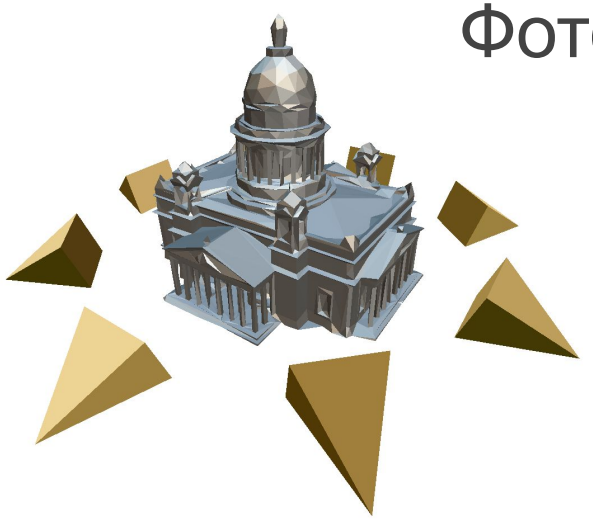
Уточнение расположения камер и их калибровок.



Фотограмметрия. Лекция 4

- Bundle Adjustment
- Ceres Solver

Полярный Николай
polarnick239@gmail.com

План лекции

1) Поиск экстремума 1: градиентный спуск
2) Поиск корня: алгоритм Ньютона
3) Поиск экстремума 2: алгоритм Ньютона
4) Поиск экстремума 3: алгоритм Гаусса-Ньютона
5) Поиск экстремума 4: алгоритм Левенберга — Марквардта
6) Модель камеры (функция проекции)
7) Bundle Adjustment
8) Ceres Solver (авто-дифференцирование)
9) Функции потерь (Loss functions)

Поиск экстремума 1: градиентный спуск

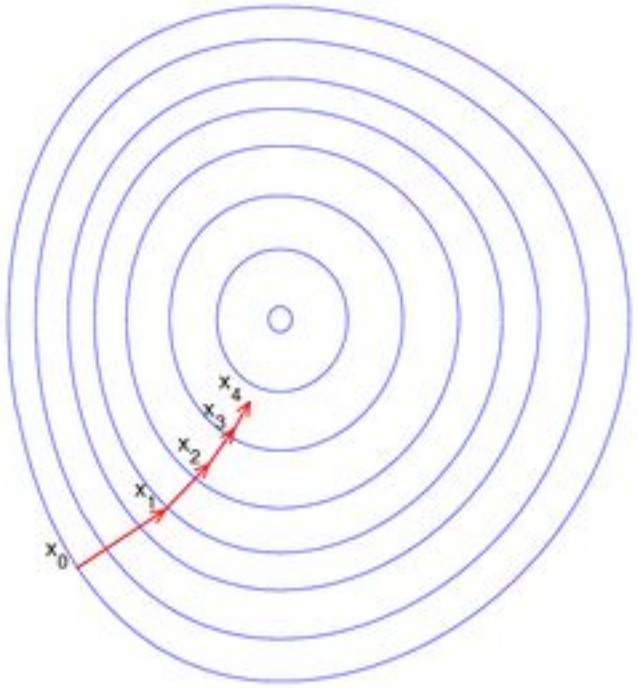
Есть функция $f(x)$ хотим найти точку экстремума, например $\operatorname{argmin}_x f(x)$

Поиск экстремума 1: градиентный спуск

Есть функция $f(x)$ хотим найти точку экстремума, например $\arg\min_x f(x)$

Есть первое приближение x_0 давайте решим в какую новую точку стоит попробовать сходить?

Аналогия: вы в лесу на горе, вокруг туман, как дойти до вершины?



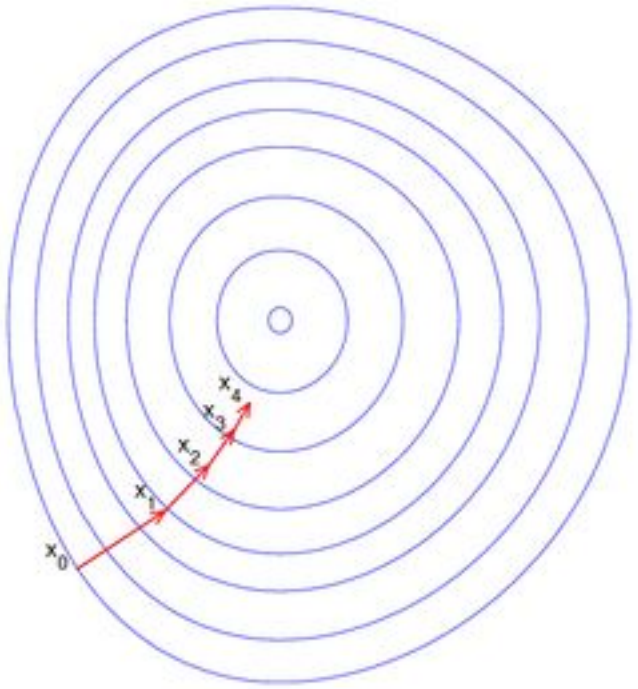
Поиск экстремума 1: градиентный спуск

Есть функция $f(x)$ хотим найти точку экстремума, например $\arg\min_x f(x)$

Есть первое приближение x_0 давайте решим в какую новую точку стоит попробовать сходить?

$$x_{n+1} = x_n - \gamma \nabla f(x_n)$$

Сдвигаемся по направлению наисильнейшего убывания $f(x)$

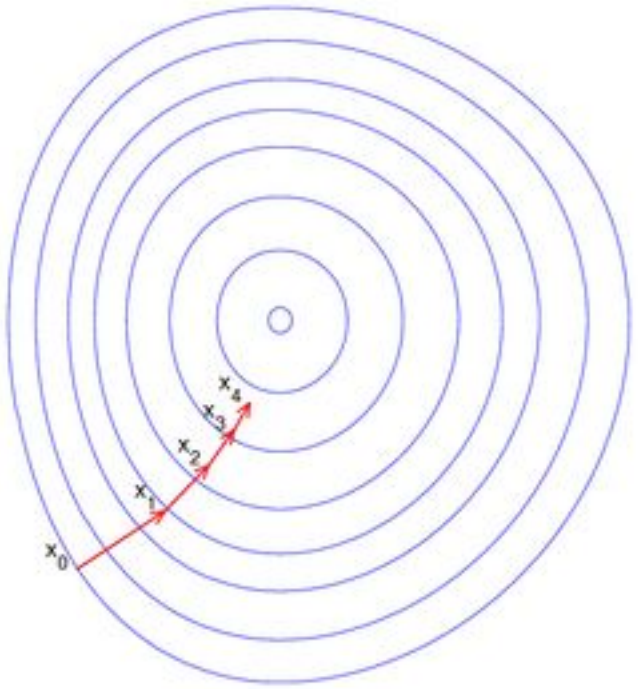


Поиск экстремума 1: градиентный спуск

Есть функция $f(x)$ хотим найти точку экстремума, например $\arg\min_x f(x)$

Есть первое приближение x_0 давайте решим в какую новую точку стоит попробовать сходить?

$$x_{n+1} = x_n - \gamma \nabla f(x_n)$$



Сдвигаемся по направлению наисильнейшего убывания $f(x)$

А как найти максимум? (две формулировки)

Поиск корня: Метод Ньютона

Сначала решим другую задачу: найдем корень функции

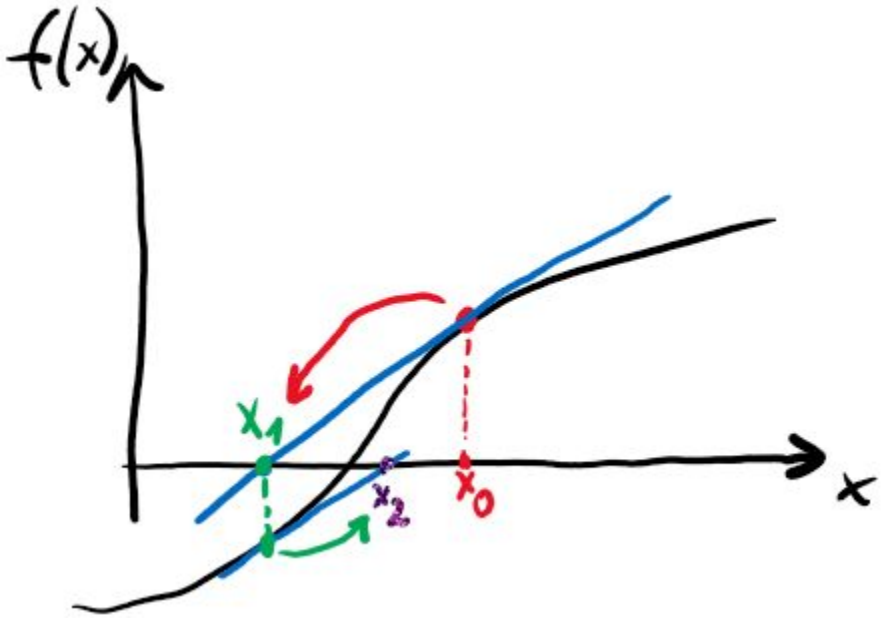
т.е. для $f(x)$ хотим решить уравнение $f(x) = 0$

Поиск корня: Метод Ньютона

Сначала решим другую задачу: найдем корень функции

т.е. для $f(x)$ хотим решить уравнение $f(x) = 0$

1) Считаем производную
2) Проводим касательную
3) Прыгаем в точку пересечения оси
4) Повторяем



Поиск корня: Метод Ньютона

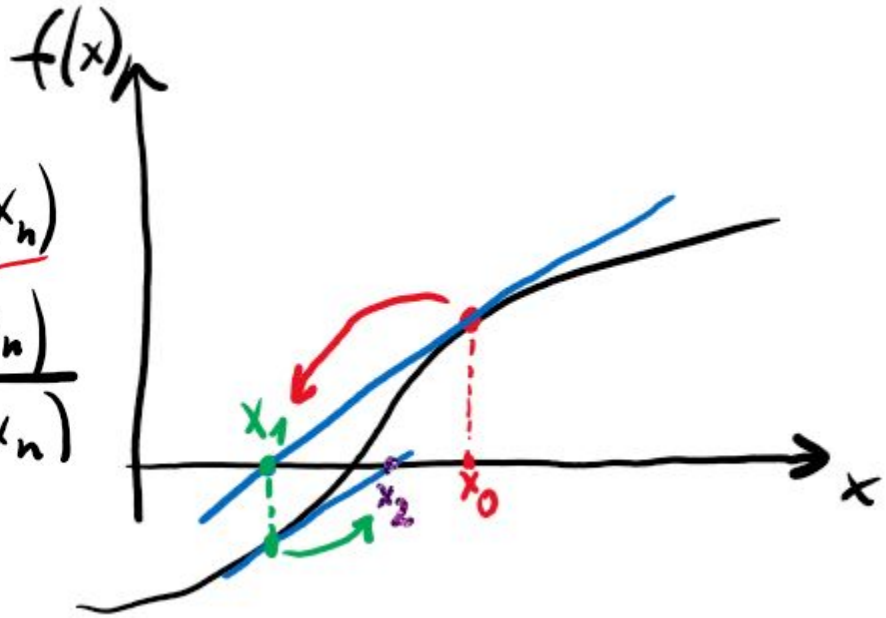
Сначала решим другую задачу: найдем корень функции

т.е. для $f(x)$ хотим решить уравнение $f(x) = 0$

Иначе говоря через ряд Тейлора:

$$f(x_{n+1}) \approx f(x_n) + (x_{n+1} - x_n) f'(x_n)$$

$$f(x_{n+1}) = 0 \Rightarrow f(x_n) + (x_{n+1} - x_n) f'(x_n) = 0 \Rightarrow x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



Поиск корня: Метод Ньютона

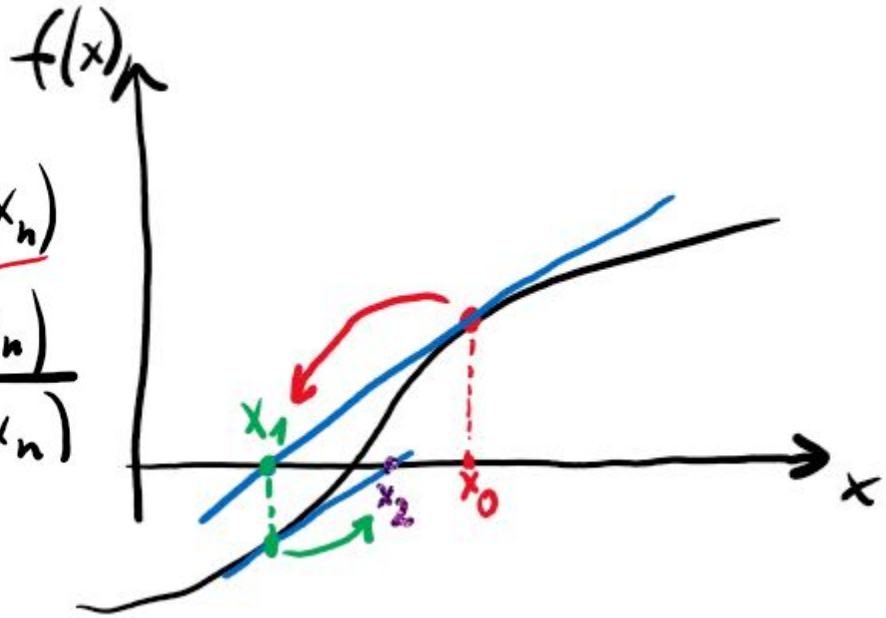
Сначала решим другую задачу: найдем корень функции

т.е. для $f(x)$ хотим решить уравнение $f(x) = 0$

Иначе говоря через ряд Тейлора:

$$f(x_{n+1}) \approx f(x_n) + (x_{n+1} - x_n) f'(x_n)$$

$$f(x_{n+1}) = 0 \Rightarrow f(x_n) + (x_{n+1} - x_n) f'(x_n) = 0 \Rightarrow x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



Для каких функций мы найдем корень за один шаг?

Как искать локальный экстремум?

Поиск экстремума 2: алгоритм Ньютона

Есть функция $f(x)$ хотим найти точку экстремума, т.е. $f'(x) = 0$

Переформулируем:

Есть функция $f'(x)$ хотим решить уравнение $f'(x) = 0$

Поиск экстремума 2: алгоритм Ньютона

Есть функция $f(x)$ хотим найти точку экстремума, т.е. $f'(x) = 0$

Переформулируем:

Есть функция $f'(x)$ хотим решить уравнение $f'(x) = 0$

Свели задачу к поиску корня алгоритмом Ньютона:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

Если $f'(x)$ - похожа на линейную функцию в окрестности содержащей нас и экстремум - то прыгнем в экстремум **за один шаг!**

$f'(x)$ - похожа на линейную $\Leftrightarrow$ $f(x)$ похожа на кривую второго порядка.

Поиск экстремума 2: алгоритм Ньютона

Есть функция $f(x)$ хотим найти точку экстремума, т.е. $f'(x) = 0$

Переформулируем:

Есть функция $f'(x)$ хотим решить уравнение $f'(x) = 0$

Свели задачу к поиску корня алгоритмом Ньютона:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

И для произвольной размерности:

$$x_{n+1} = x_n - [Hf(x_n)]^{-1} \nabla f(x_n)$$

Гессиан

Поиск экстремума 3: алгоритм Гаусса-Ньютона

Пусть есть функция и наблюдения.

Функция может быть нелинейной, поэтому **Non-linear Least squares**.

$$g(x): \mathbb{R}^n \to \mathbb{R}$$

m наблюдений: y_i

ищем такой $X \in \mathbb{R}^n$, что:

$$\forall\ i = 1 \dots m: \quad g(x) \approx y_i$$

Поиск экстремума 3: алгоритм Гаусса-Ньютона

Переформулируем:

$$\forall\ i=1\dots m:\ g(x)\approx y_i$$

$$f(\underset{\mathbb{R}^n}{x}) = \frac{1}{2}\sum_{i=1\dots m}\left(g(x)-y_i\right)^2 = \frac{1}{2}\sum r_i(x)^2 = r(x)^T r(x)$$

$$r(x) = \begin{pmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_m(x) \end{pmatrix}$$

x — (искомый параметр)

r_i — residual (невязка)

Итого опять ищем $\underset{x}{\operatorname{argmin}} f(x)$

Поиск экстремума 3: алгоритм Гаусса-Ньютона

$$f(x) = r(x)^T \cdot r(x) \qquad r(x) = \begin{pmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_m(x) \end{pmatrix}$$

$$\nabla f(x) = \begin{pmatrix} \frac{\partial r_1}{\partial x_1} \\ \frac{\partial r_1}{\partial x_2} \\ \vdots \\ \frac{\partial r_1}{\partial x_m} \end{pmatrix} r_1(x) + \begin{pmatrix} \frac{\partial r_2}{\partial x_1} \\ \frac{\partial r_2}{\partial x_2} \\ \vdots \\ \frac{\partial r_2}{\partial x_m} \end{pmatrix} r_2(x) + \ldots + \begin{pmatrix} \frac{\partial r_m}{\partial x_1} \\ \frac{\partial r_m}{\partial x_2} \\ \vdots \\ \frac{\partial r_m}{\partial x_n} \end{pmatrix} r_m(x)$$

Поиск экстремума 3: алгоритм Гаусса-Ньютона

$$\nabla f(x) = \begin{pmatrix} \frac{\partial r_1}{\partial x_1} \\ \frac{\partial r_1}{\partial x_2} \\ \vdots \\ \frac{\partial r_1}{\partial x_m} \end{pmatrix} r_1(x) + \begin{pmatrix} \frac{\partial r_2}{\partial x_1} \\ \frac{\partial r_2}{\partial x_2} \\ \vdots \\ \frac{\partial r_2}{\partial x_m} \end{pmatrix} r_2(x) + \ldots + \begin{pmatrix} \frac{\partial r_m}{\partial x_1} \\ \frac{\partial r_m}{\partial x_2} \\ \vdots \\ \frac{\partial r_m}{\partial x_m} \end{pmatrix} r_m(x)$$

$$\nabla f(x) = \begin{pmatrix} \frac{\partial r_1}{\partial x_1} & \cdots & \frac{\partial r_m}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial r_1}{\partial x_m} & & \frac{\partial r_m}{\partial x_m} \end{pmatrix} \begin{pmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{pmatrix} = J^T(x) r(x)$$

Якобиан (трансп.)

Поиск экстремума 3: алгоритм Гаусса-Ньютона

$$\nabla f(x) = J^T(x) r(x)$$

Предположим, что $r(x)$ – линейна (ряд Тейлора I-го порядка)

$$f(x) = \frac{1}{2} \| Jx + r(0) \|^2$$

тогда J-матрица константа

$$\nabla f(x) = J^T(x) r(x)$$

Поиск экстремума 3: алгоритм Гаусса-Ньютона

$$\nabla f(x) = J^T(x)\, r(x)$$

В методе Ньютона было: $x_{n+1} = x_n - \left[H(x_n)\right]^{-1} \nabla f(x_n)$, где $\nabla f(x_n) = J^T(x_n)\, r(x_n)$

В методе Гаусса-Ньютона приближенный гессиан:

$$H(x_n) \approx J^T(x_n)\, J(x_n)$$

т.о. метод Гаусса-Ньютона: $x_{n+1} = x_n - \left[J^T(x_n)\, J(x_n)\right]^{-1} J^T(x_n)\, r(x_n)$

Сравнение методов оптимизации (поиска экстремума)

Метод Ньютона:

- хорошо: если окрестность похожа на кривую второго порядка, то прыгнем в экстремум за один шаг
- плохо: требуется считать вторые производные (матрица Гессиана гораздо плотнее матрицы Якобиана)

Метод Гаусса-Ньютона:

- хорошо: если окрестность похожа на кривую второго порядка, то прыгнем в экстремум за один шаг
- хорошо: не нужно считать вторые производные (матрица Якобиана гораздо разреженнее матрицы Гессиана)
- плохо: подходит только для минимизации суммы квадратов

Сравнение методов оптимизации (поиска экстремума)

Метод Ньютона:

- хорошо: если окрестность похожа на кривую второго порядка, то прыгнем в экстремум за один шаг
- плохо: требуется считать вторые производные (матрица Гессиана гораздо плотнее матрицы Якобиана)

Метод Гаусса-Ньютона:

- хорошо: если окрестность похожа на кривую второго порядка, то прыгнем в экстремум за один шаг
- хорошо: не нужно считать вторые производные (матрица Якобиана гораздо разреженнее матрицы Гессиана)
- **нас устраивает:** подходит только для минимизации суммы квадратов

Сравнение методов оптимизации (поиска экстремума)

Метод Градиентного спуска:

- плохо: (решаемо) не учитывает кривизну - медленно сходится по пологой поверхности, далеко прыгает при движении по резким обрывам, пример: протяженная впадина-овраг
- плохо: медленно сходится даже если экстремум близко
- **хорошо: всегда движется в правильном направлении**

Метод Гаусса-Ньютона:

- хорошо: если окрестность (содержащая текущую точку и экстремум) похожа на кривую второго порядка, то прыгнем в экстремум за один шаг
- **плохо: может прыгнуть не туда**

Как взять лучшее от Градиентного спуска и Гаусса-Ньютона?

Поиск минимума: алгоритм Левенберга — Марквардта

Градиентный спуск: $x_{n+1} = x_n - \lambda \nabla f(x_n) = x_n - \lambda J^T(x_n) r(x_n)$

Гаусс-Ньютон: $x_{n+1} = x_n - \left[J^T(x_n) J(x_n) \right]^{-1} J^T(x_n) r(x_n)$

$J^T(x_n) J(x_n) \approx H(x_n) = \nabla^2 f(x_n)$

Совместим: $x_{n+1} = x_n - \left[J^T(x_n) J(x_n) + \lambda I \right]^{-1} J^T(x_n) r(x_n)$

I — единичная матр.

Если λ мало - Гаусс-Ньютон, если велико - почти градиентный спуск.

Поиск минимума: алгоритм Левенберга — Марквардта

$$x_{n+1} = x_n - \left[J^T(x_n) J(x_n) + \lambda I \right]^{-1} J^T(x_n) r(x_n)$$

единичная матр.

$$\lambda \to +\infty : \quad \lambda \left[J^T(x_n) J(x_n) + \lambda I \right]^{-1} = \left(I - J^T(x_n) J(x_n) / \lambda + \dots \right) \to I$$

А значит при больших лямбдах мы движемся в направлении антиградиента.

Можем ускорить/замедлить темп там где градиент сильный/слабый - учтя кривизну, т.е. заменив I на $\operatorname{diag}(J^T J)$ (тогда будем идти по основанию оврага).

Поиск минимума: алгоритм Левенберга — Марквардта

Как менять параметр Марквардта лямбду?

Уменьшать если значение функции уменьшается (хотим лучше угадывать шаги методом Гаусса-Ньютона чтобы быстрее сойтись).

В результате когда мы будем рядом с минимумом - мы прыгнем в него Гаусс-Ньютоном за мало шагов, т.к. окрестность у экстремума обычно похожа на поверхность второго порядка.

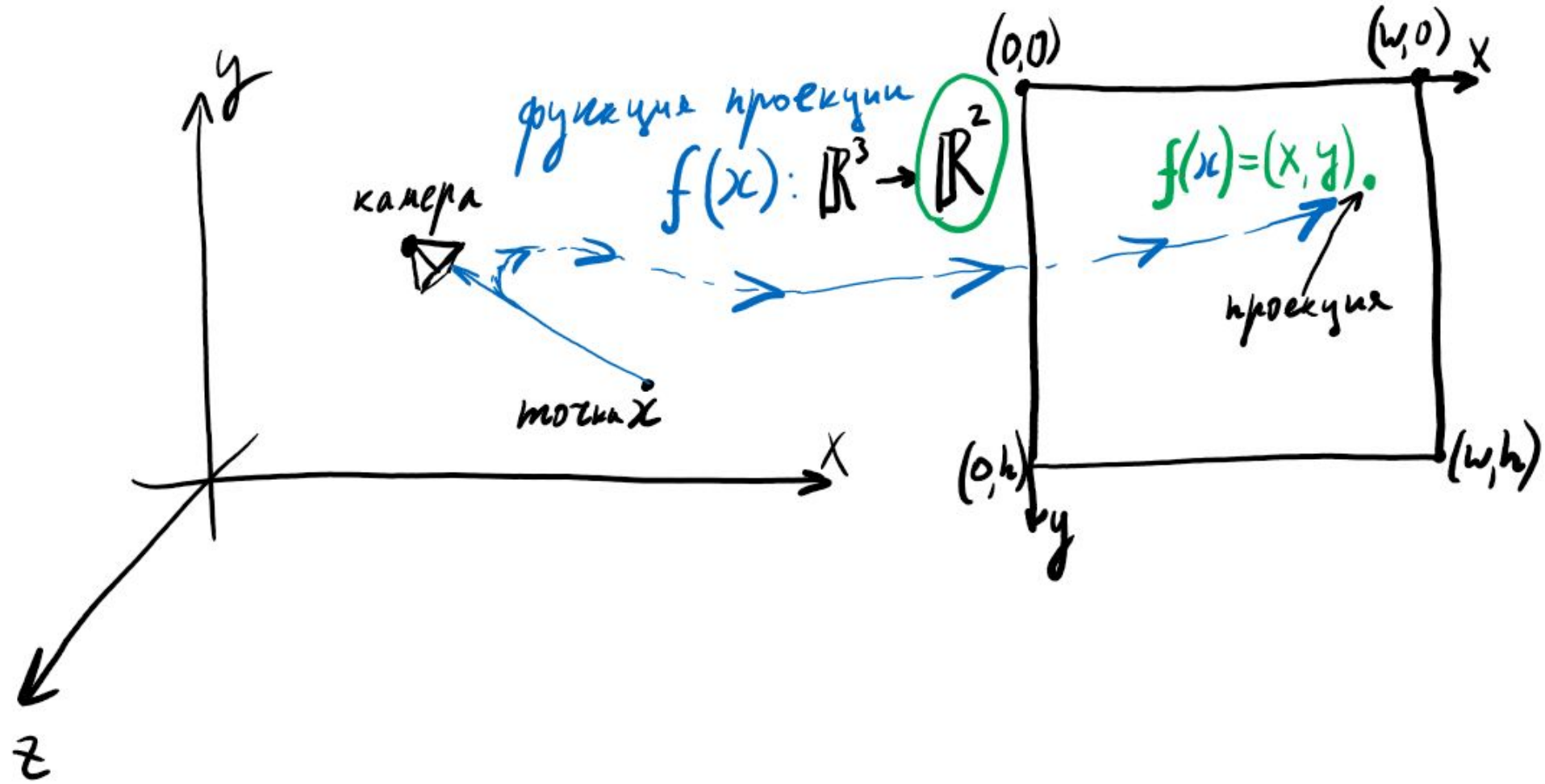
Увеличивать если что-то пошло не так и функция увеличилась (паникуем и переходим на простой и надежный но медленный - градиентный спуск).

Что же мы хотим сделать?

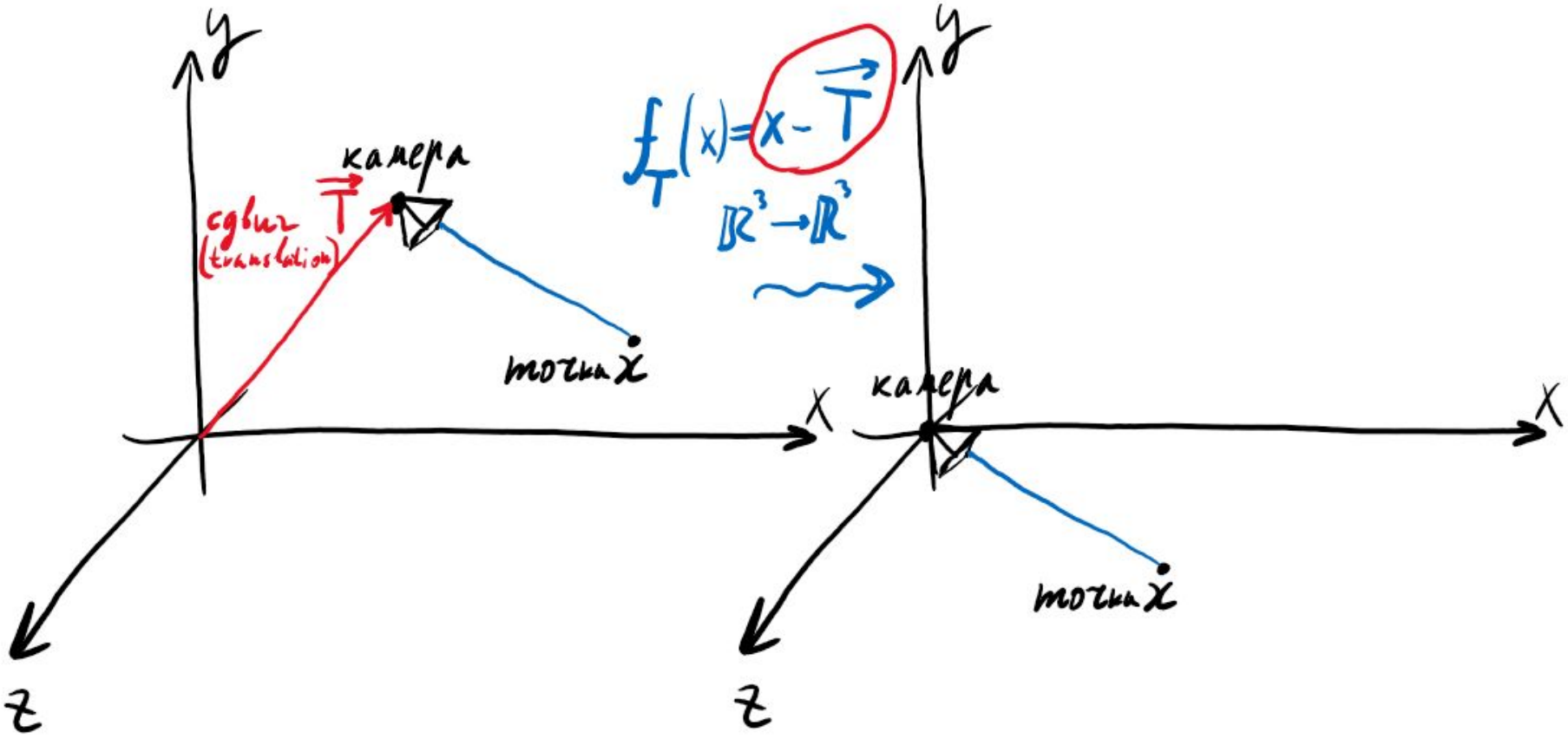
1) Ключевые точки (SIFT)
2) Сопоставление ключевых точек (RANSAC, k-ratio test)
3) Определение взаимного расположения пар камер
4) **Уточнение расположения камер и их калибровок (Bundle Adjustment)**

Модель камеры: функция проекции

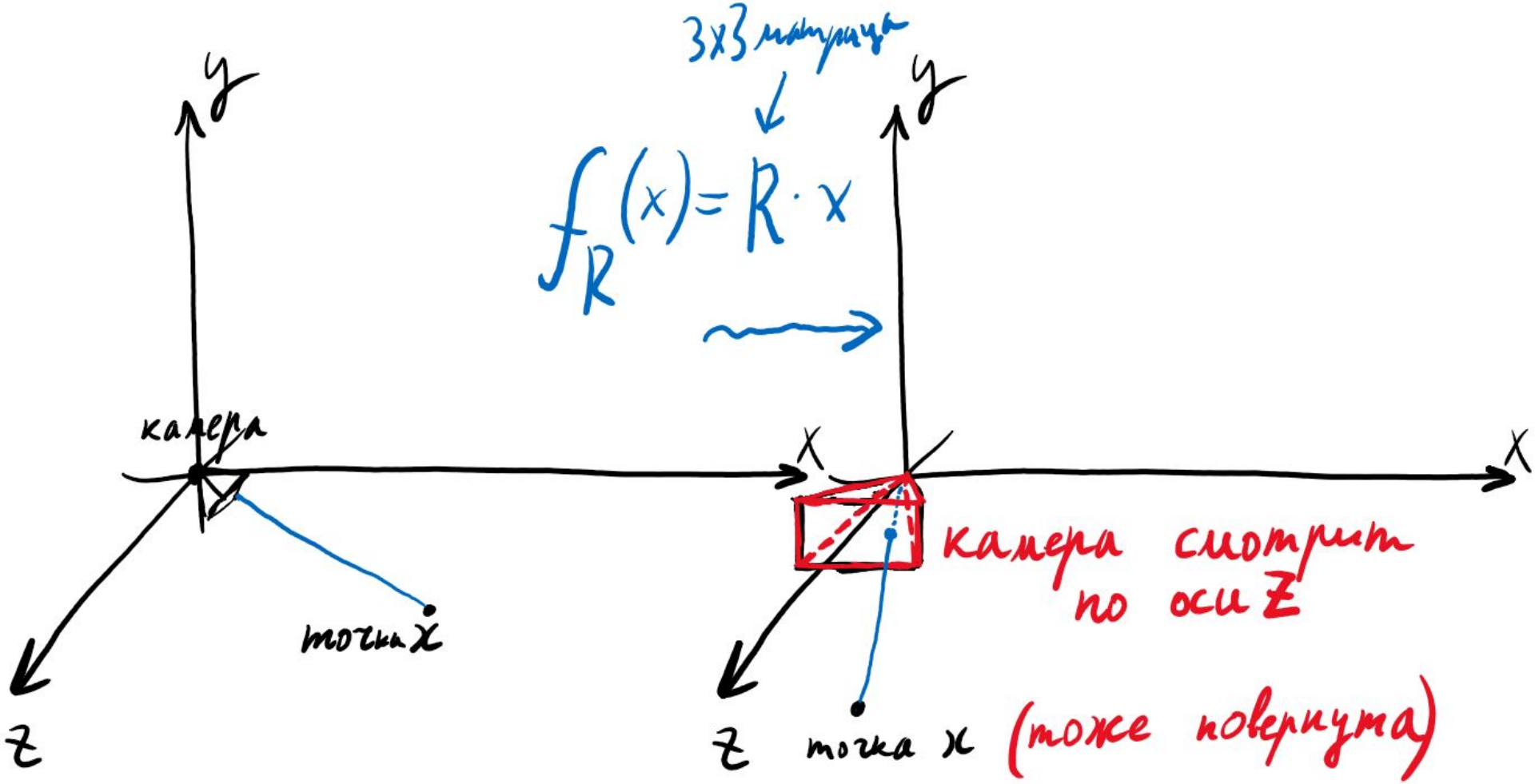
Определим функцию проекции:



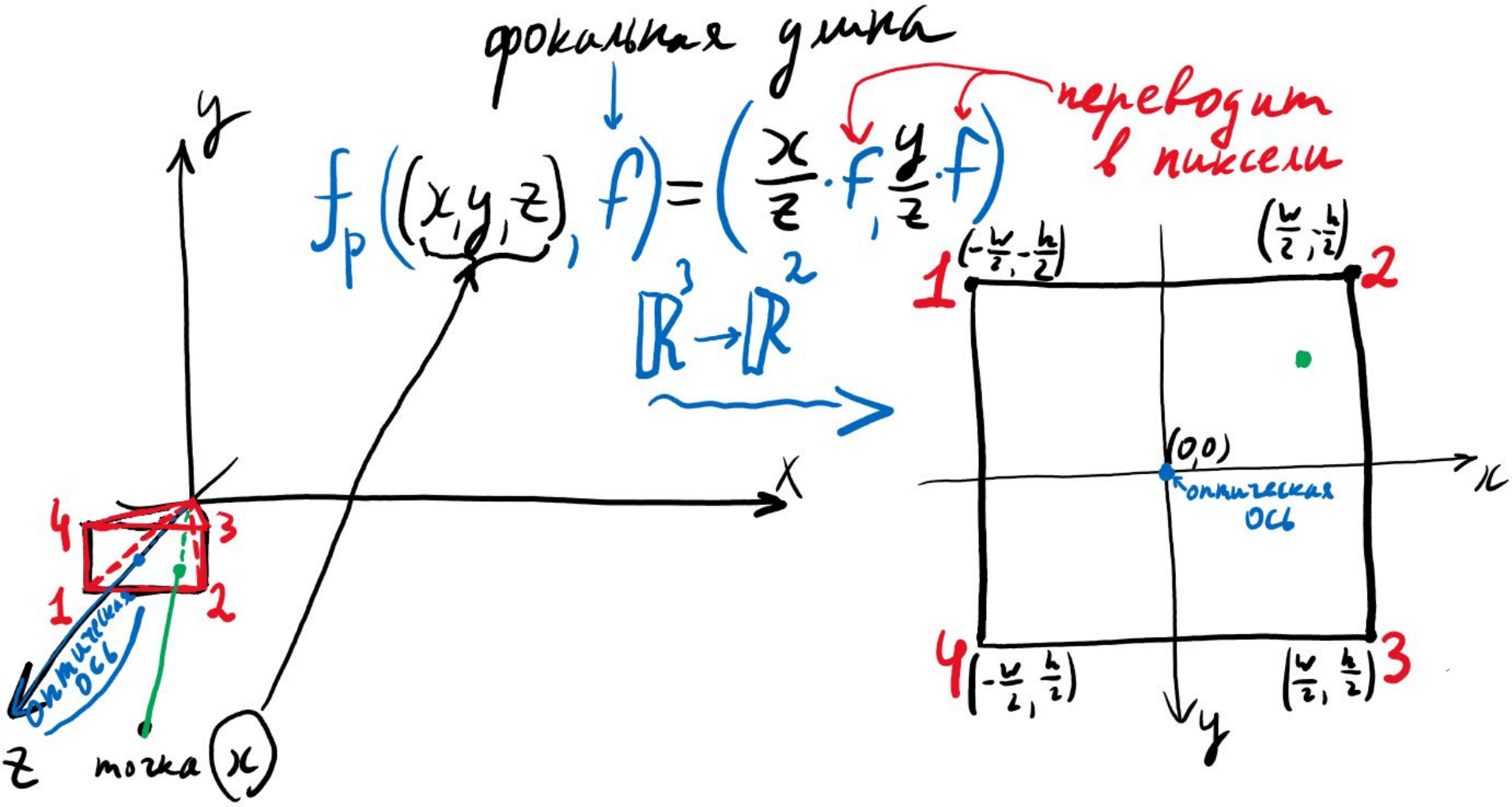
Проекция: сдвиг в локальную систему координат камеры



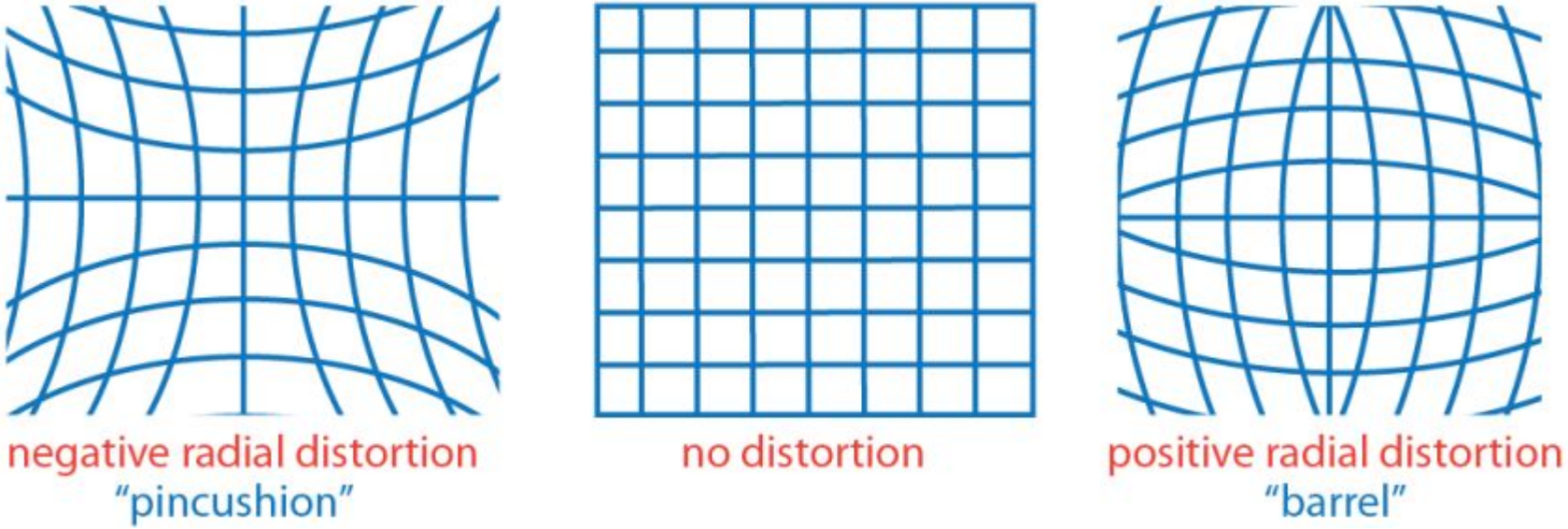
Проекция: поворот локальной системы координат



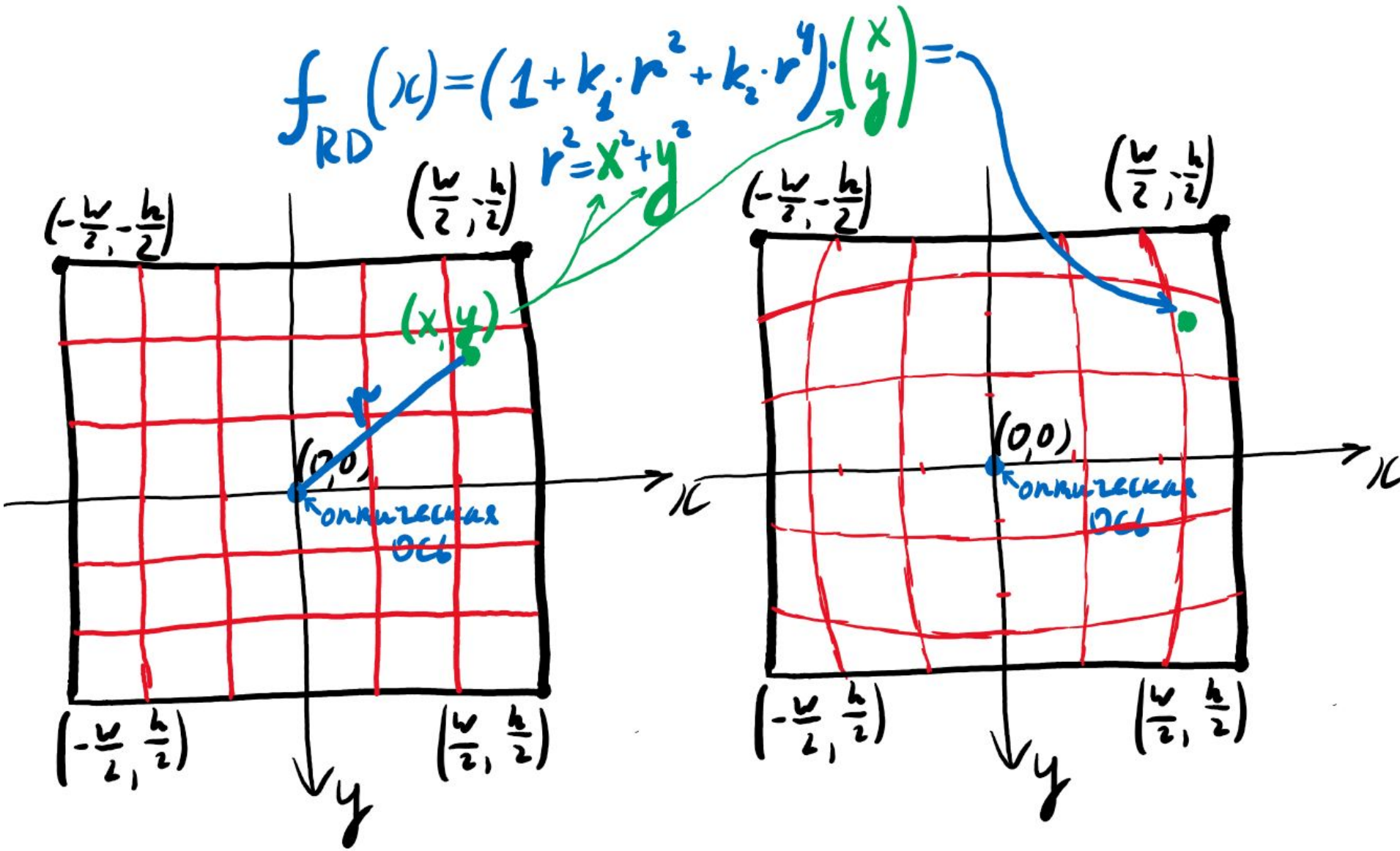
Проекция: на плоскость изображения



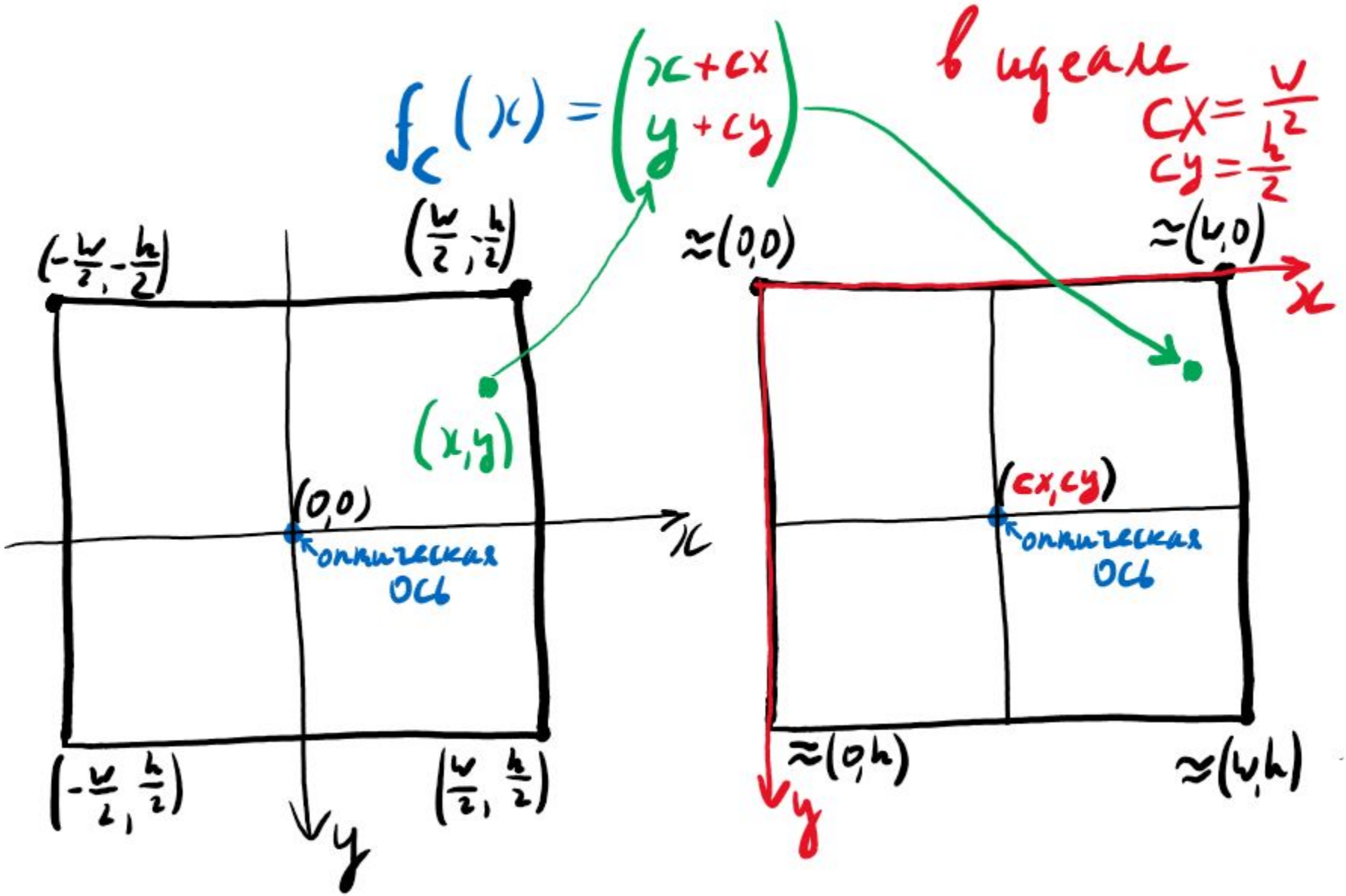
Проекция: учет радиальных искажений



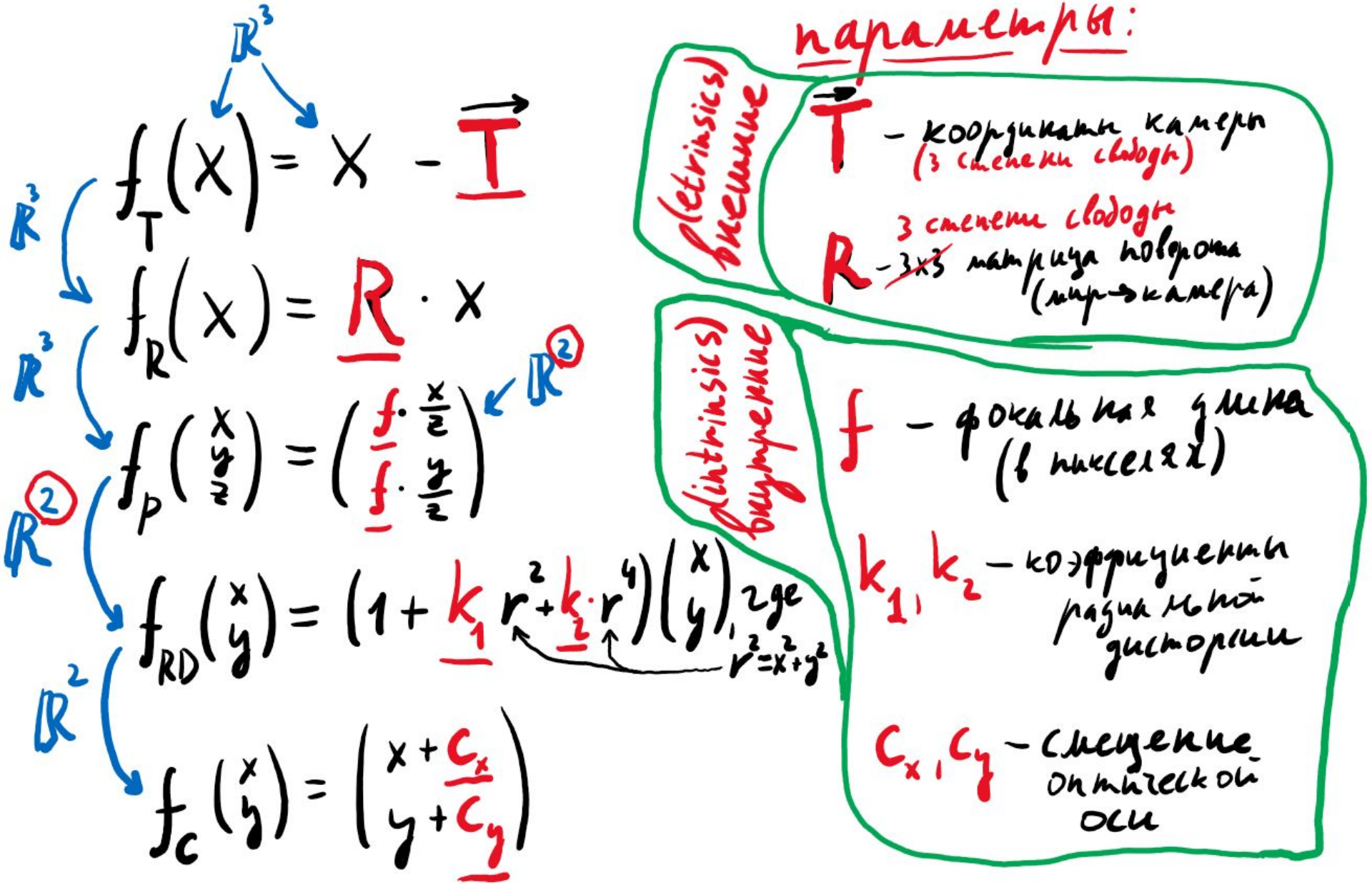
Проекция: учет радиальных искажений



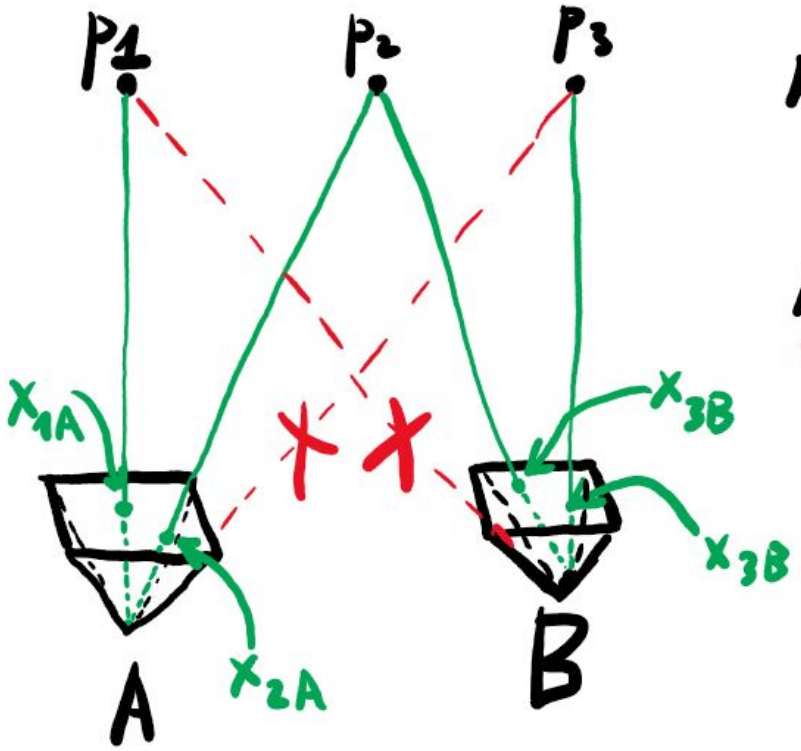
Проекция: учет смещения оптической оси



Модель камеры: функция проекции



Bundle Adjustment



проекция $g(x): \mathbb{R}^3 \to \mathbb{R}^2$

невязки (residuals):

$$r_{1A}(x) = g_A(P_1) - x_{1A}$$
$$r_{1B}(x) = g_B(P_1) - x_{1B}$$
$$r_{2A}(x) = g_A(P_2) - x_{2A}$$
$$r_{2B}(x) = g_B(P_2) - x_{2B}$$

m наблюдений

минимизируем

$$f(x) = \frac{1}{2} \sum_{ij=1\ldots m} \left(g_j(P_i) - x_i\right)^2$$

Bundle Adjustment

Structure from motion: **structure** of the points, **motion** of the cameras.

Bundle - связка пучков-лучей из центров камер к 3D точкам, эту связку итеративно улучшают (**Adjustment**) через минимизацию ошибки репроекции.

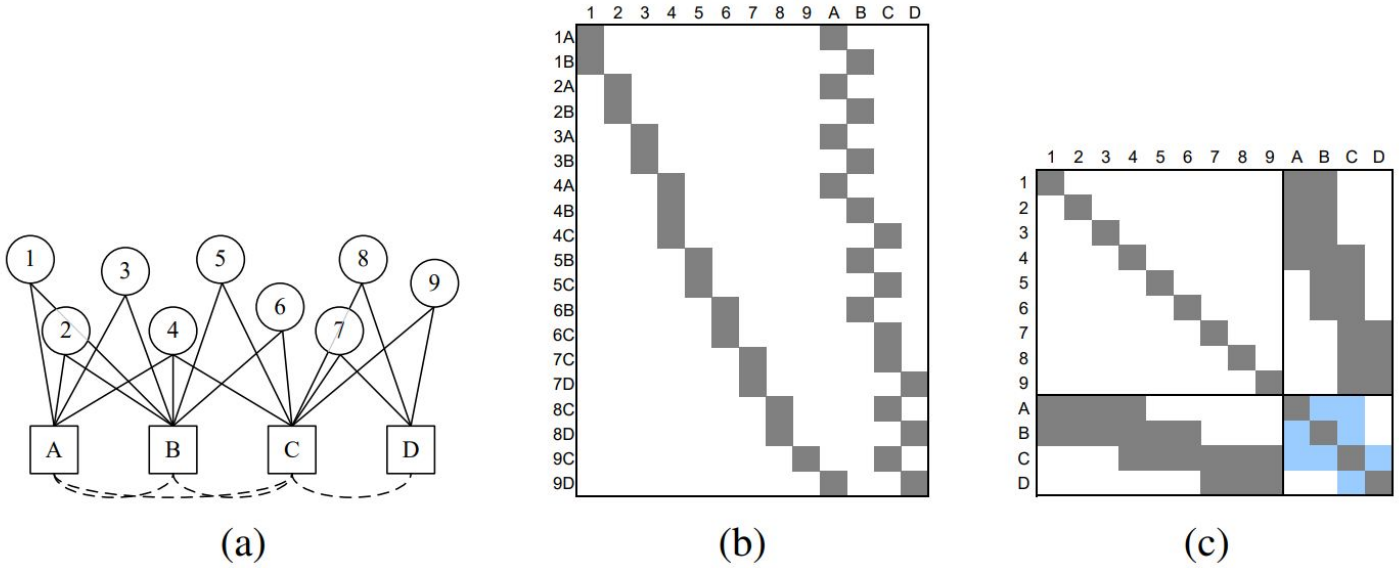


Figure 11.16 *(a) Bipartite graph for a toy structure from motion problem and (b) its associated Jacobian* **J** *and (c) Hessian* **A**. *Numbers indicate 3D points and letters indicate cameras. The dashed arcs and light blue squares indicate the fill-in that occurs when the structure (point) variables are eliminated.*

- Multiple View Geometry in Computer Vision, Richard Hartley & Andrew Zisserman (здесь особенно много деталей, в т.ч. про L-M)
- Computer Vision: Algorithms and Applications, Richard Szeliski

Ceres-solver

Non-linear Least Squares

Introduction

Ceres can solve bounds constrained robustified non-linear least squares problems of the form

$$\min_{\mathbf{x}} \quad \frac{1}{2} \sum_{i} \quad \left(\| f_i \left(x_{i_1}, \ldots, x_{i_k} \right) \|^2 \right)$$

Ceres-solver

Non-linear Least Squares

Introduction

Ceres can solve bounds constrained robustified non-linear least squares problems of the form

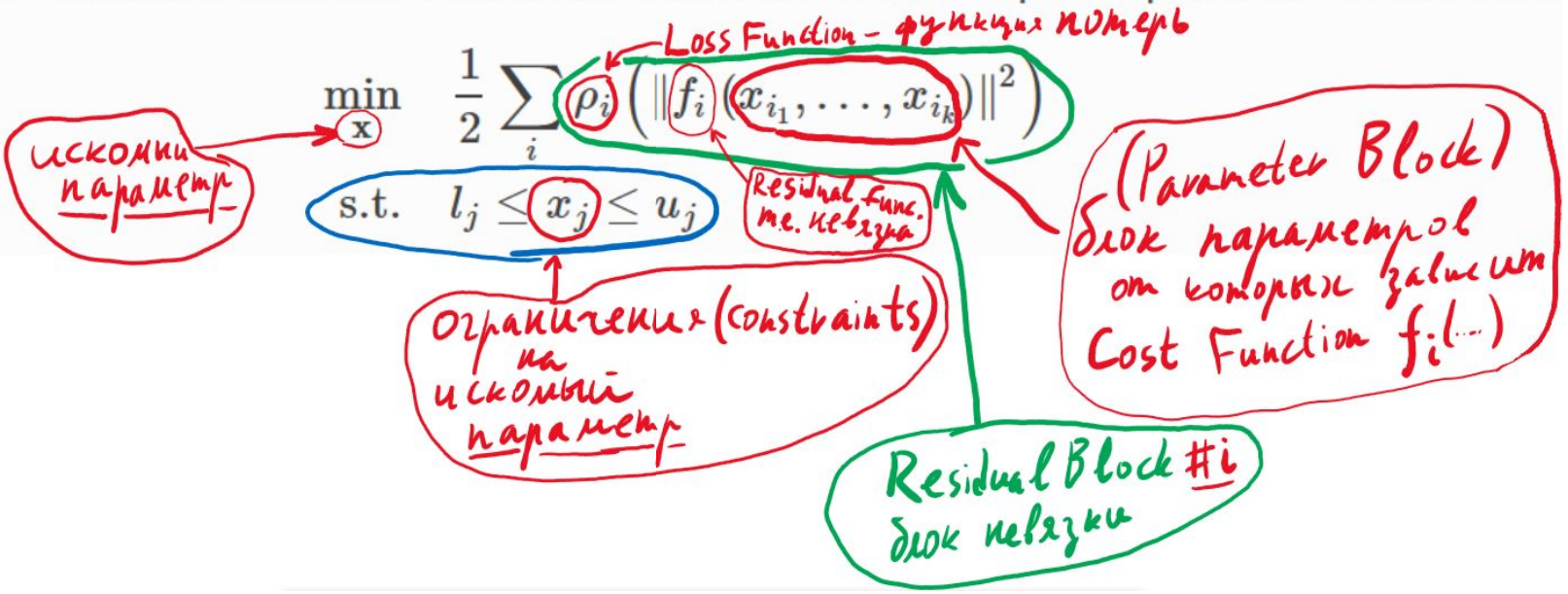
$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{1}{2} \sum_{i} \rho_i \left(\left\| f_i \left(x_{i_1}, \ldots, x_{i_k} \right) \right\|^2 \right) \\ \text{s.t.} \quad & l_j \leq x_j \leq u_j \end{aligned}$$

Ceres-solver

Non-linear Least Squares

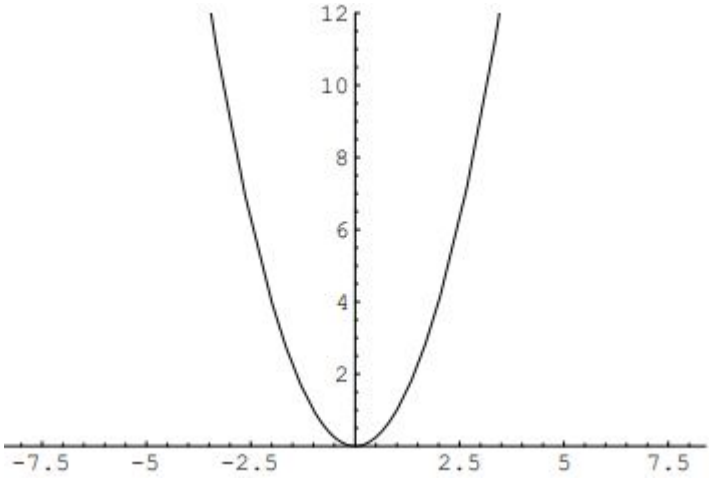
Introduction

Ceres can solve bounds constrained robustified non-linear least squares problems of the form



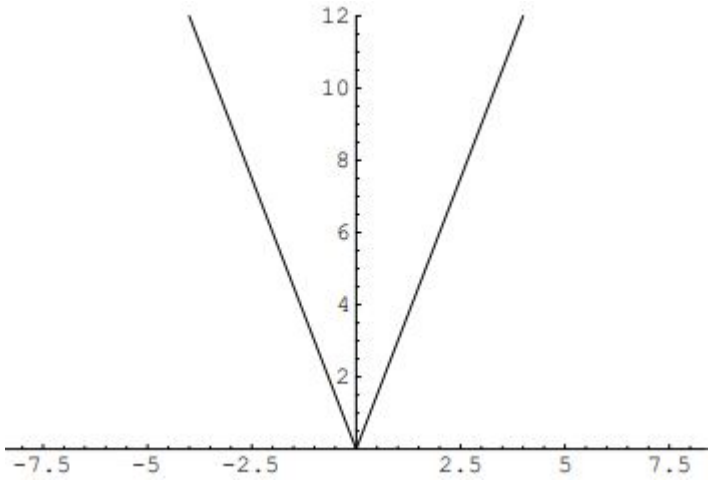
Loss functions (функции потерь)

$$C(\delta) = \delta^2$$



Squared-Error
L2 norm
Trivial Loss

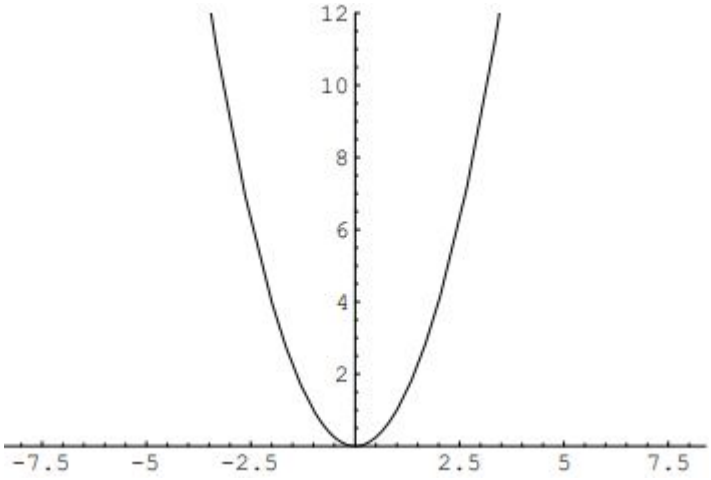
$$C(\delta) = |\delta|$$



L1 norm
Total Variation
Absolute Loss

Loss functions (функции потерь)

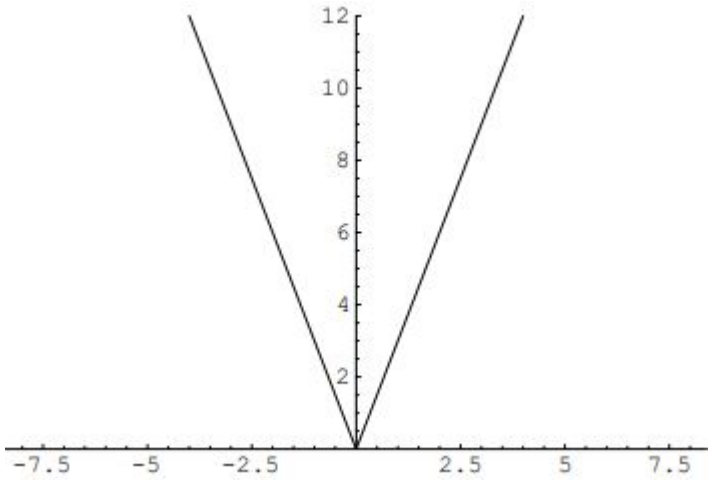
$$C(\delta) = \delta^2$$



Squared-Error
L2 norm
Trivial Loss

- Наблюдения-выбросы (outliers) имеют сильное влияние.

$$C(\delta) = |\delta|$$

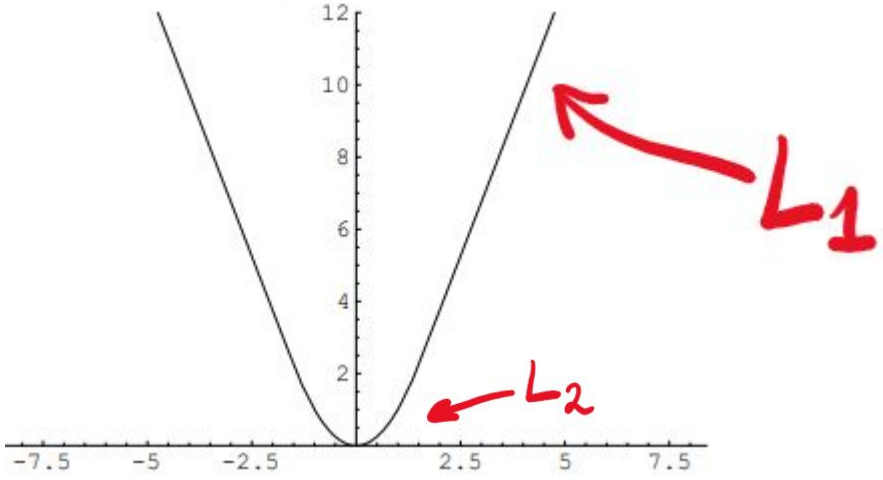


L1 norm
Total Variation
Absolute Loss

- Устойчива к выбросам (robust to outliers).
- Ищет медиану значений, игнорирует остальные значения.

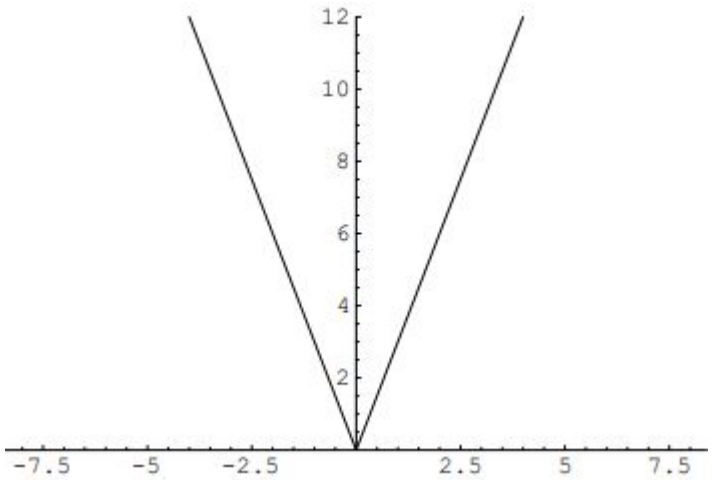
Loss functions (функции потерь)

$$C(\delta) = \delta^2 \text{ for } |\delta| < b$$
$$= 2b|\delta| - b^2 \text{ otherwise}$$



Huber Loss

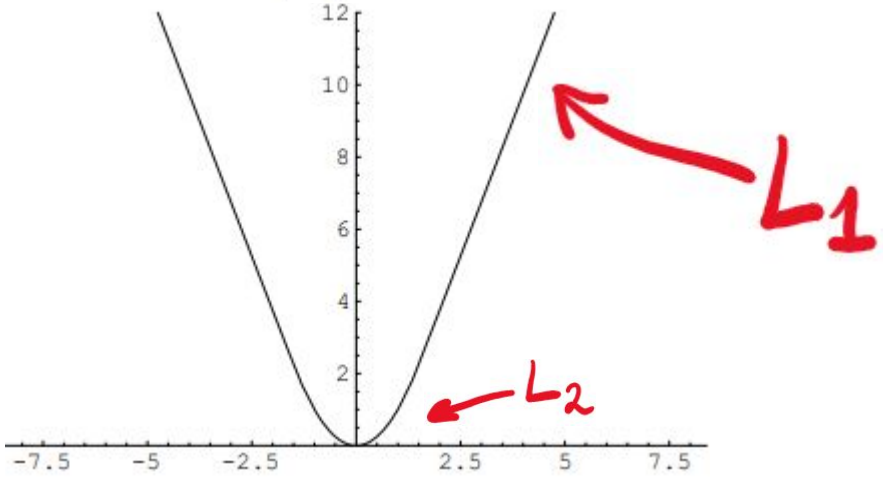
$$C(\delta) = |\delta|$$



L1 norm
Total Variation
Absolute Loss
- Устойчива к выбросам (robust to outliers).
- Ищет медиану значений, игнорирует остальные значения.

Loss functions (функции потерь)

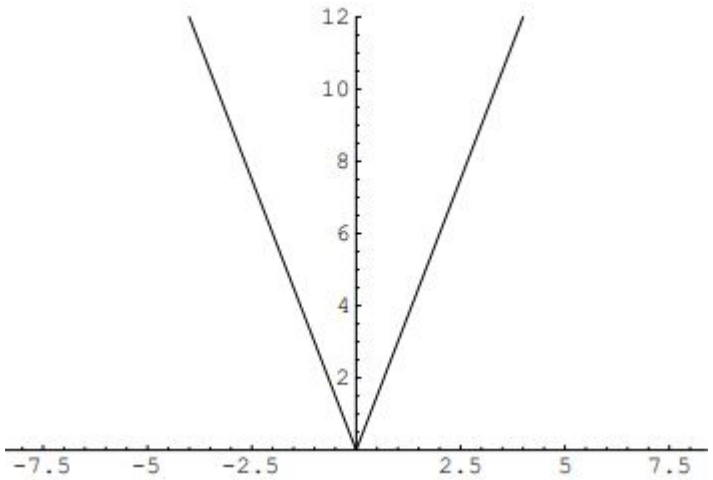
$$C(\delta) = \delta^2 \text{ for } |\delta| < b$$
$$= 2b|\delta| - b^2 \text{ otherwise}$$



Huber Loss

- Устойчива к выбросам (robust to outliers).
- Нет проблемы про медиану (т.е. хорошо приближает норм. распределение).

$$C(\delta) = |\delta|$$

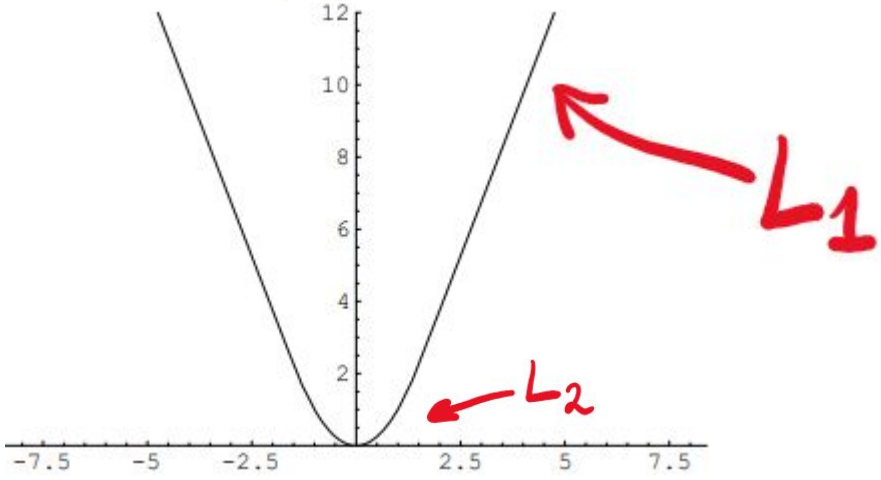


L1 norm
Total Variation
Absolute Loss

- Устойчива к выбросам (robust to outliers).
- Ищет медиану значений, игнорирует остальные значения.

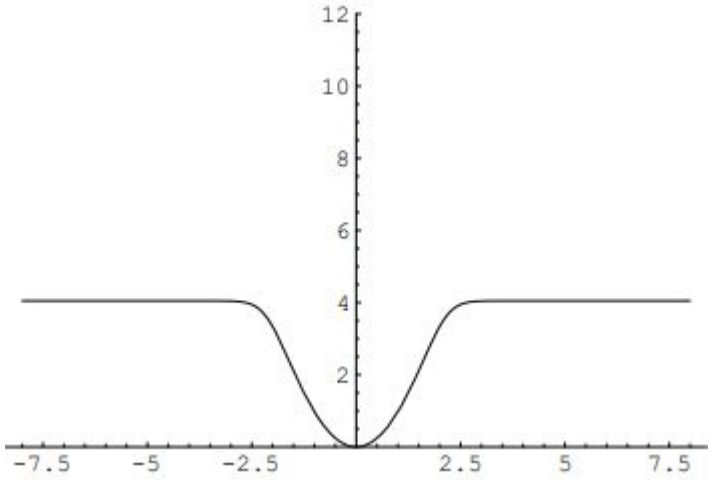
Loss functions (функции потерь)

$$C(\delta) = \delta^2 \text{ for } |\delta| < b$$
$$= 2b|\delta| - b^2 \text{ otherwise}$$



Huber Loss

$$C(\delta) = -\log(\exp(-\delta^2) + \epsilon) \text{ and } \epsilon = \exp(-\alpha^2)$$

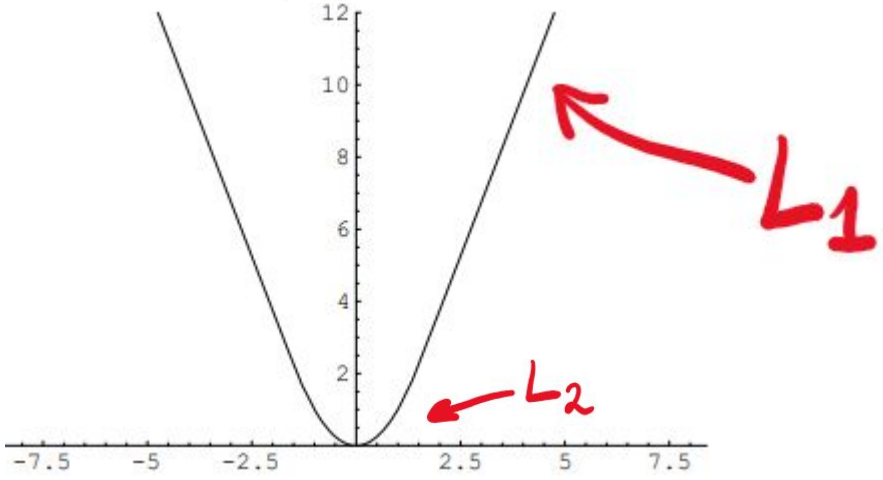


Blake-Zisserman

- Устойчива к выбросам (robust to outliers).
- Нет проблемы про медиану (т.е. хорошо приближает норм. распределение).

Loss functions (функции потерь)

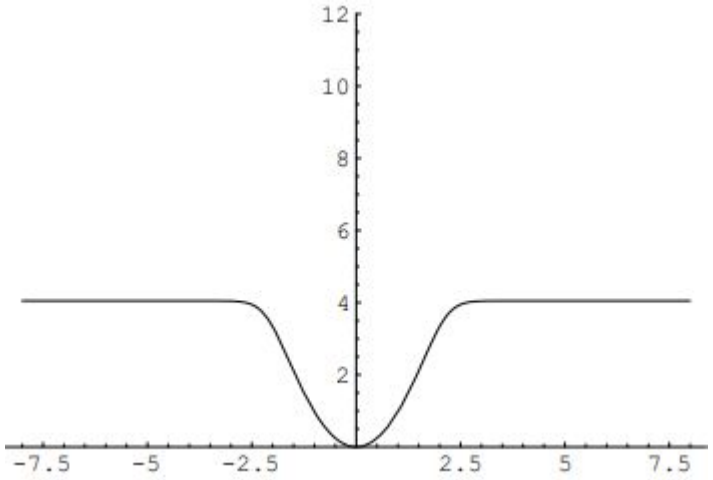
$$C(\delta) = \delta^2 \text{ for } |\delta| < b$$
$$= 2b|\delta| - b^2 \text{ otherwise}$$



Huber Loss

- Устойчива к выбросам (robust to outliers).
- Нет проблемы про медиану (т.е. хорошо приближает норм. распределение).

$$C(\delta) = -\log(\exp(-\delta^2) + \epsilon) \text{ and } \epsilon = \exp(-\alpha^2)$$



Blake-Zisserman

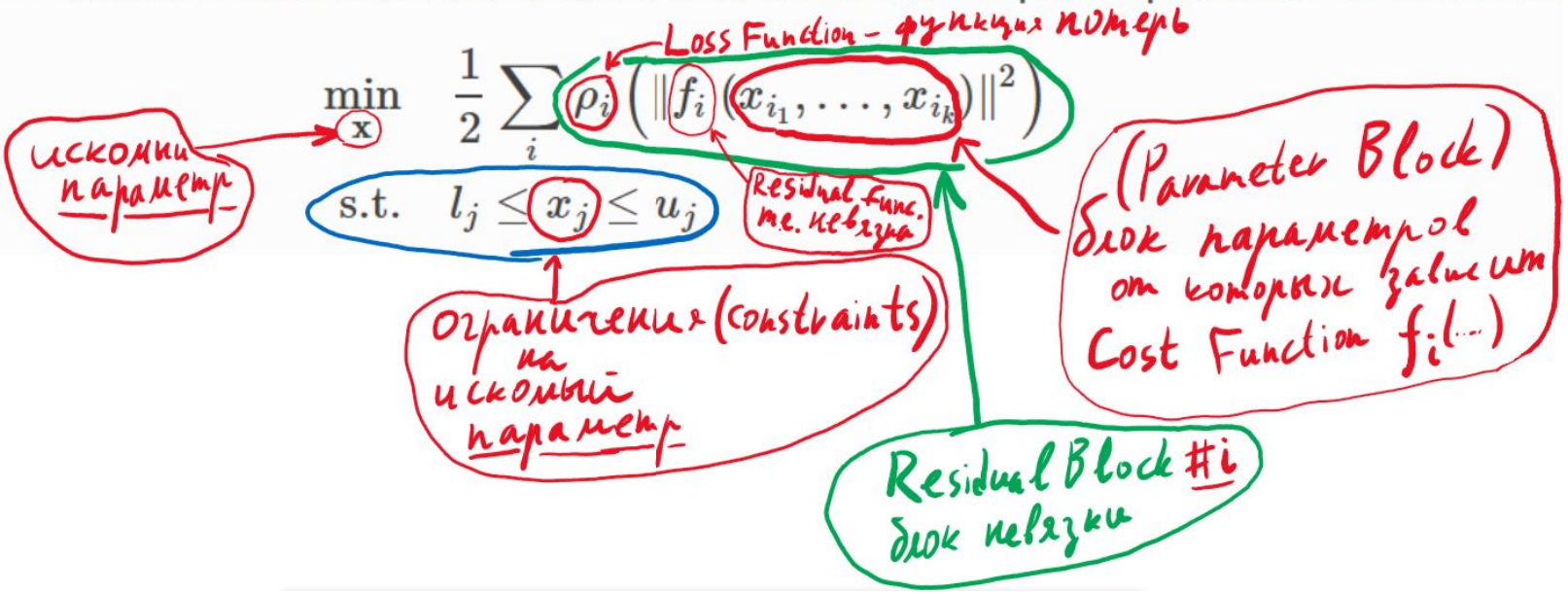
- Супер устойчива к выбросам.
- Если изначальное приближение - плохое, то в нем и застрянем.

Ceres-solver

Non-linear Least Squares

Introduction

Ceres can solve bounds constrained robustified non-linear least squares problems of the form



Примеры

1) Минимизировать $\frac{1}{2}(10 - x)^2$

Какие блоки параметров? Какая функция невязки (residual)?

Примеры

1) Минимизировать $\frac{1}{2}(10 - x)^2$
2) Есть фиксированная прямая и фиксированный параболоид. Найти точку пересечения.

Какие блоки параметров? Какая функция невязки (residual)?

Примеры

1) Минимизировать $\frac{1}{2}(10 - x)^2$
2) Есть фиксированная прямая и фиксированный параболоид. Найти точку пересечения.
3) Есть сколько-то шумных замеров (с выбросами), нужно зафитить прямой.

Какие блоки параметров? Какая функция невязки (residual)? Какая функция потерь (Loss function)?

Примеры

1) Минимизировать $\frac{1}{2}(10 - x)^2$
2) Есть фиксированная прямая и фиксированный параболоид. Найти точку пересечения.
3) Есть сколько-то шумных замеров (с выбросами), нужно зафитить прямой.
4) Bundle Adjustment.

Какие блоки параметров? Какая функция невязки (residual)? Какая функция потерь (Loss function)?

Примеры

1) Минимизировать $\frac{1}{2}(10 - x)^2$
2) Есть фиксированная прямая и фиксированный параболоид. Найти точку пересечения.
3) Есть сколько-то шумных замеров (с выбросами), нужно зафитить прямой.
4) Bundle Adjustment.

Какие блоки параметров? Какая функция невязки (residual)? Какая функция потерь (Loss function)?

Как фильтровать шумные ключевые точки из разреженного облака точек для более надежного последующего добавления очередной камеры?

Примеры

1) Минимизировать $\frac{1}{2}(10 - x)^2$
2) Есть фиксированная прямая и фиксированный параболоид. Найти точку пересечения.
3) Есть сколько-то шумных замеров (с выбросами), нужно зафитить прямой.
4) Bundle Adjustment.

Какие блоки параметров? Какая функция невязки (residual)? Какая функция потерь (Loss function)?

Как фильтровать шумные ключевые точки из разреженного облака точек для более надежного последующего добавления очередной камеры?

3*sigma фильтрация! Считаем ошибку наблюдения, если она в 3 раза выше среднеквадратичного отклонения по всем наблюдениям - выкидываем наблюдение.

Bundle Adjustment. Ссылки

Две ключевые книги-библии:

- Multiple View Geometry in Computer Vision, Richard Hartley & Andrew Zisserman (здесь особенно много деталей, в т.ч. про L-M)
- Computer Vision: Algorithms and Applications, Richard Szeliski

Статьи про детали BA:

- Bundle adjustment—a modern synthesis, Triggs et al.
- Bundle Adjustment in the Large, Agarwal et. al.

Лекции:

- youtube/Behnam Asadi: про методы оптимизации и BA
- youtube/Cyrill Stachniss: BA 1, BA 2
- coursera/Robotics: Perception (BA)

Вопросы?

Полярный Николай
polarnick239@gmail.com