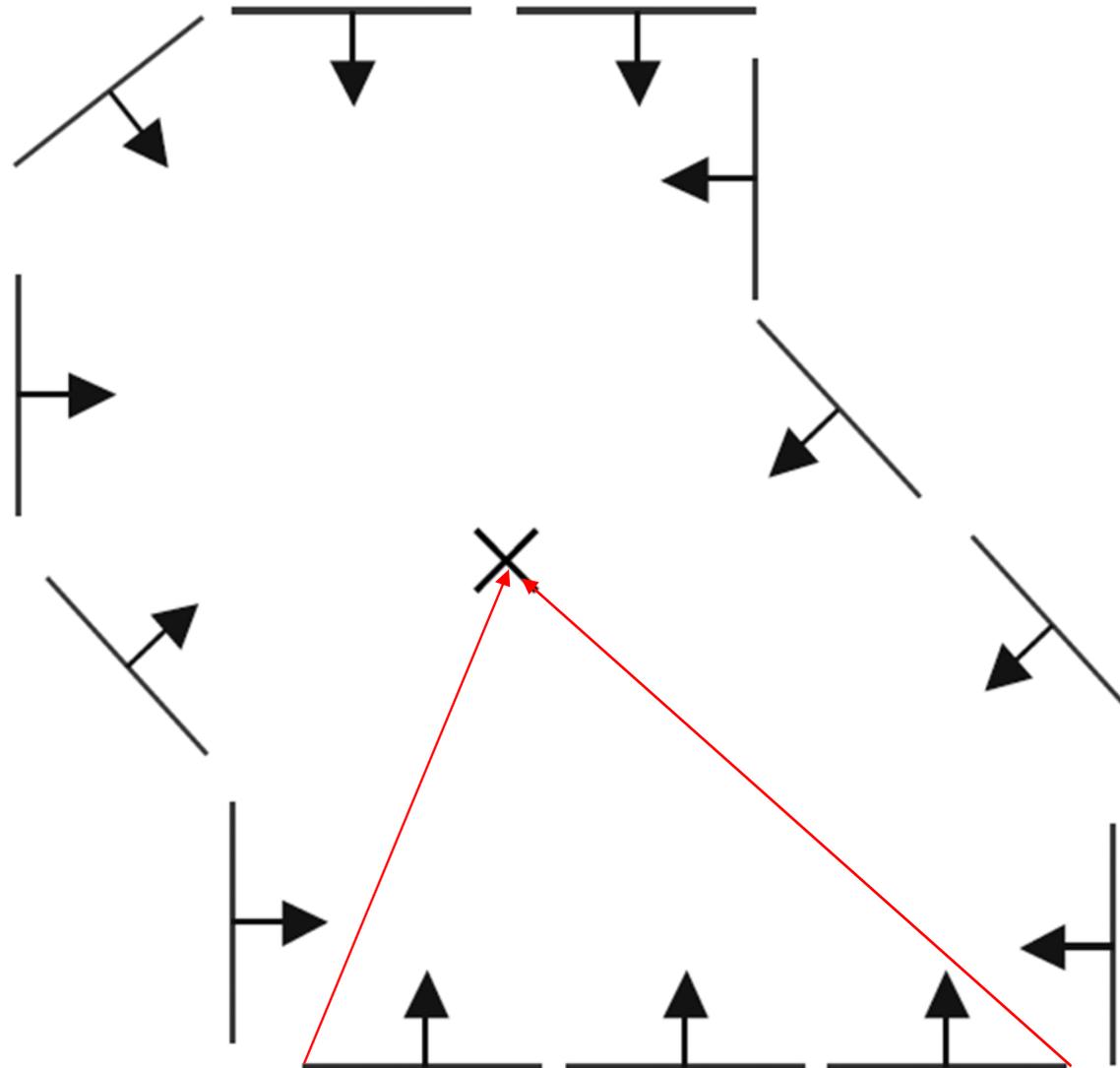


Пример





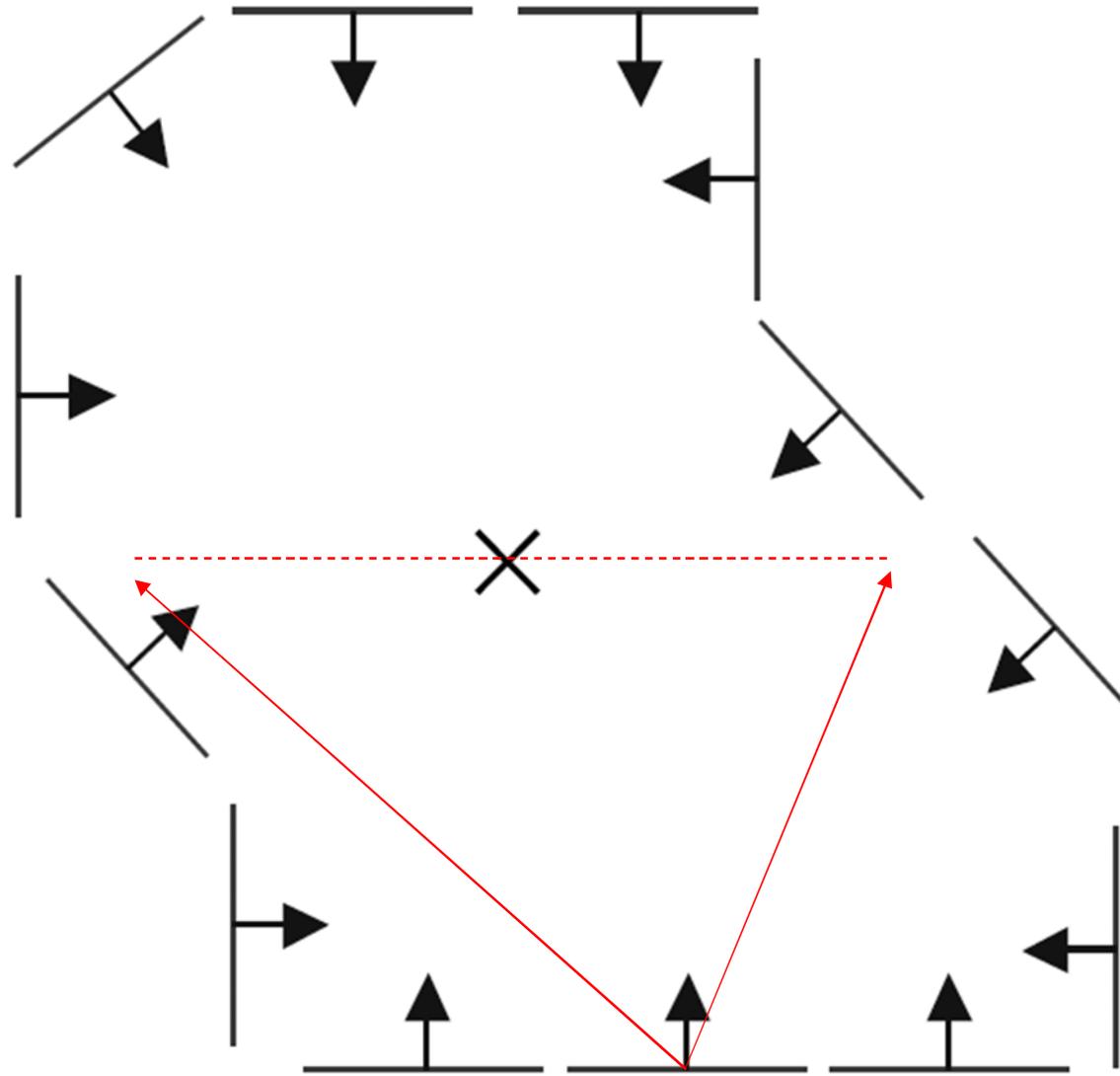
Пример



displacement vectors for model points



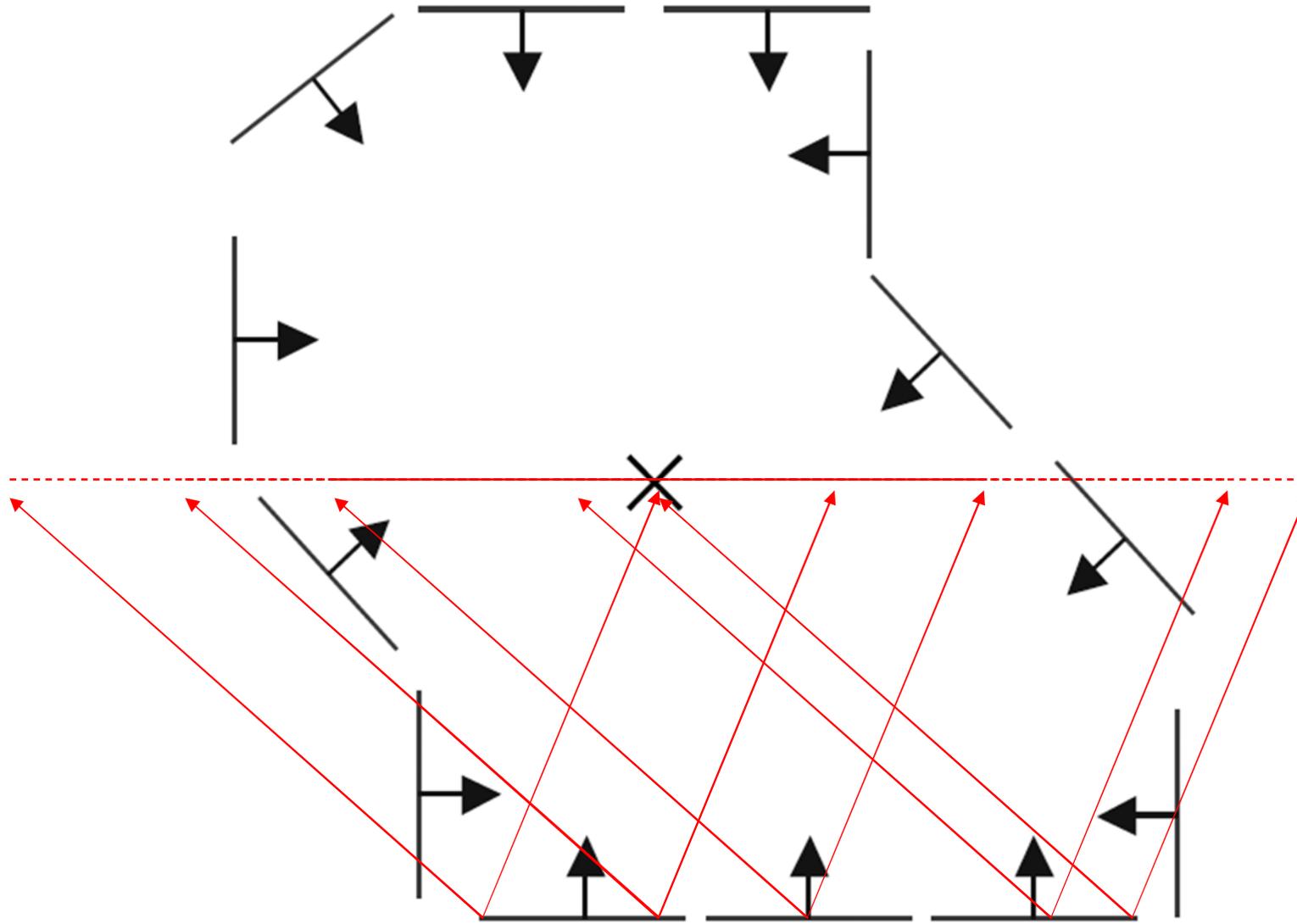
Пример



range of voting locations for test point



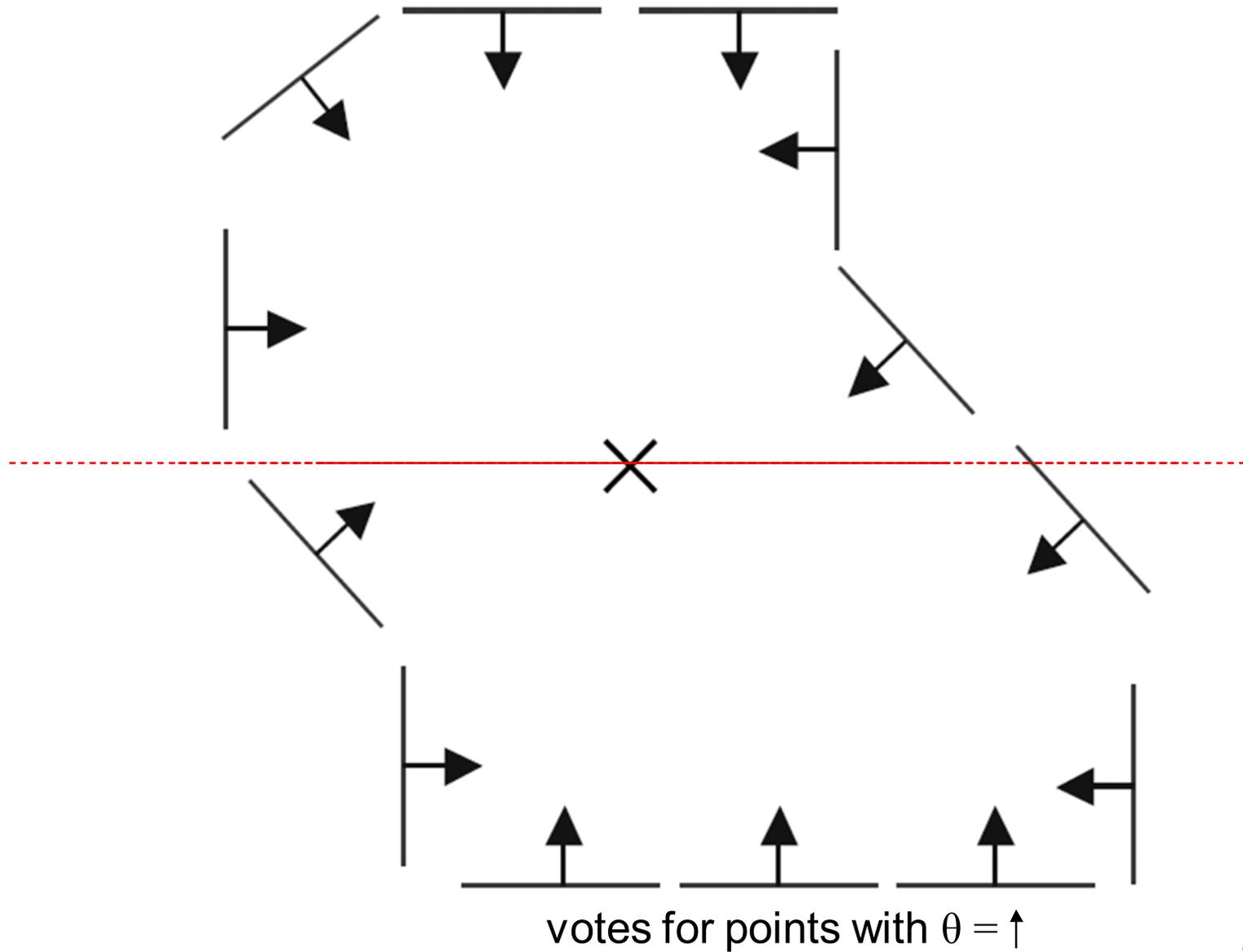
Пример



range of voting locations for test point

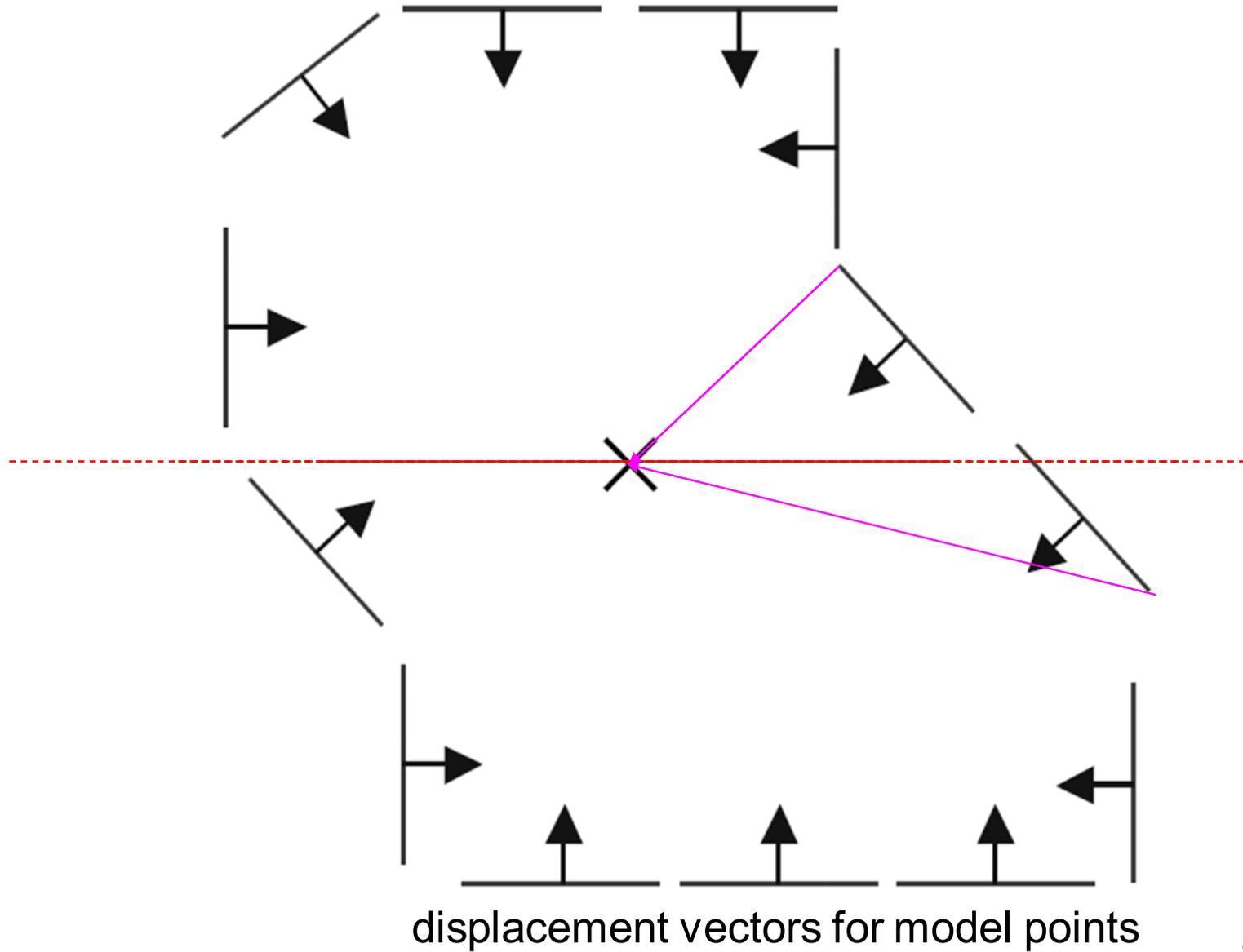


Пример



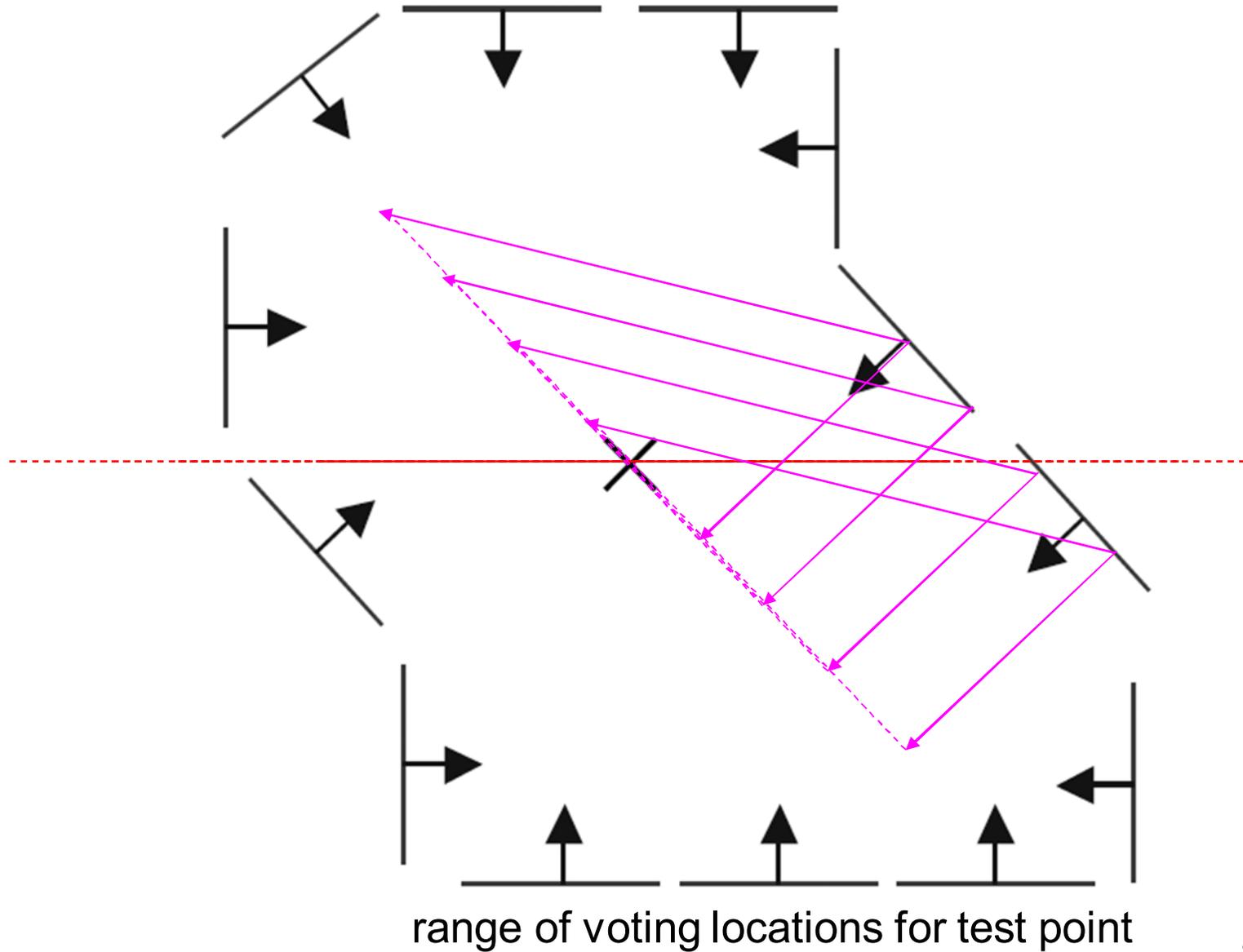


Пример



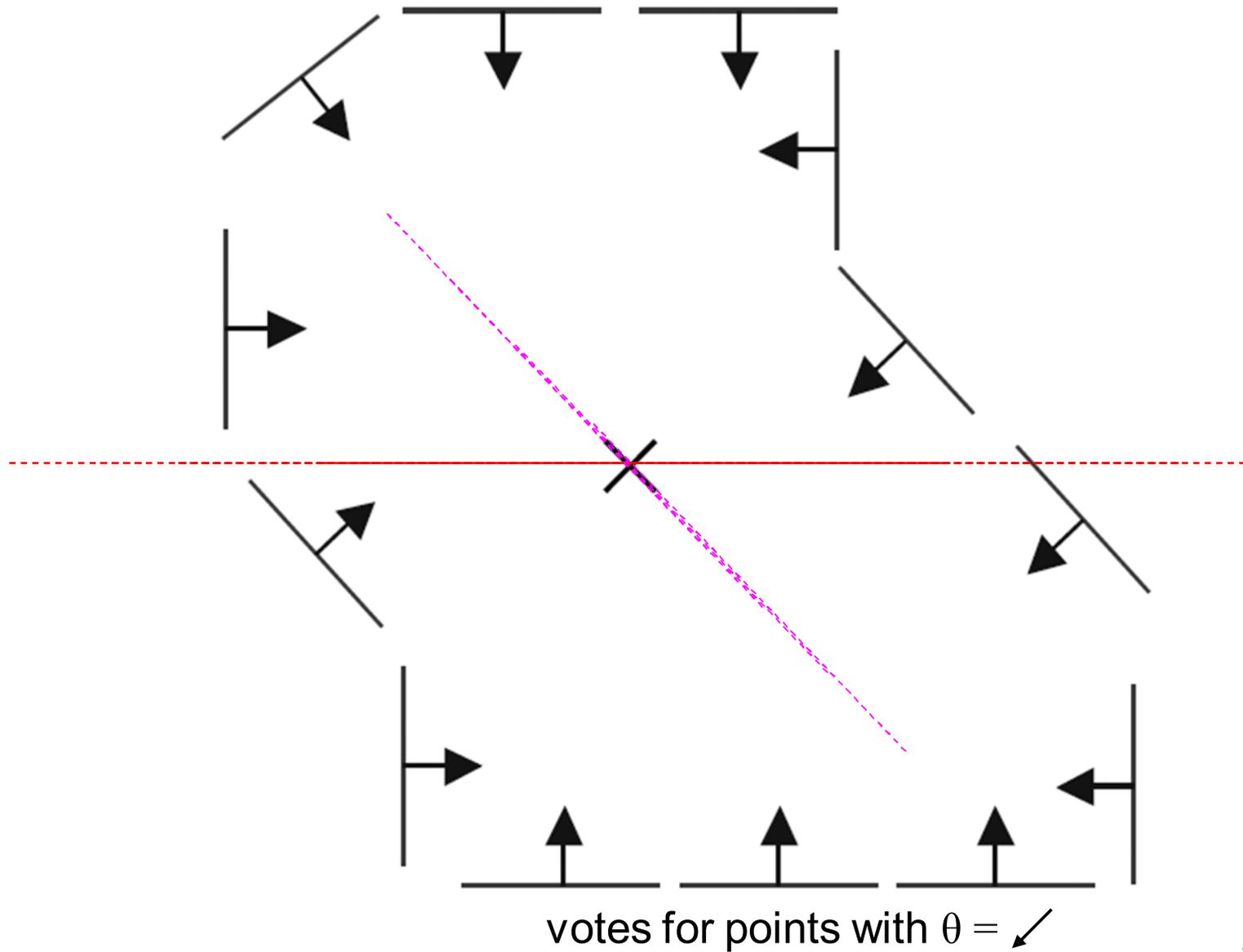


Пример





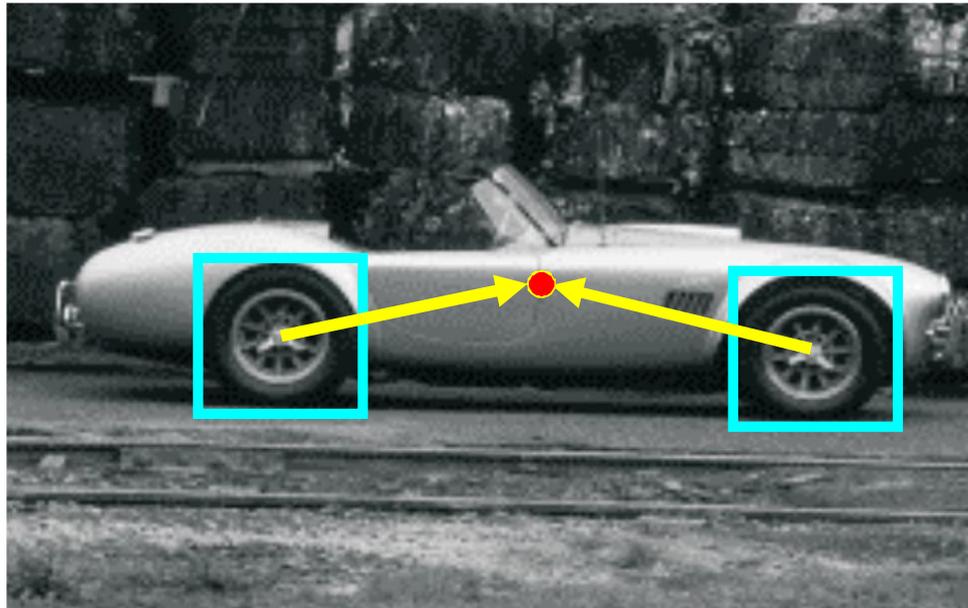
Пример



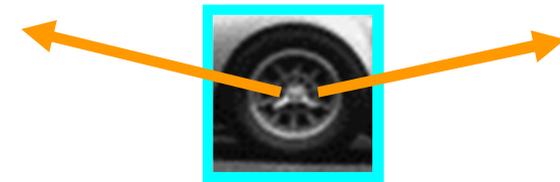


Применение в распознавании

- Вместо индексирования смещений по ориентации градиента индексируем по «кодовым словам»



Тренировочное изображение



Кодовое слово
с векторами смещения

B. Leibe, A. Leonardis, and B. Schiele, [Combined Object Categorization and Segmentation with an Implicit Shape Model](#), ECCV Workshop on Statistical Learning in Computer Vision 2004



Применение в распознавании

- Вместо индексирования смещений по ориентации градиента индексируем по фрагментам изображения



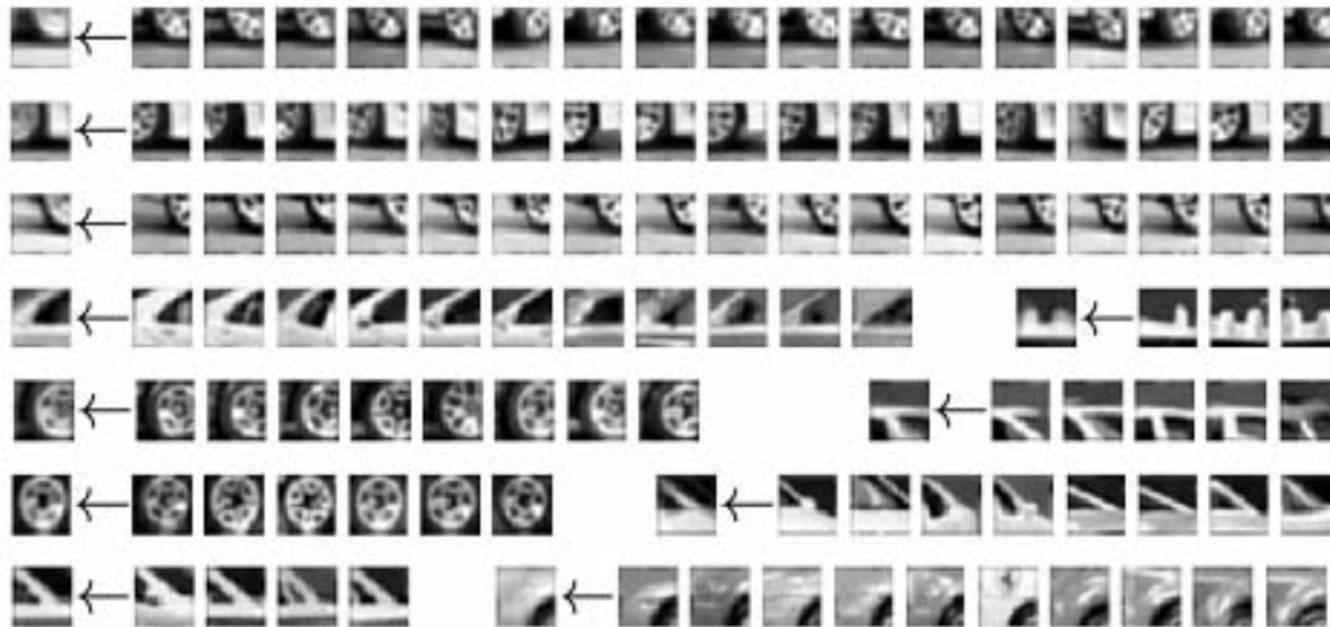
Тестовое изображение

B. Leibe, A. Leonardis, and B. Schiele, [Combined Object Categorization and Segmentation with an Implicit Shape Model](#), ECCV Workshop on Statistical Learning in Computer Vision 2004



Неявные модели формы

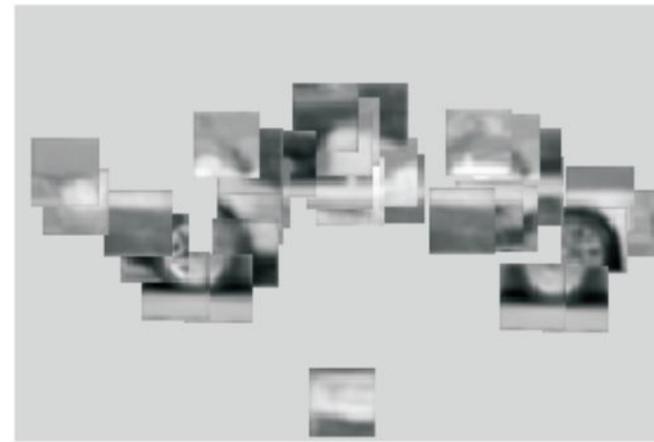
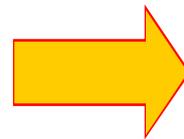
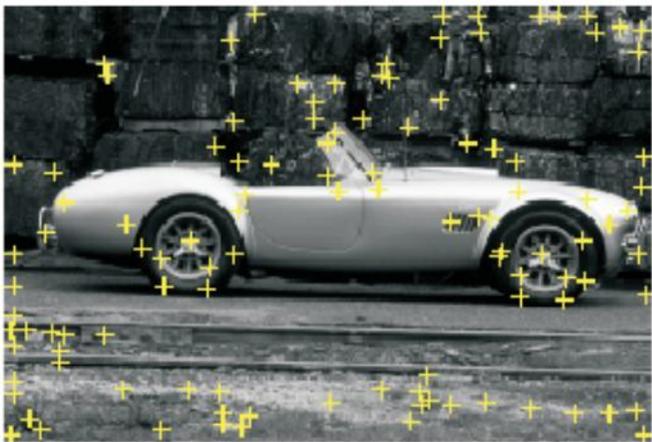
1. Строим словарь фрагментов по окрестностям выделенных особых точек с помощью кластеризации (подробнее дальше)





Неявные модели формы

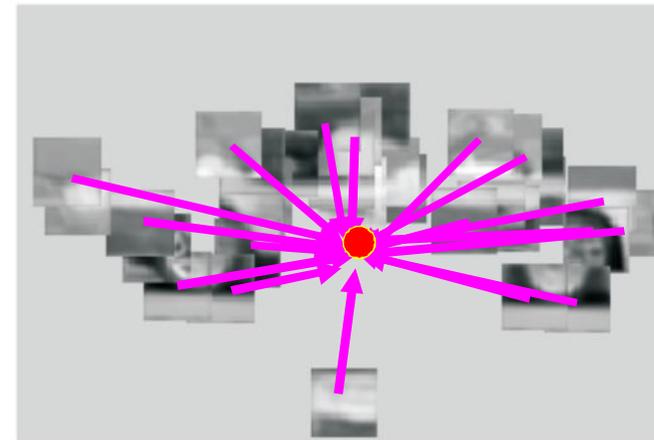
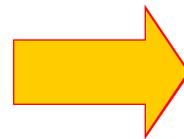
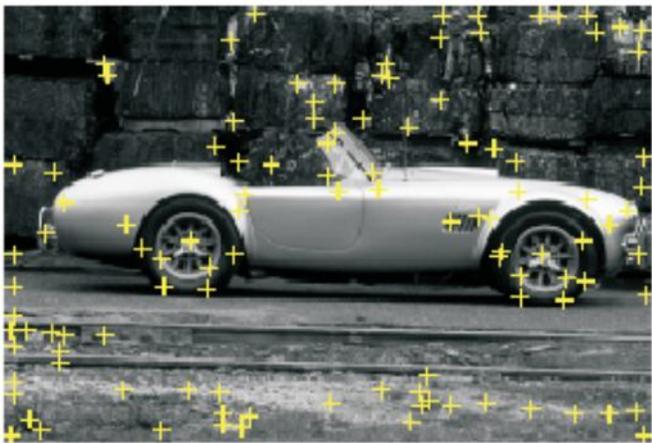
1. Строим словарь фрагментов по окрестностям выделенных особых точек с помощью кластеризации
2. Сопоставляем каждый фрагмент вокруг особой точки с ближайшим значением из словаря





Неявные модели формы

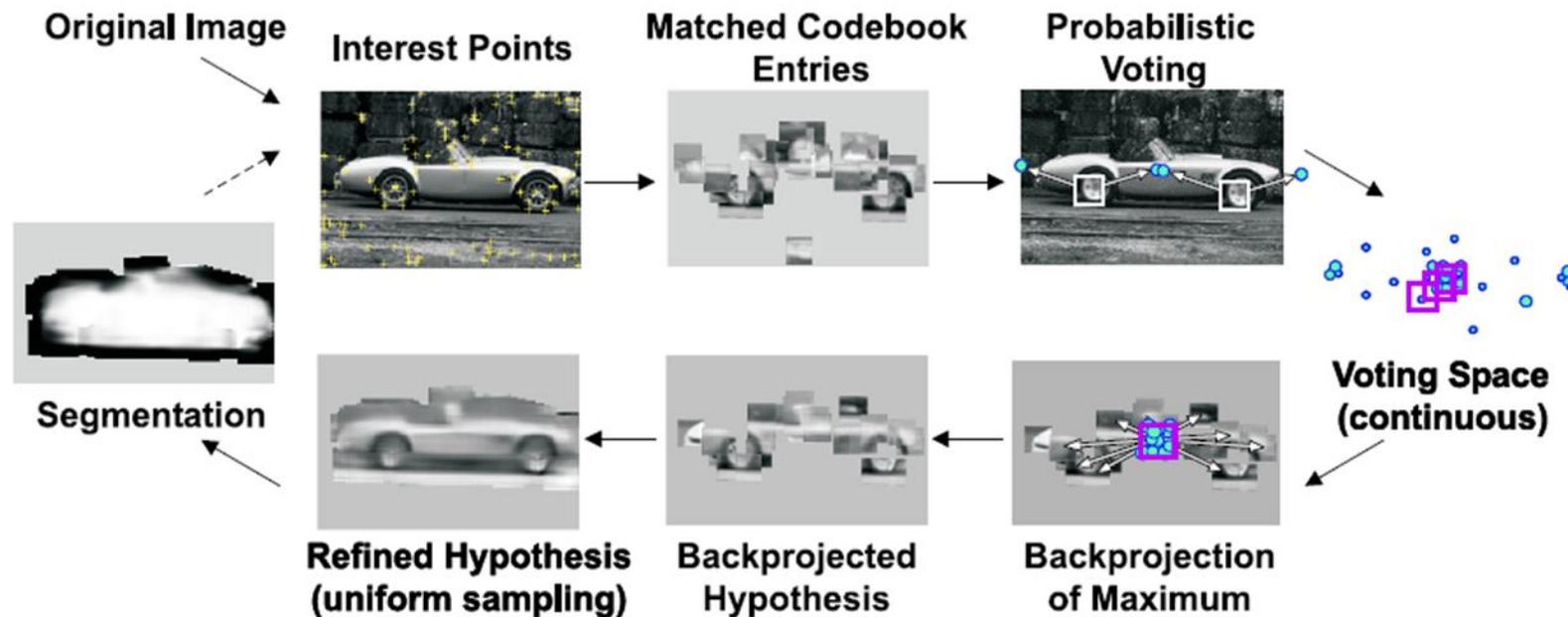
1. Строим словарь фрагментов по окрестностям выделенных особых точек с помощью кластеризации
2. Сопоставляем каждый фрагмент вокруг особой точки с ближайшим значением из словаря
3. Для каждой записи в словаре, сохраняем все вектора смещения (к центру объекта от фрагмента)





Неявные модели формы

1. Извлекаем фрагменты из тестового изображения, ищем соответствия в словаре
2. Голосуем за все центры объекта по словарю
3. Ищем максимумы в фазовом пространстве
4. Строим взвешенную маску сегментации по маскам из словаря





Неявные модели формы

- Обучение с учителем
 - Нужно знать положение объекта и маску сегментации для каждого тренировочного изображения
- Фазовое пространство непрерывное
 - Нужно применять методы кластеризации для поиска максимумов
- Как искать объекты на разных масштабах?
 - Вариант 1: проводить поиск для нескольких масштабов
 - Вариант 2: использовать особые точки, инвариантные к масштабу
- Этап верификации очень важен
 - Когда гипотеза о положении объекта получена, накладываем детальный шаблон, проводим попиксельное сравнение и т.д.



Резюме

- **Метод наименьших квадратов**
 - Он же Direct Linear Transform
 - Базовый метод оценки параметров модели по избыточным данным
- **Робастные методы**
 - М-оценки, итеративное уточнение
- **Рандомизированные методы – RANSAC**
 - Широкое применение при сопоставлении изображений
- **Схемы голосования**
 - Поиск объектов на изображении – линии, окружности, распознавание



На следующей лекции

- Классификация изображений
- Машинное обучение
- Метод опорных векторов
- Мешок слов