Лекция 11

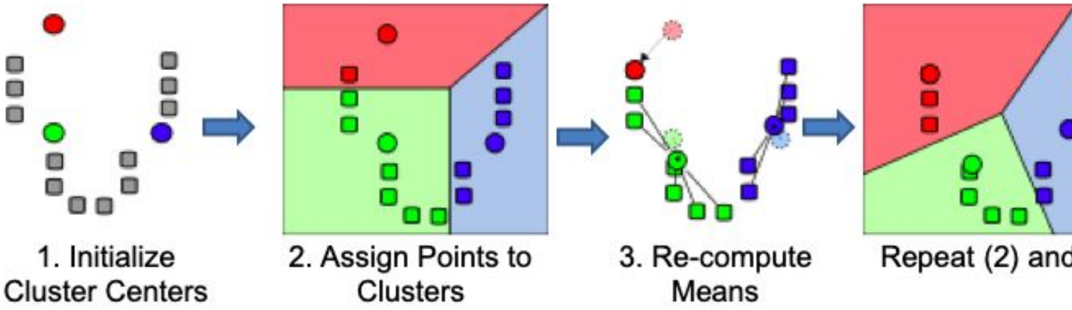
Optical flow. Object detection.

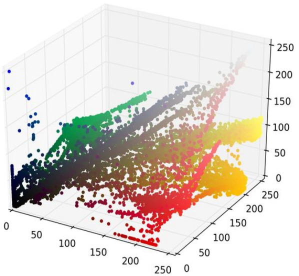
18.02.2020
Руслан Алиев

В предыдущих сериях

Кластеризация

1. K-means
2. Mean-shift
3. Пространство признаков пикселя
   a. Интенсивность
   b. RGB
   c. Отклики на фильтры
   d. Пространственная информация (x,y)





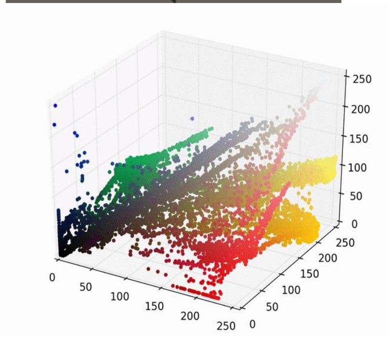


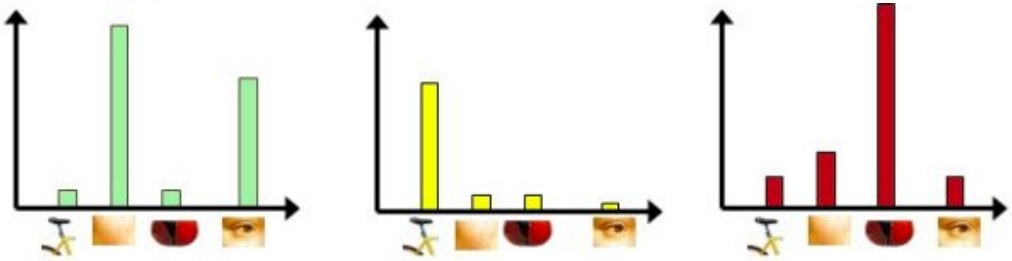
Image compression

1. SVD - самое крутое разложение матриц $U\Sigma V^T = A$
2. С помощью SVD можно получать PCA, уменьшать размерность
3. Изображение - матрица, у нее можно выделить k<<n главных компонент, и получать восстановленное изображение
4. Если у нас есть датасет лиц, используя PCA, мы можем найти “подпространство” лиц, формируемое собственными векторами (“eigenfaces”)


Reconstructed training face

Bag of visual words

1. Извлекаем фичи
2. Строим “визуальный словарь”
3. Для каждой фичи находим ближайшее “слово” из словаря
4. Строим гистограмму частот “слов” для словаря



Optical flow

Темы

- Sparse* Optical Flow (Lucas-Kanade, 1981)
- Dense Optical Flow (Farneback, 2003)

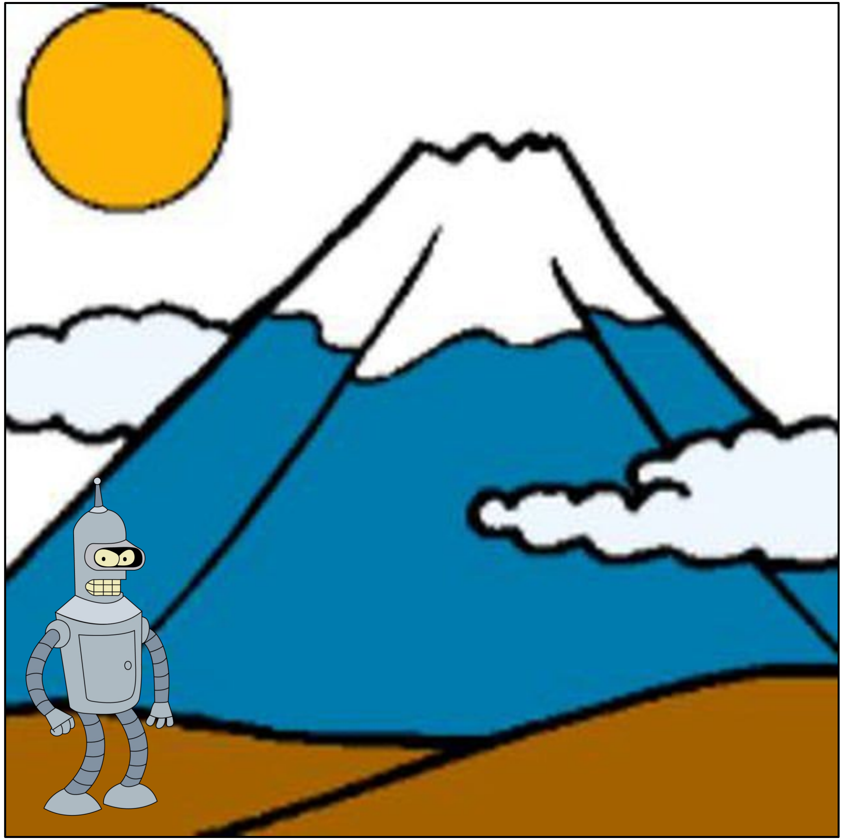
Что такое Optical Flow?

Что такое Optical Flow?

Движение!

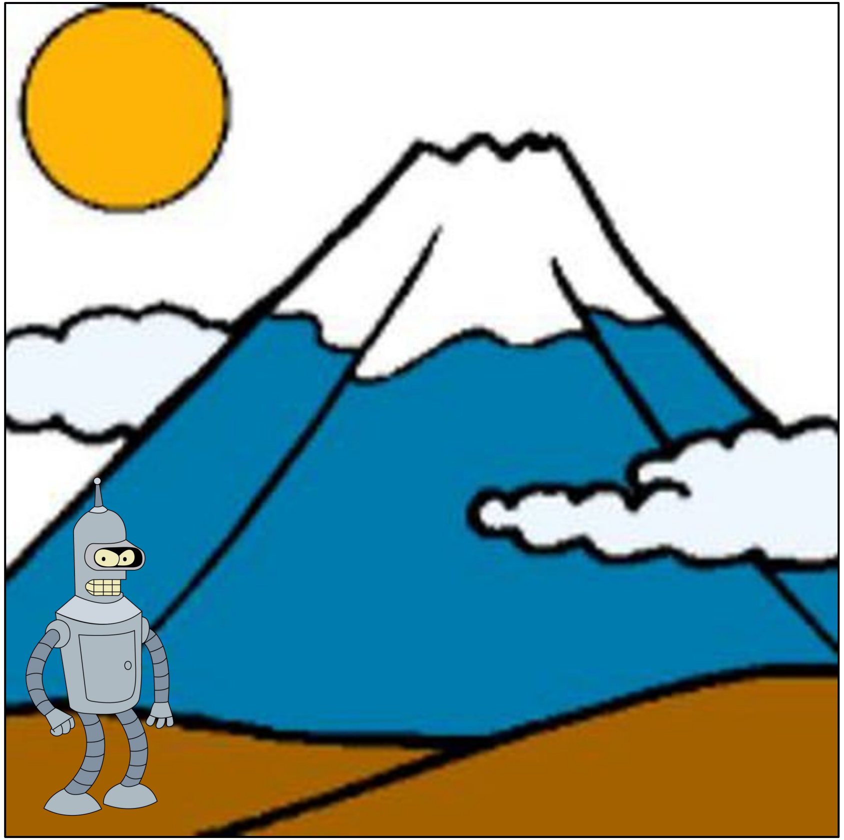
Что такое Optical Flow?

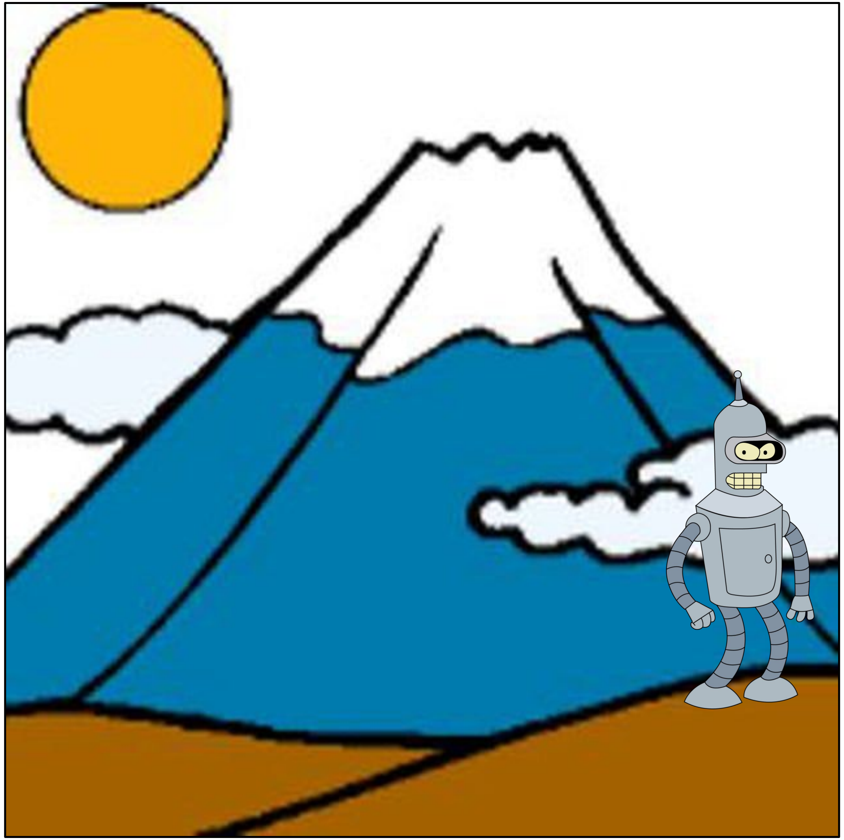
Движение!



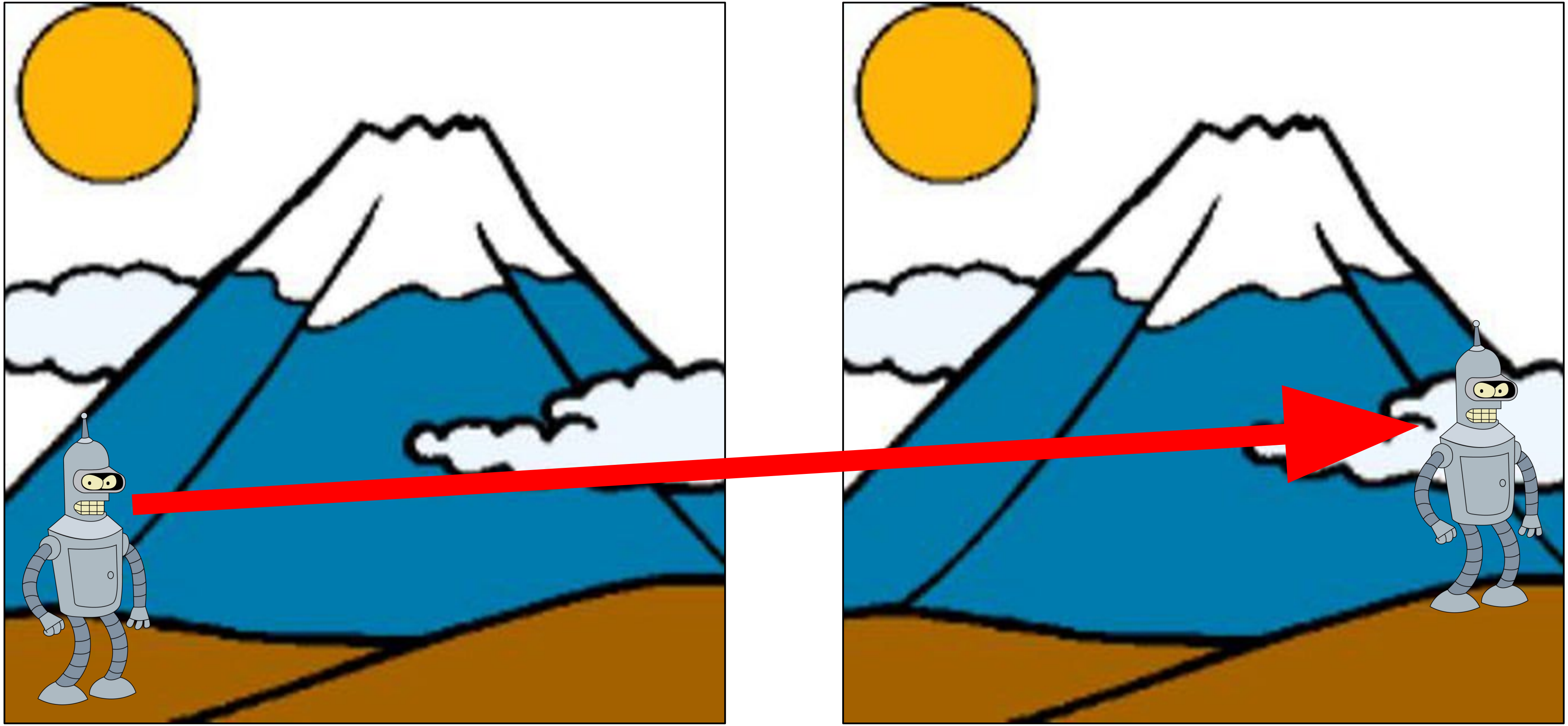
Что такое Optical Flow?

Движение!

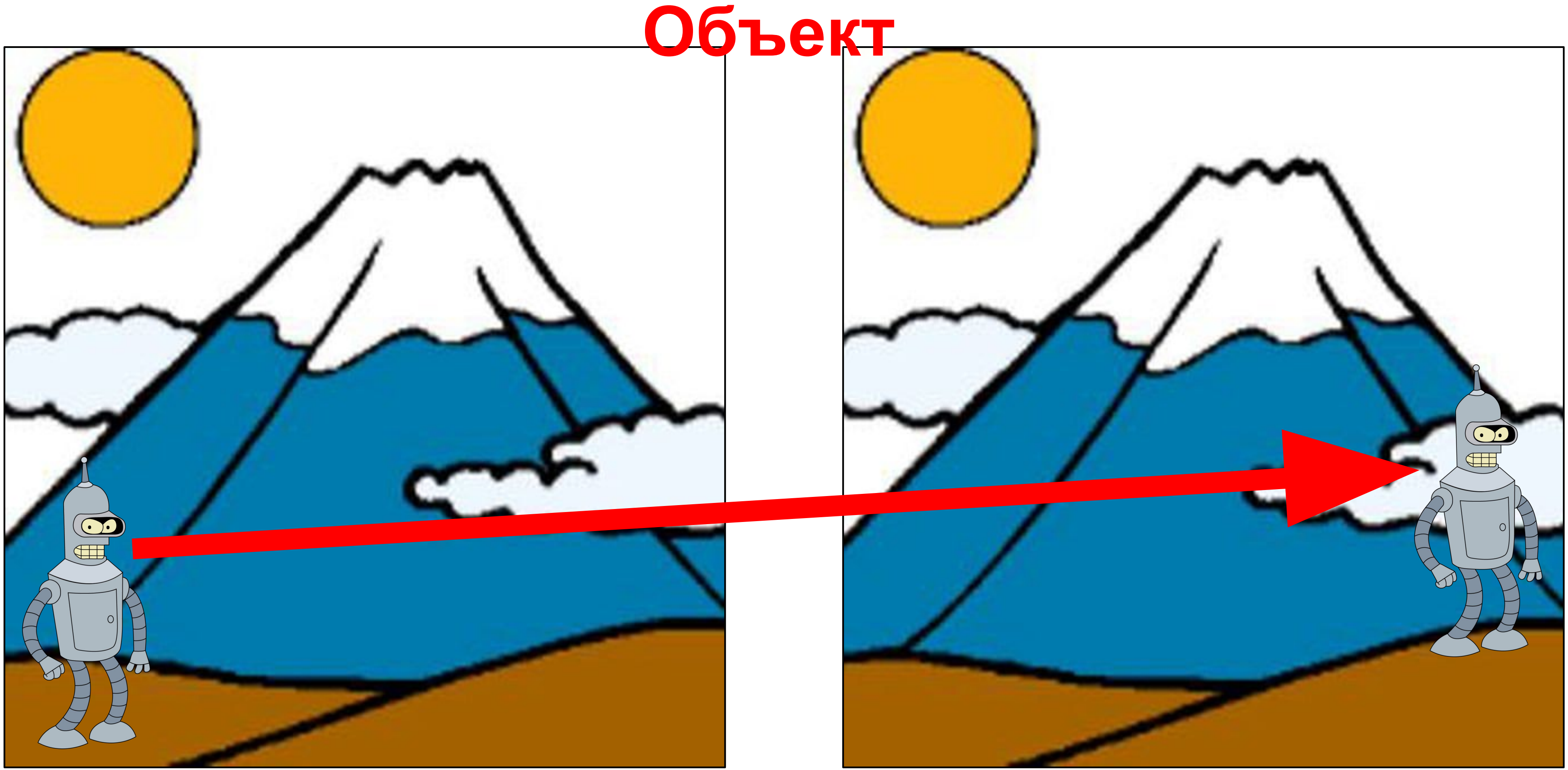




Что такое Optical Flow? **Движение!**



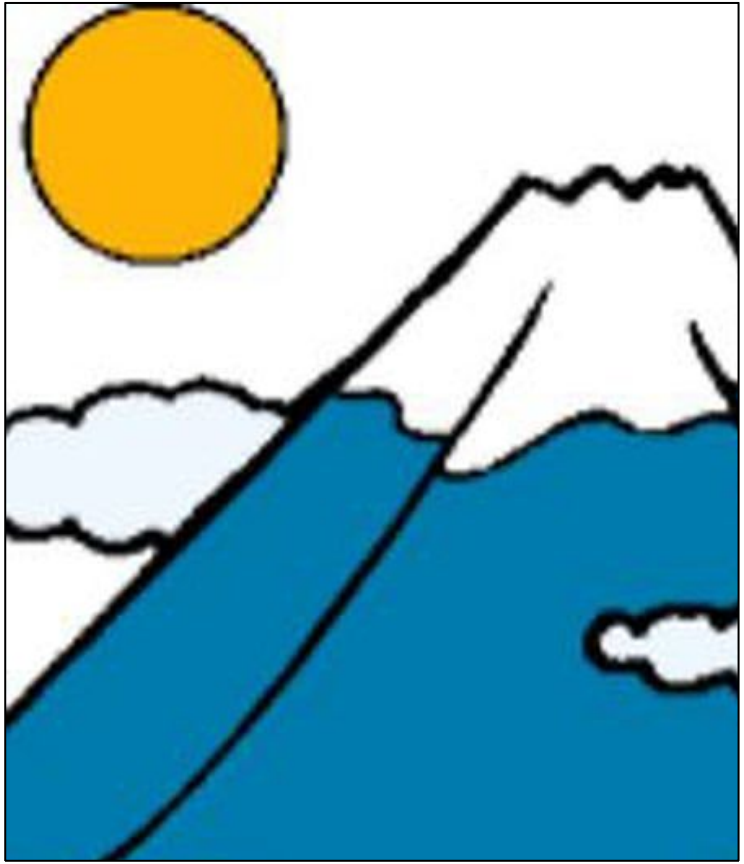
Что такое Optical Flow?

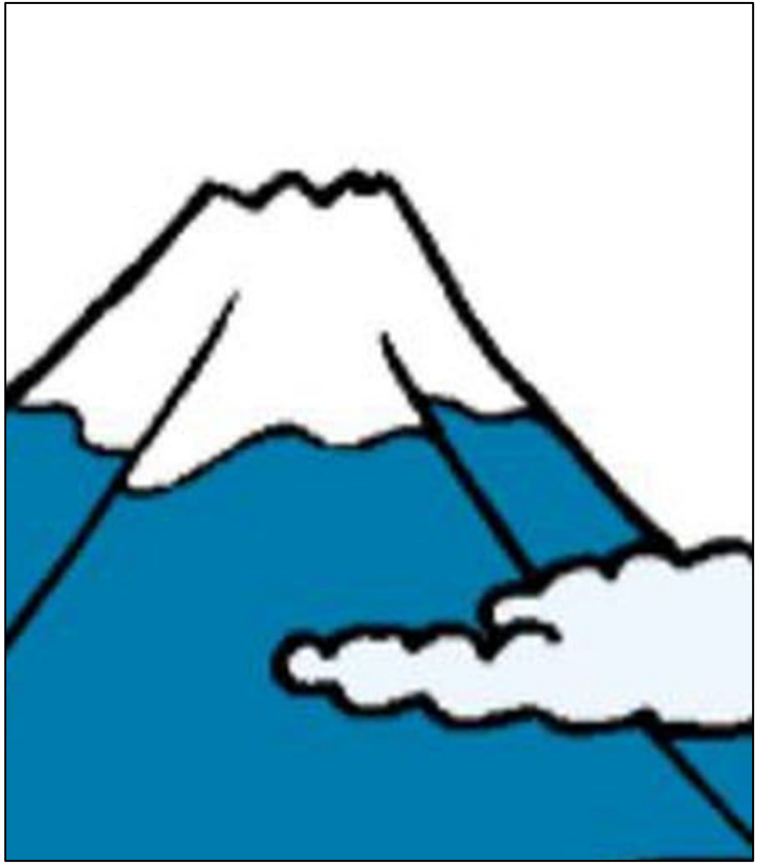


Что такое Optical Flow?

Движение

Сдвиг

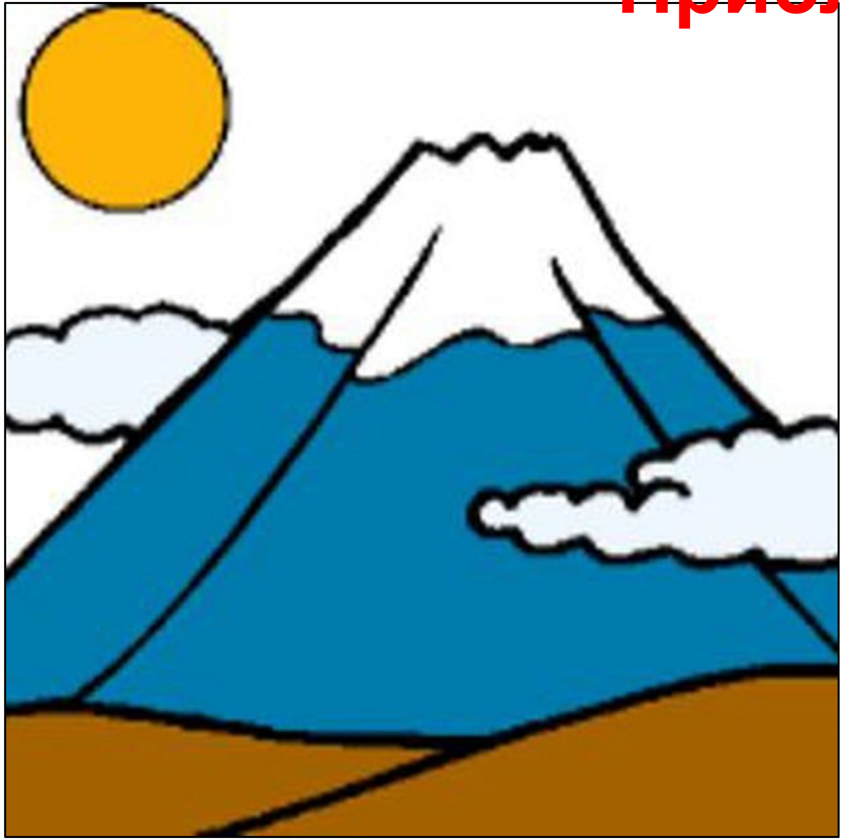


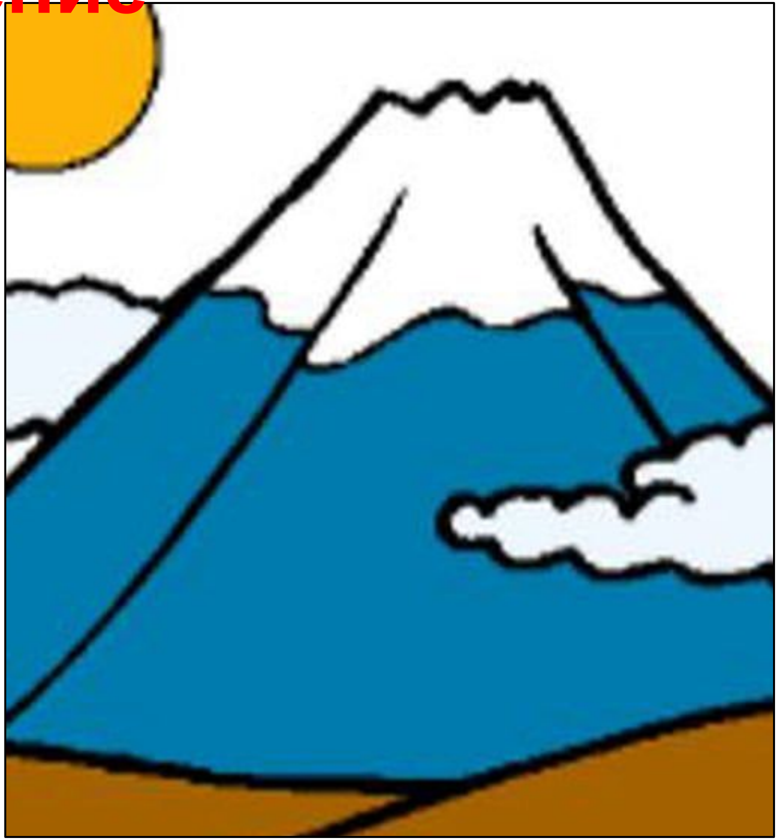


Что такое Optical Flow?

Движение

Приближение





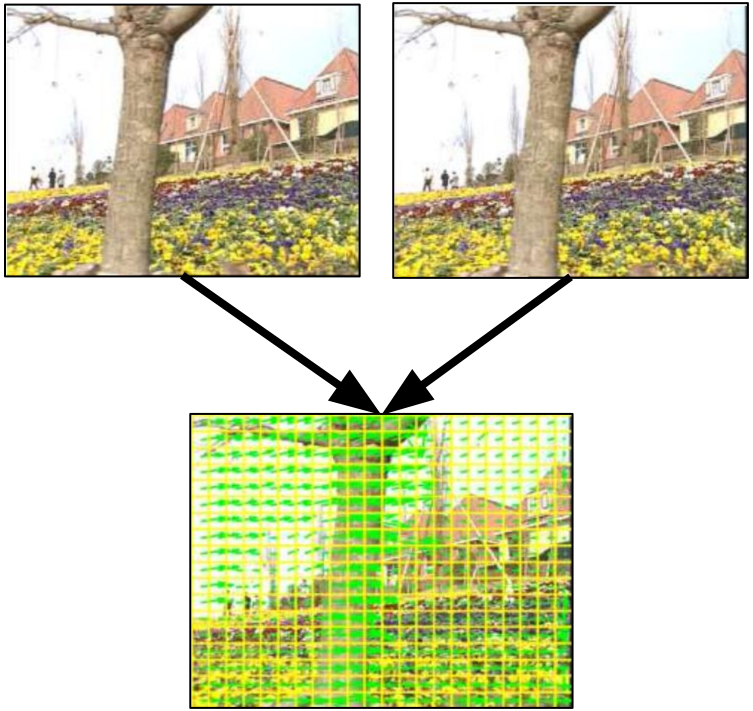
Что такое Optical Flow?

Движение

Зачем?

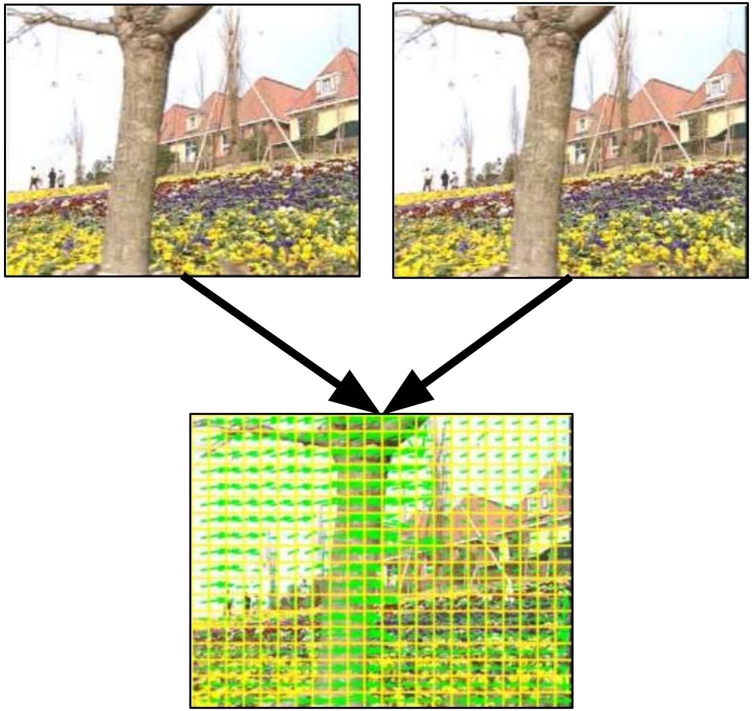
Зачем?

Motion Estimation



Зачем?

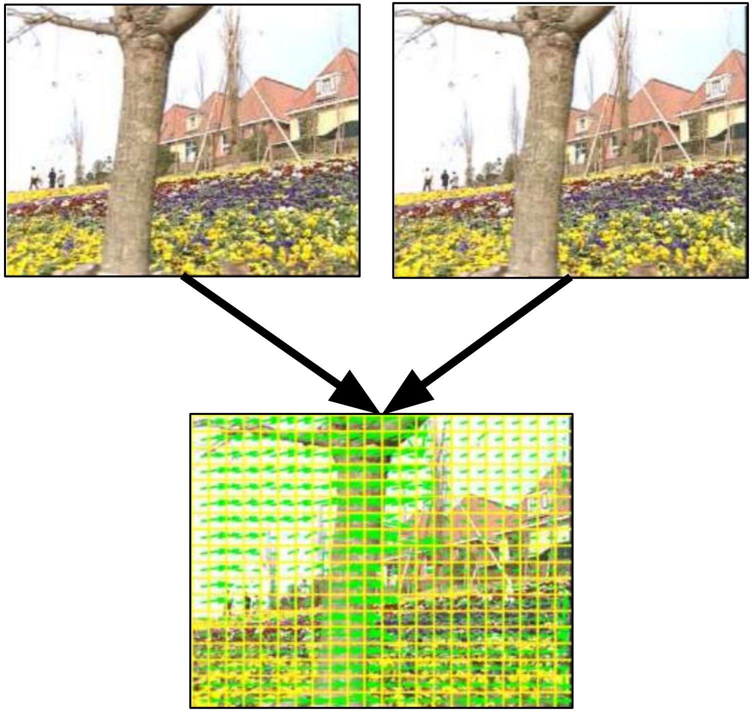
Motion Estimation



Object Tracking

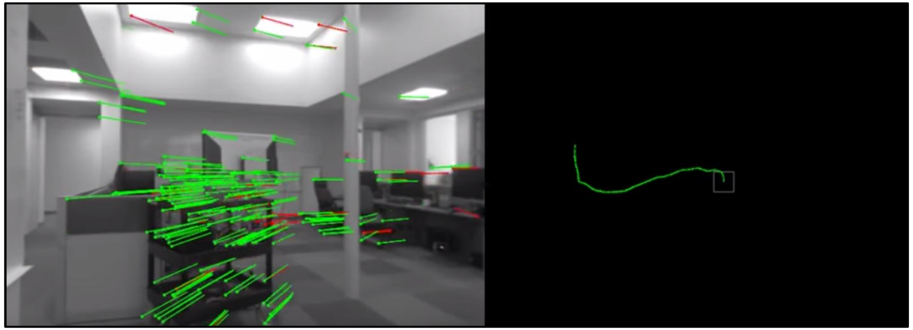
Зачем?

Motion Estimation



Object Tracking

Visual Odometry

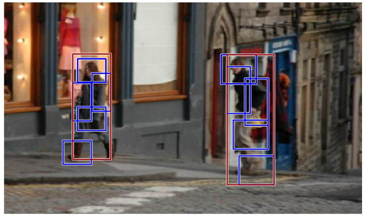


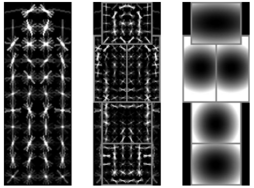
Как находить Optical flow?

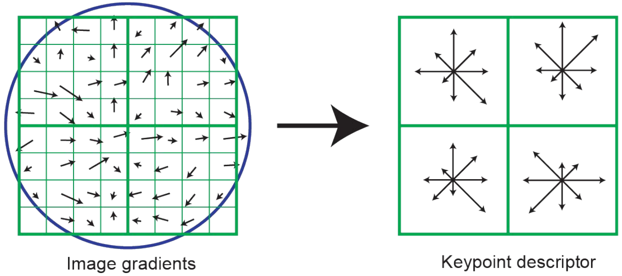
Feature Matching

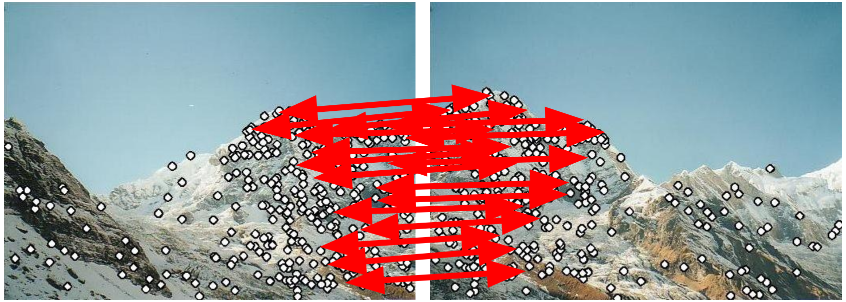
В предыдущих сериях: Features!

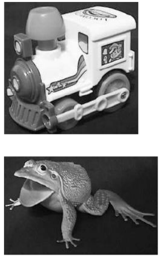
- Характерные регионы изображения
- Векторное описание этих регионов
- Используется в:
    - Matching
    - Recognition
    - Detection



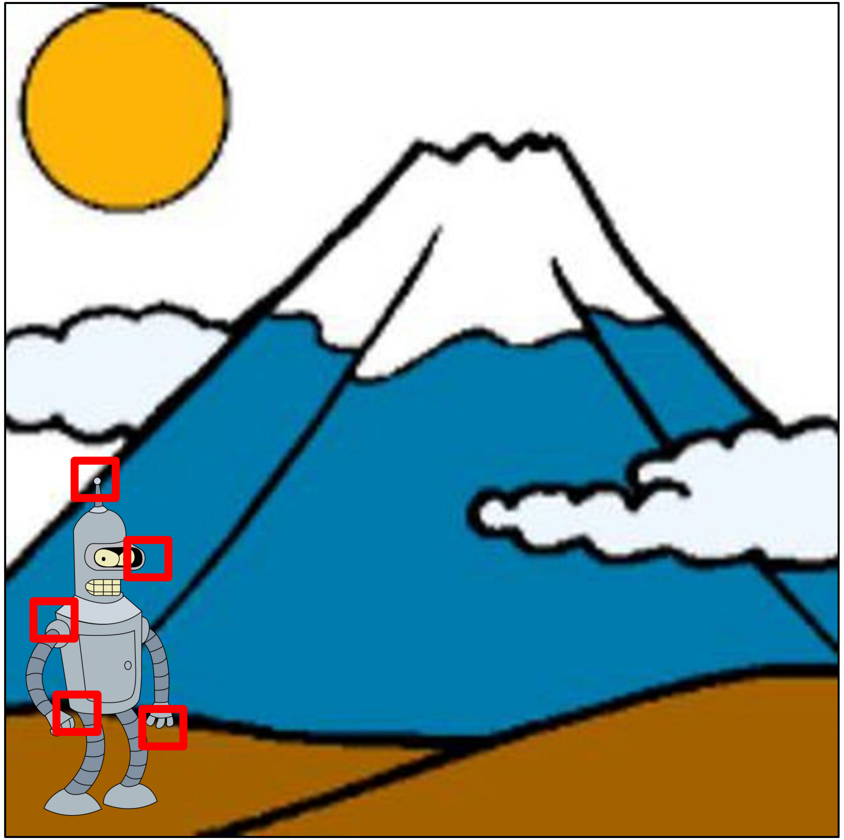


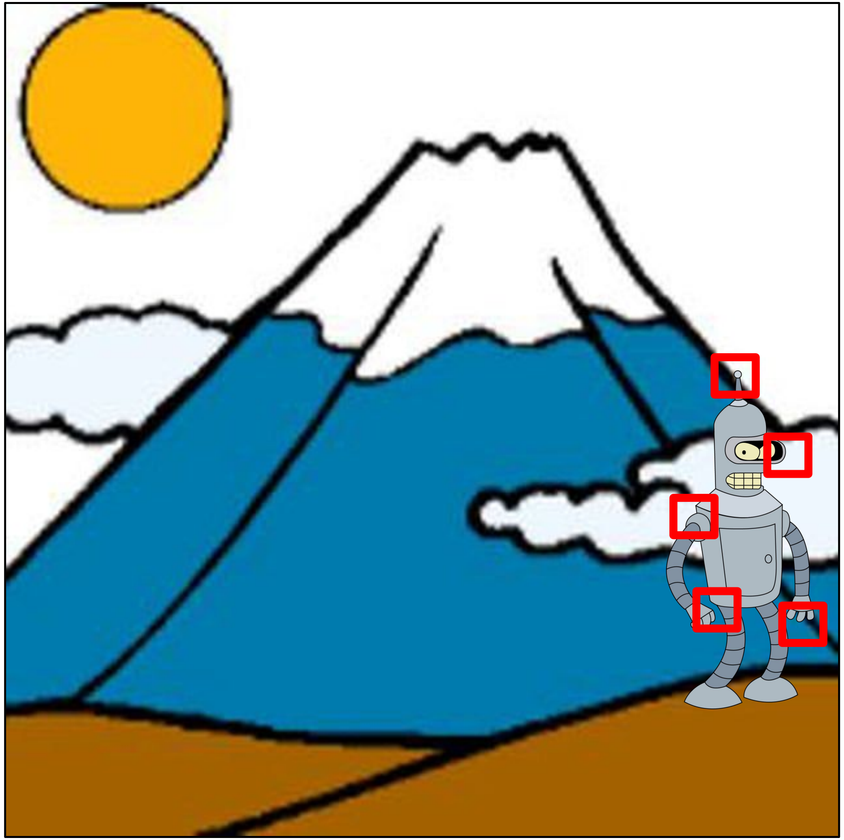




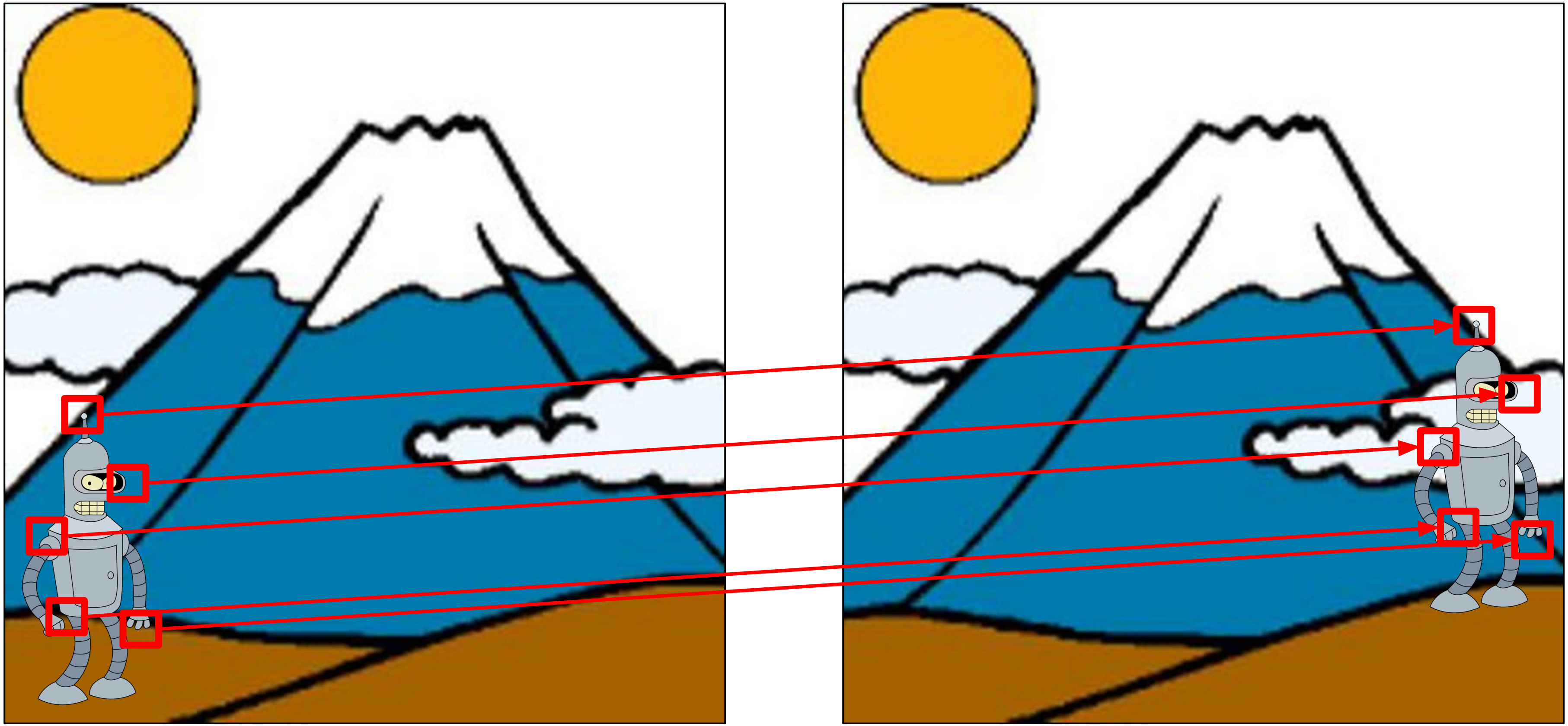


Feature Matching





Feature Matching



Feature Matching

Cons:

Feature Matching

Cons:

-Мы оцениваем движение для слишком маленького количества точек (sparse)!

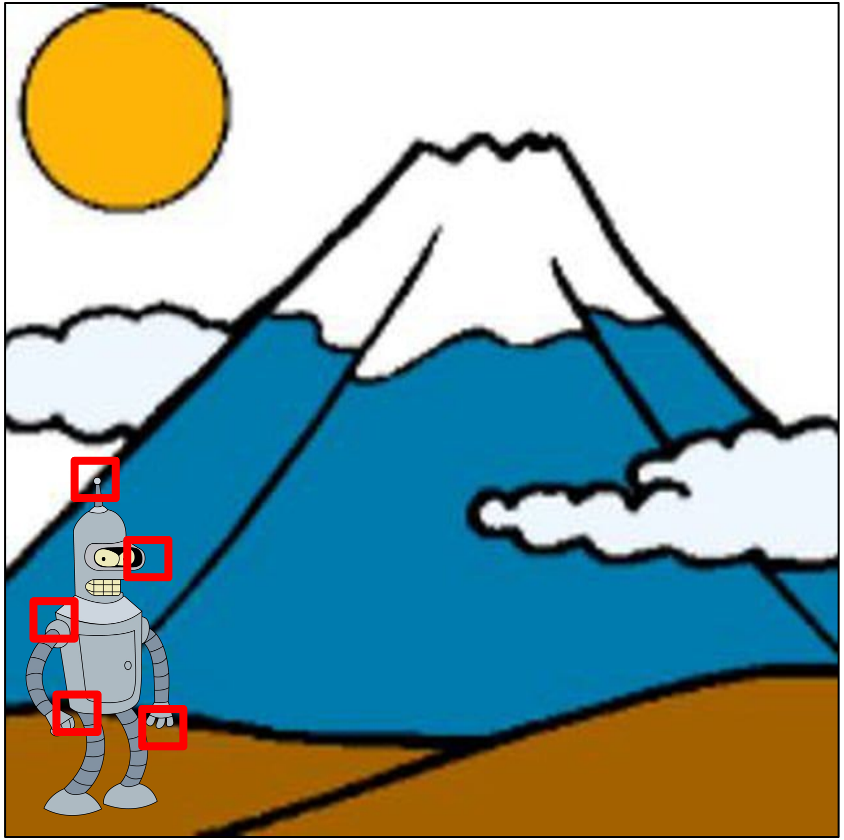
Feature Matching

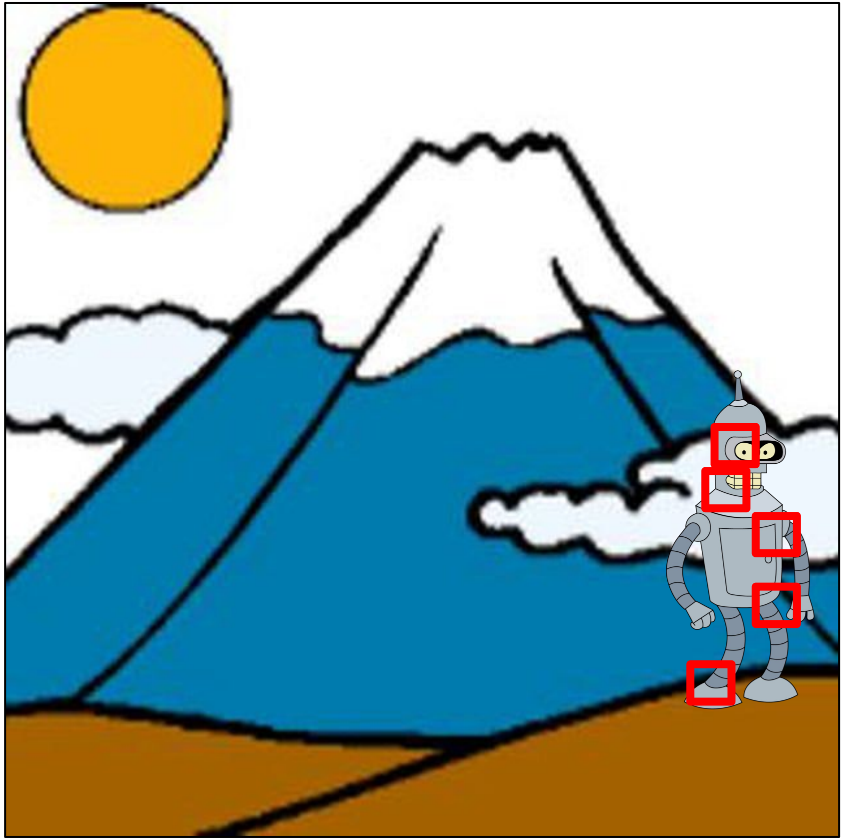
Cons:

-Мы оцениваем движение для слишком маленького количества точек (sparse)!

-Точки матчатся не точно

Feature Matching





Feature Matching

Cons:

-Мы оцениваем движение для слишком маленького количества точек (sparse)!

-Точки матчатся не точно

-Низкая точность

Feature Matching

Cons:

-Мы оцениваем движение для слишком маленького количества точек (sparse)!

-Точки матчатся не точно

-Низкая точность

Pros:

Feature Matching

Cons:

-Мы оцениваем движение для слишком маленького количества точек (sparse)!

-Точки матчатся не точно

-Низкая точность

Pros:

-Инварианты к поворотам и масштабированию

-Инвариант к изменению освещенности (почти)

-Может справляться с большими перемещениями

Feature Matching

Cons:

-Мы оцениваем движение для слишком маленького количества точ

-Точки матчатся

-Низкая точнос

Pros:

-Инварианты к поворотам и масштабированию

изменению

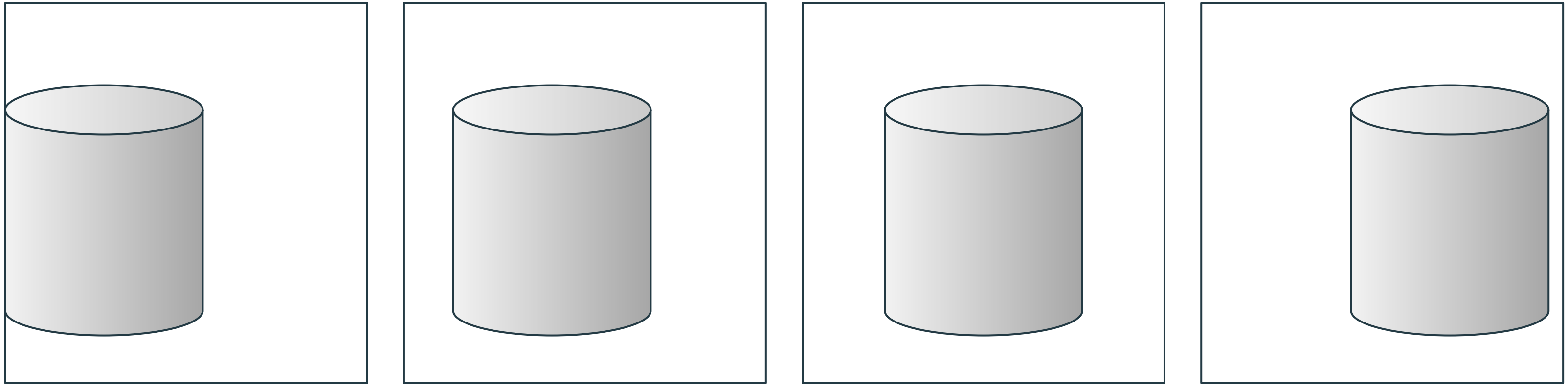
(почти)

яться с большими перемещаниями

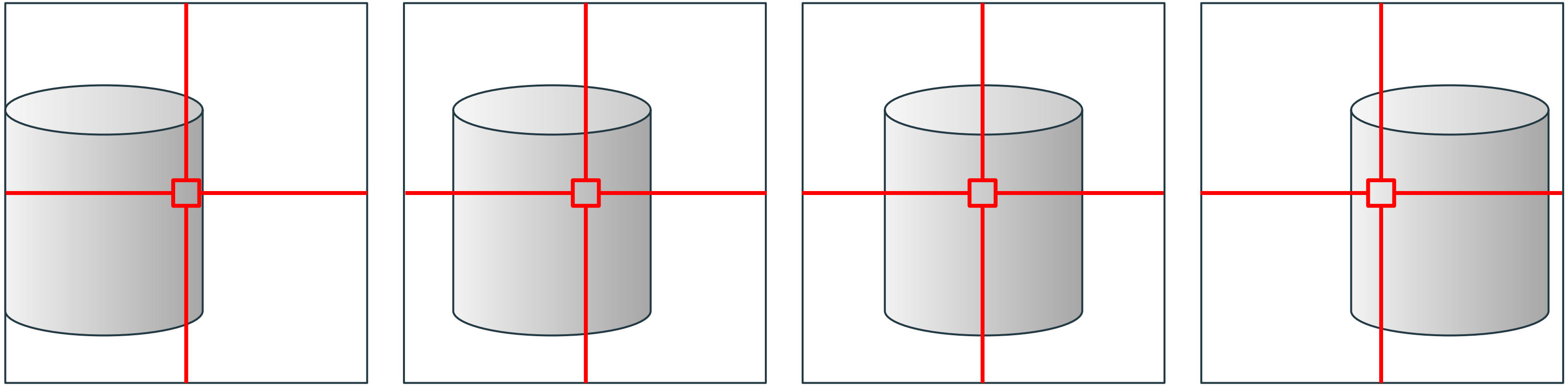
В общем случае плохо работает для задачи Optical Flow

Что делать?

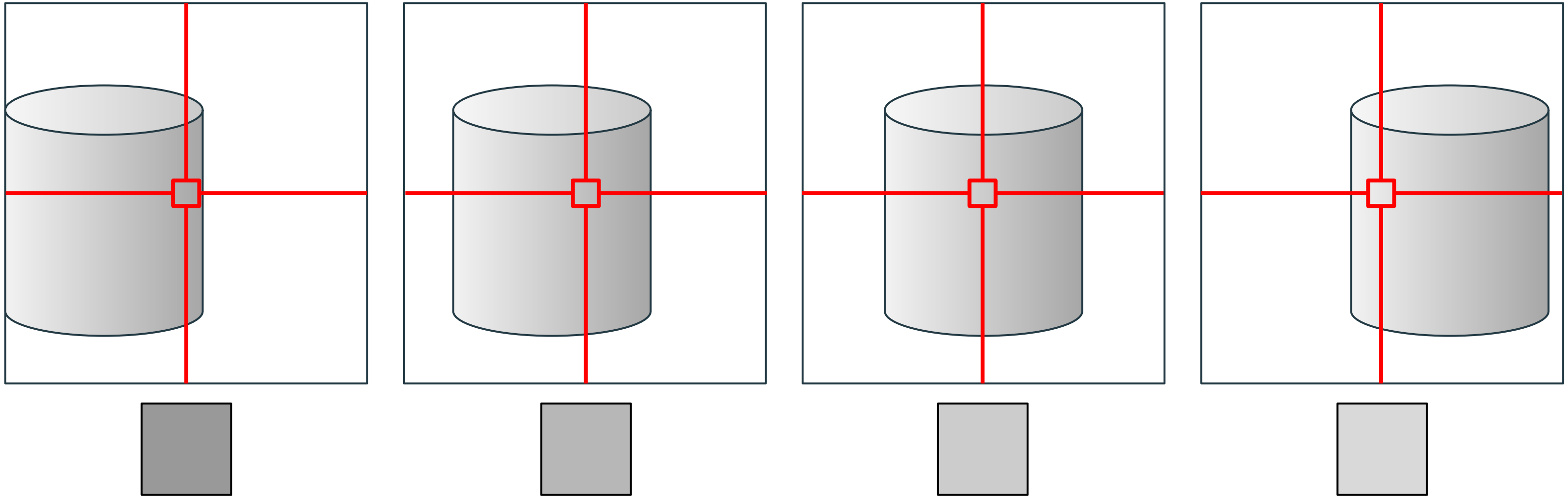
Наблюдение...



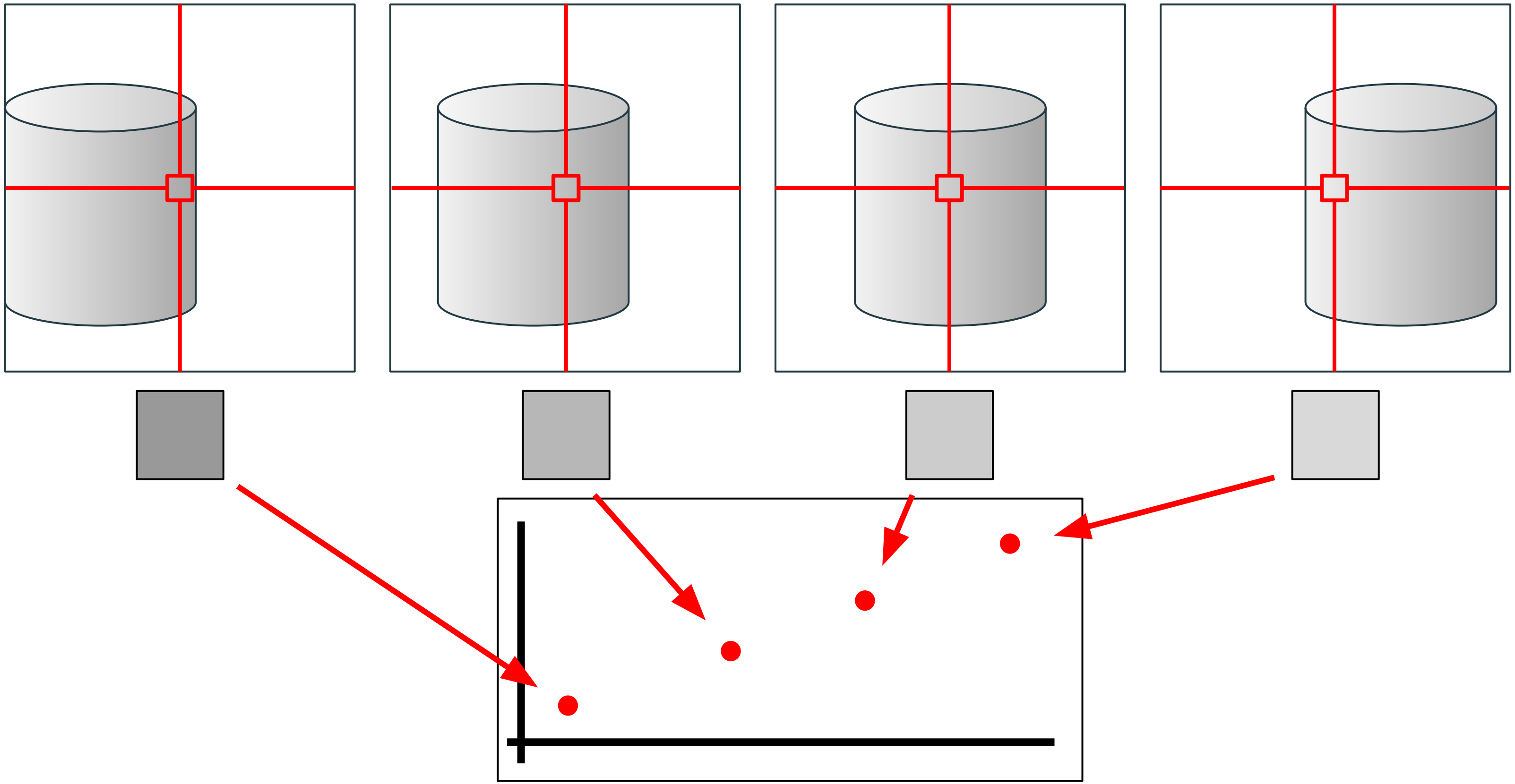
Наблюдение...



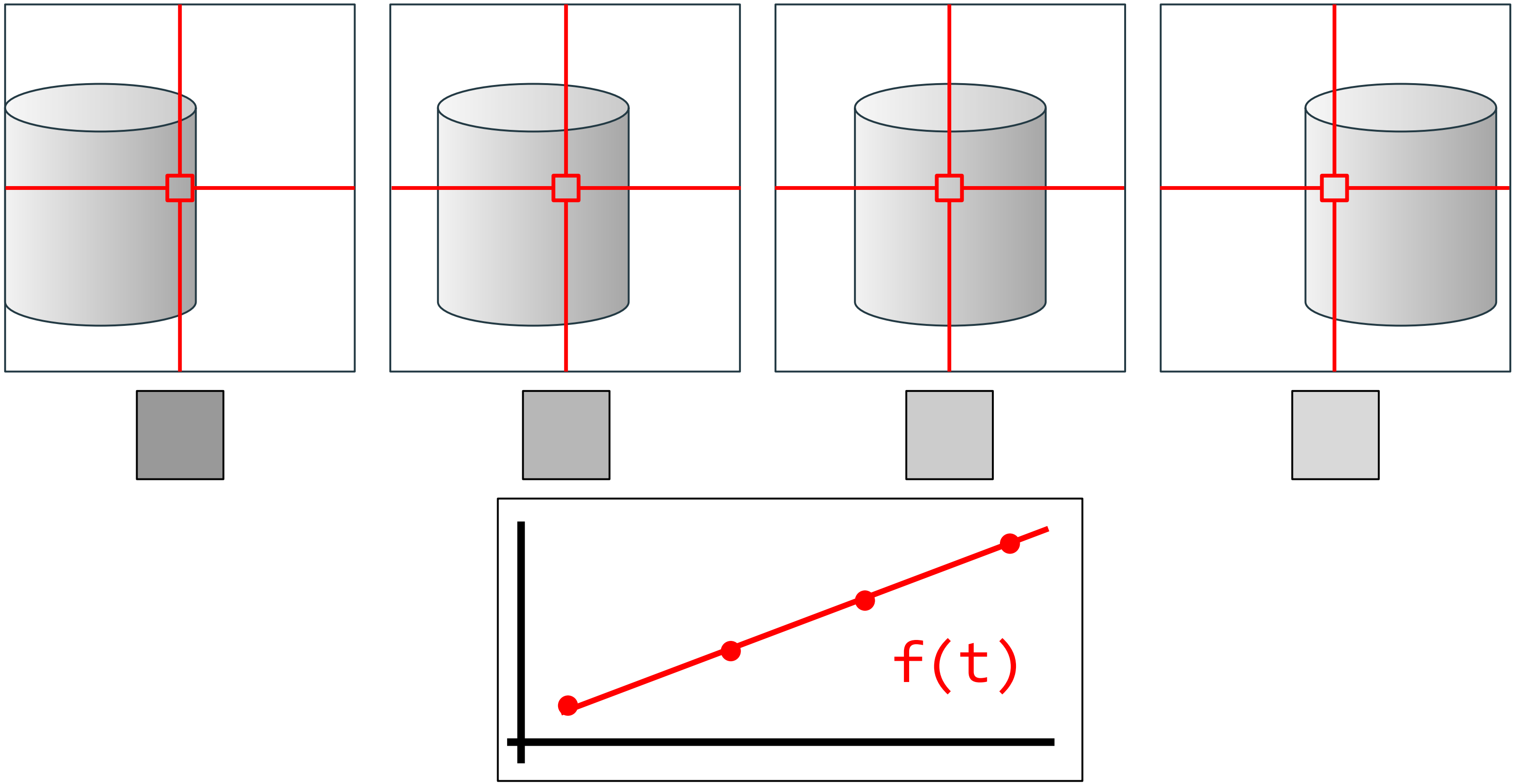
Наблюдение...



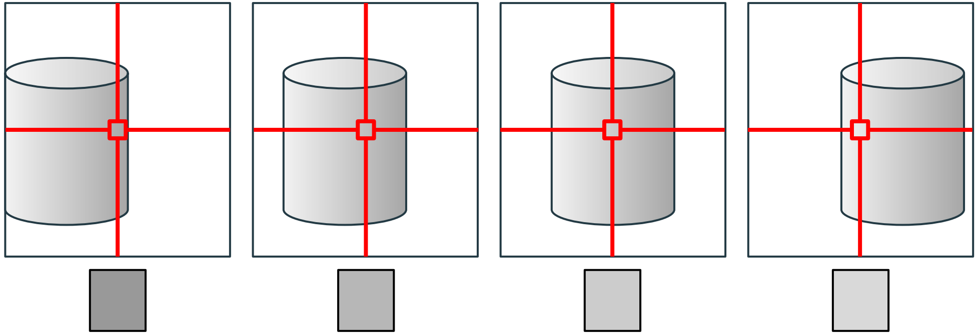
Наблюдение...

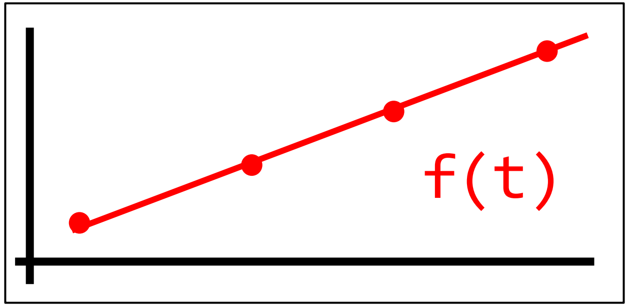


Наблюдение...

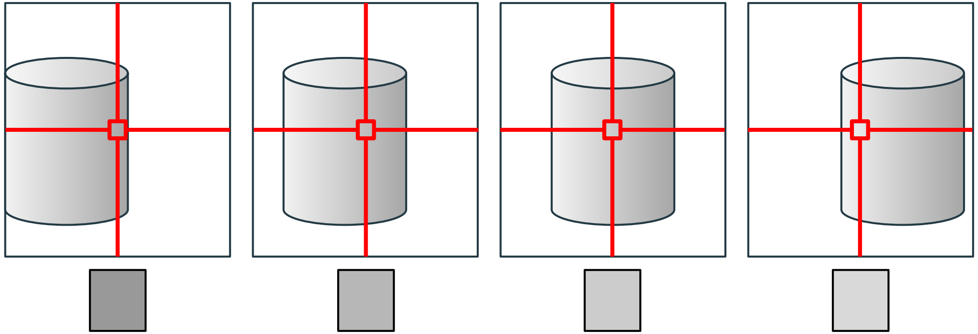


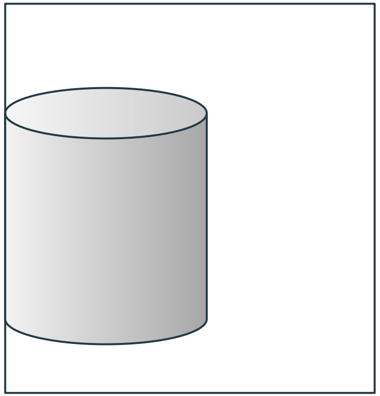
Наблюдение...

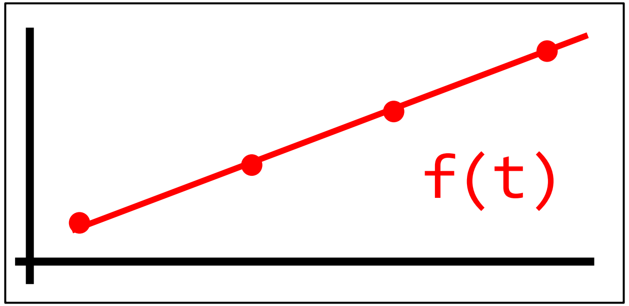




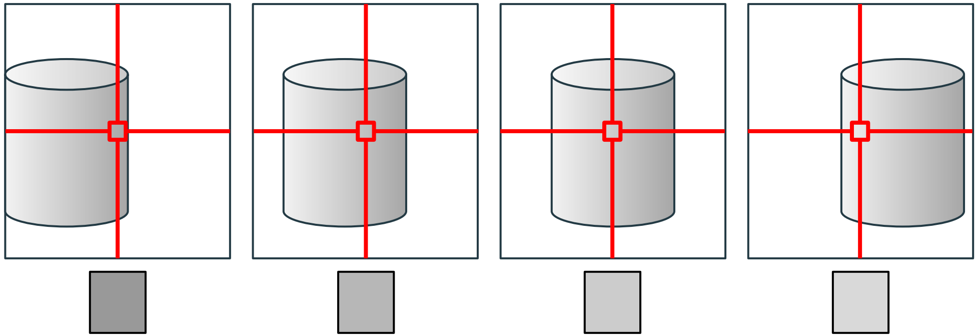
Наблюдение...

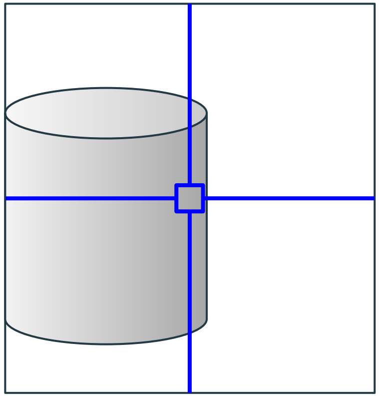


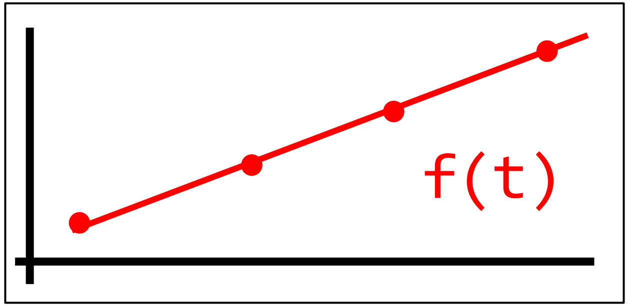




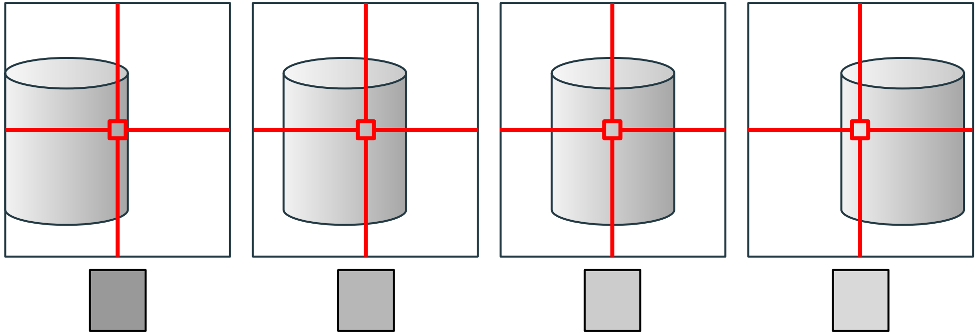
Наблюдение...

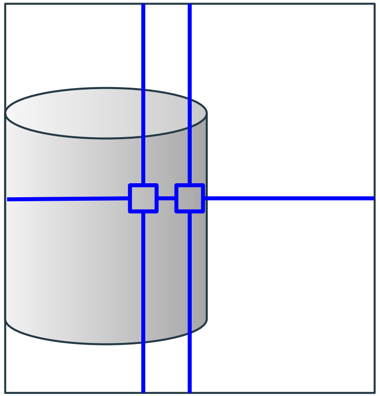


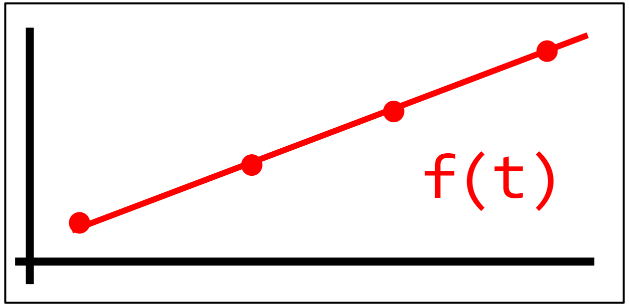




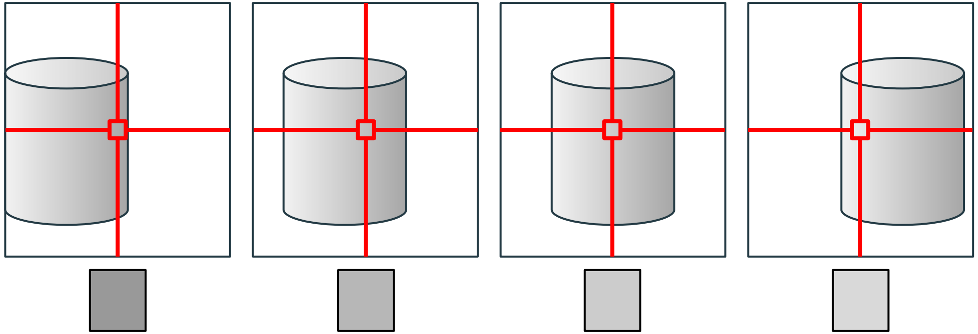
Наблюдение...

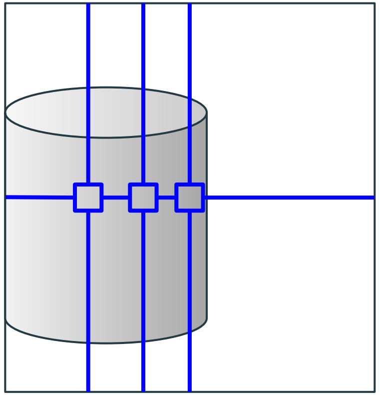


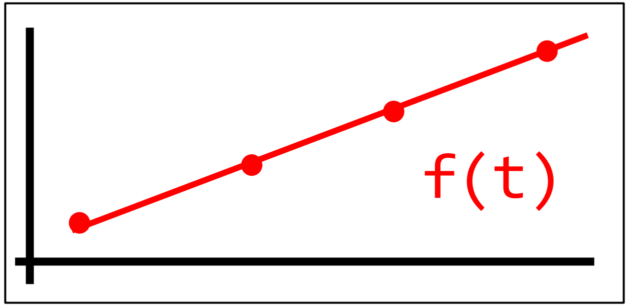




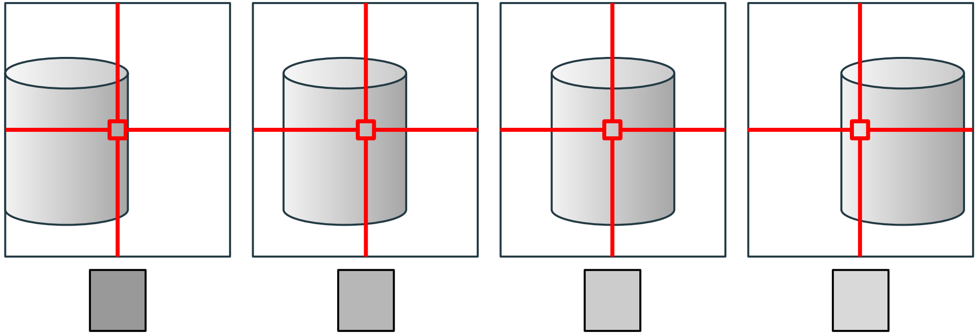
Наблюдение...

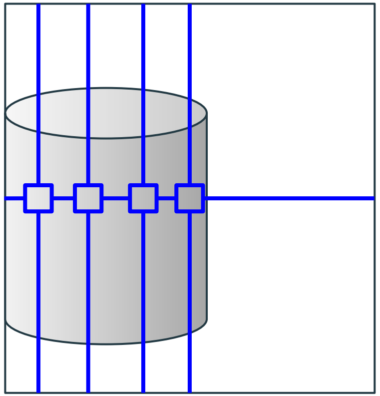


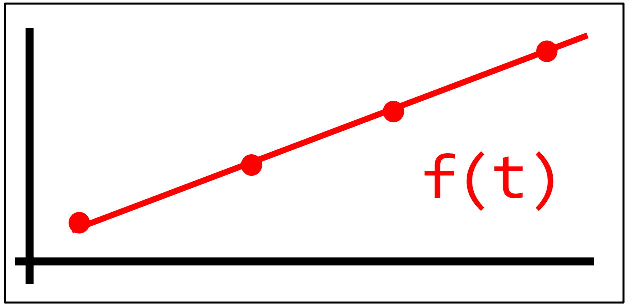




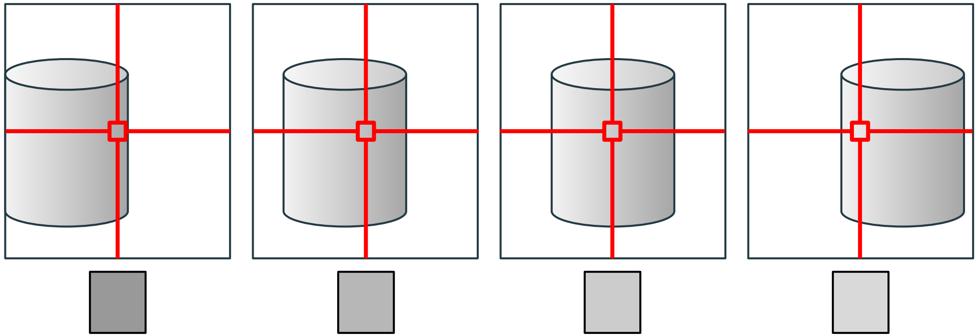
Наблюдение...

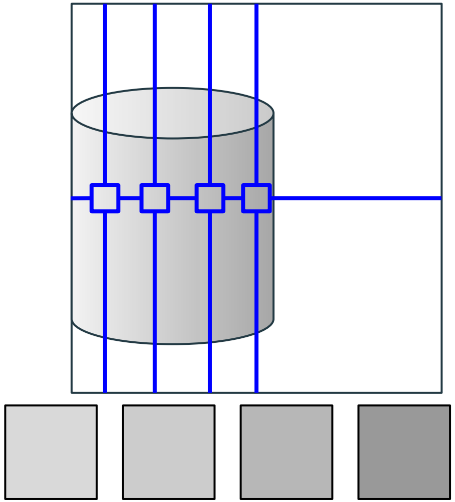


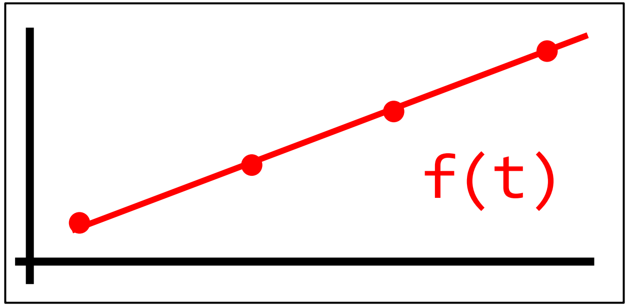




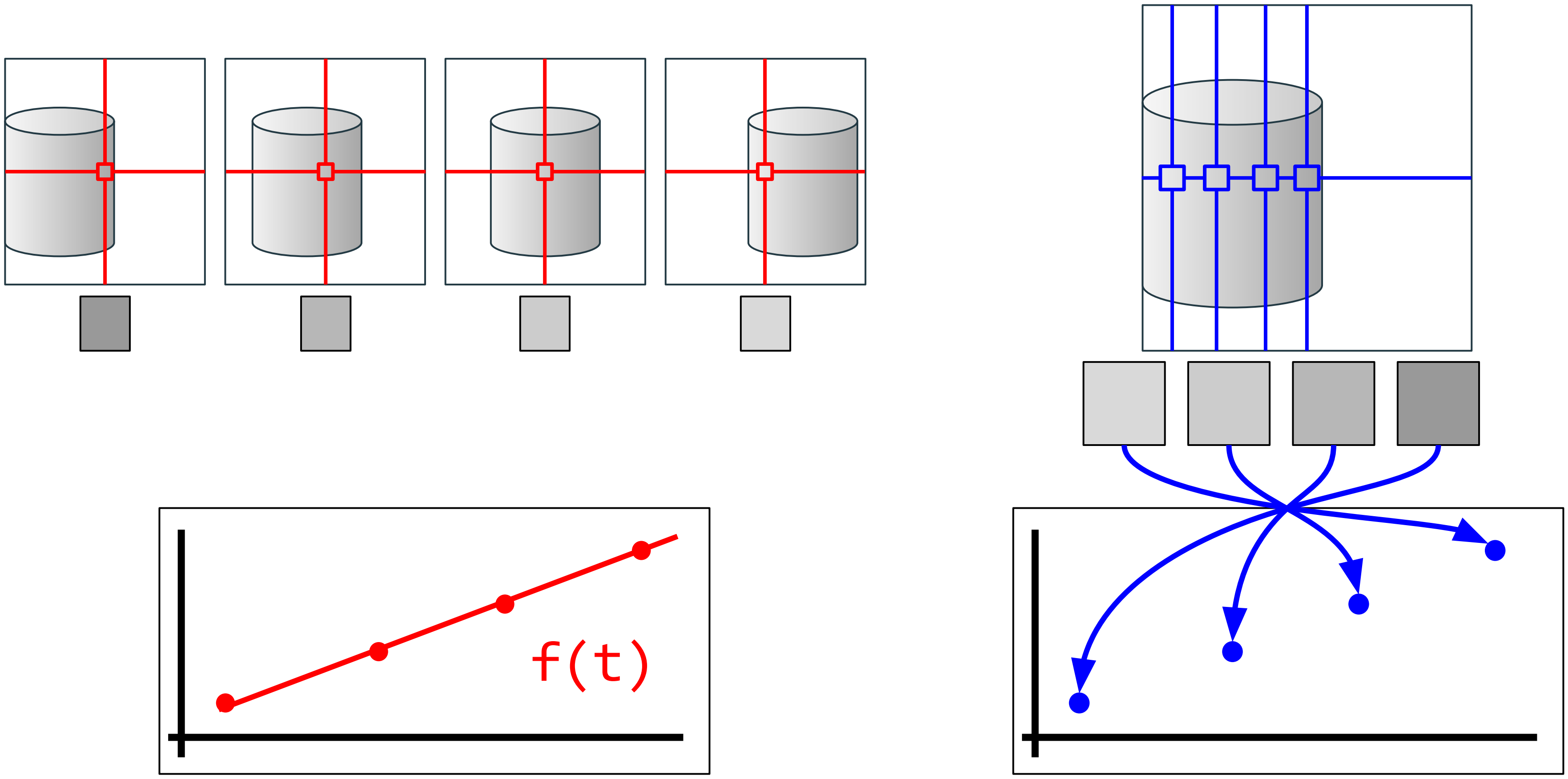
Наблюдение...



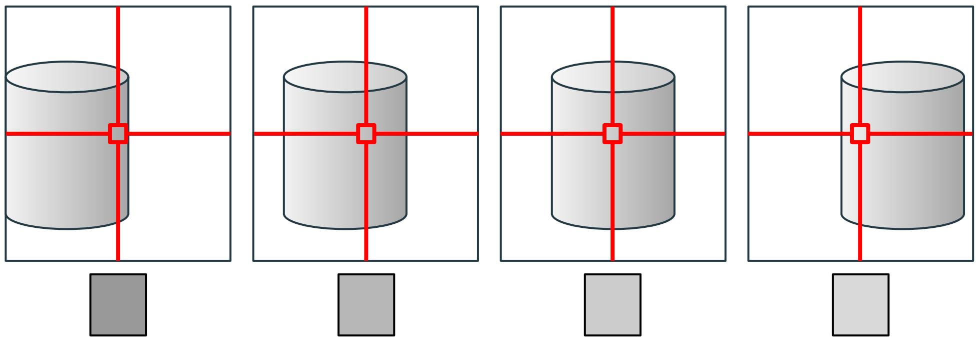


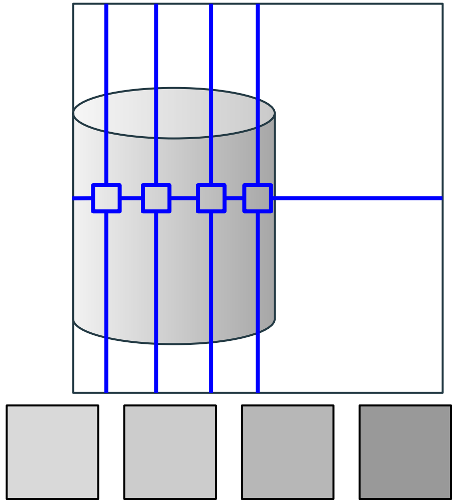


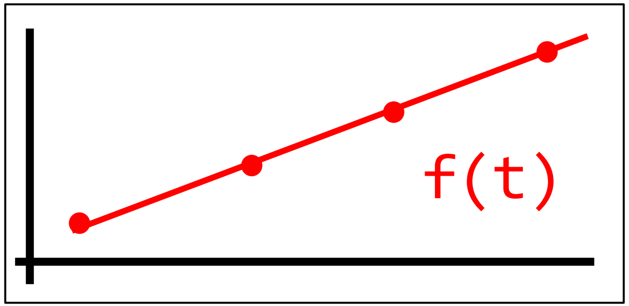
Наблюдение...

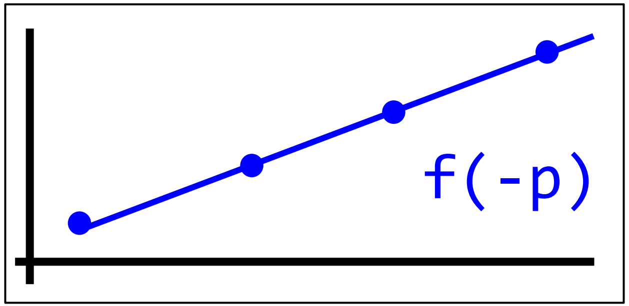


Наблюдение...

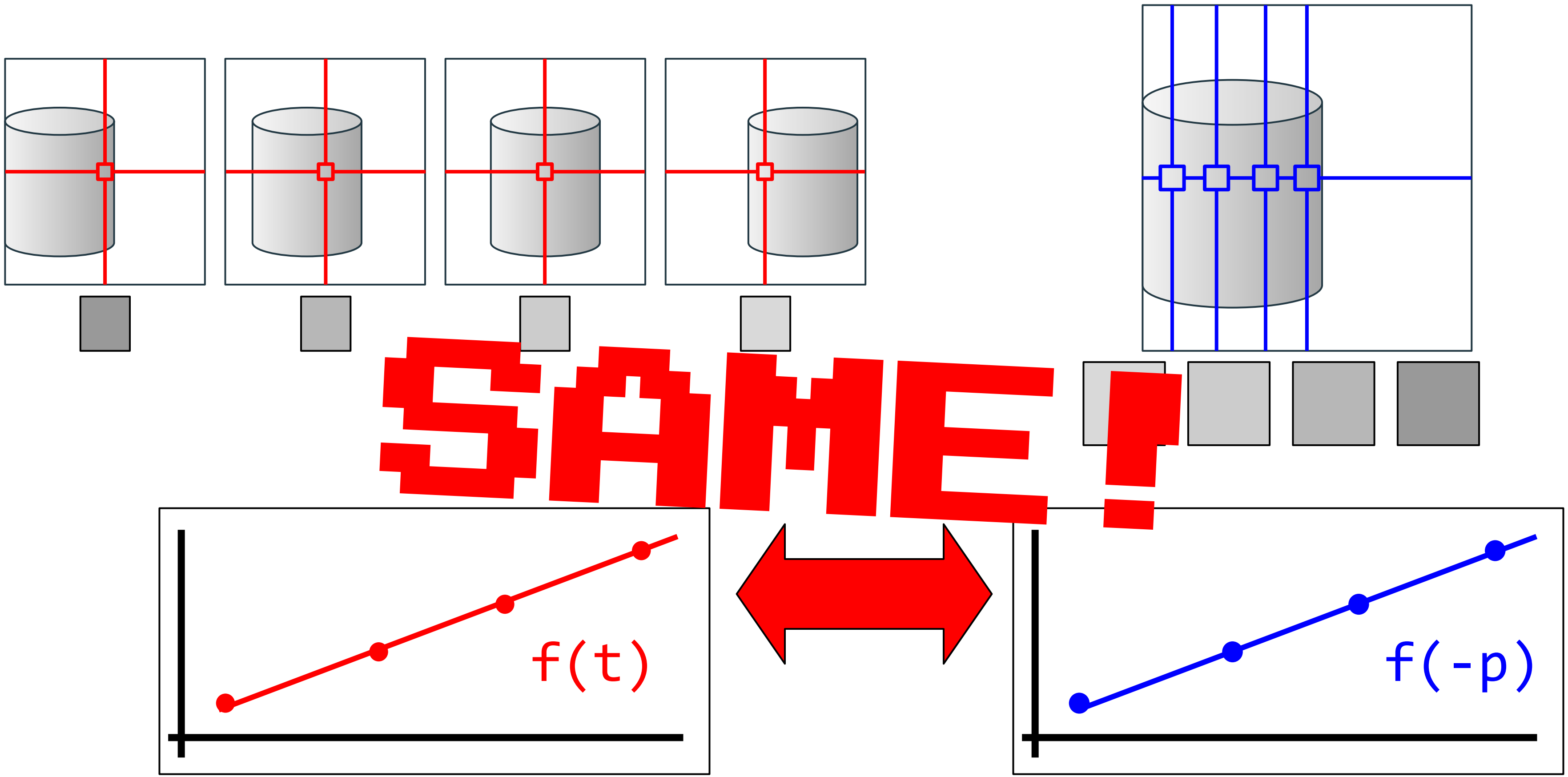




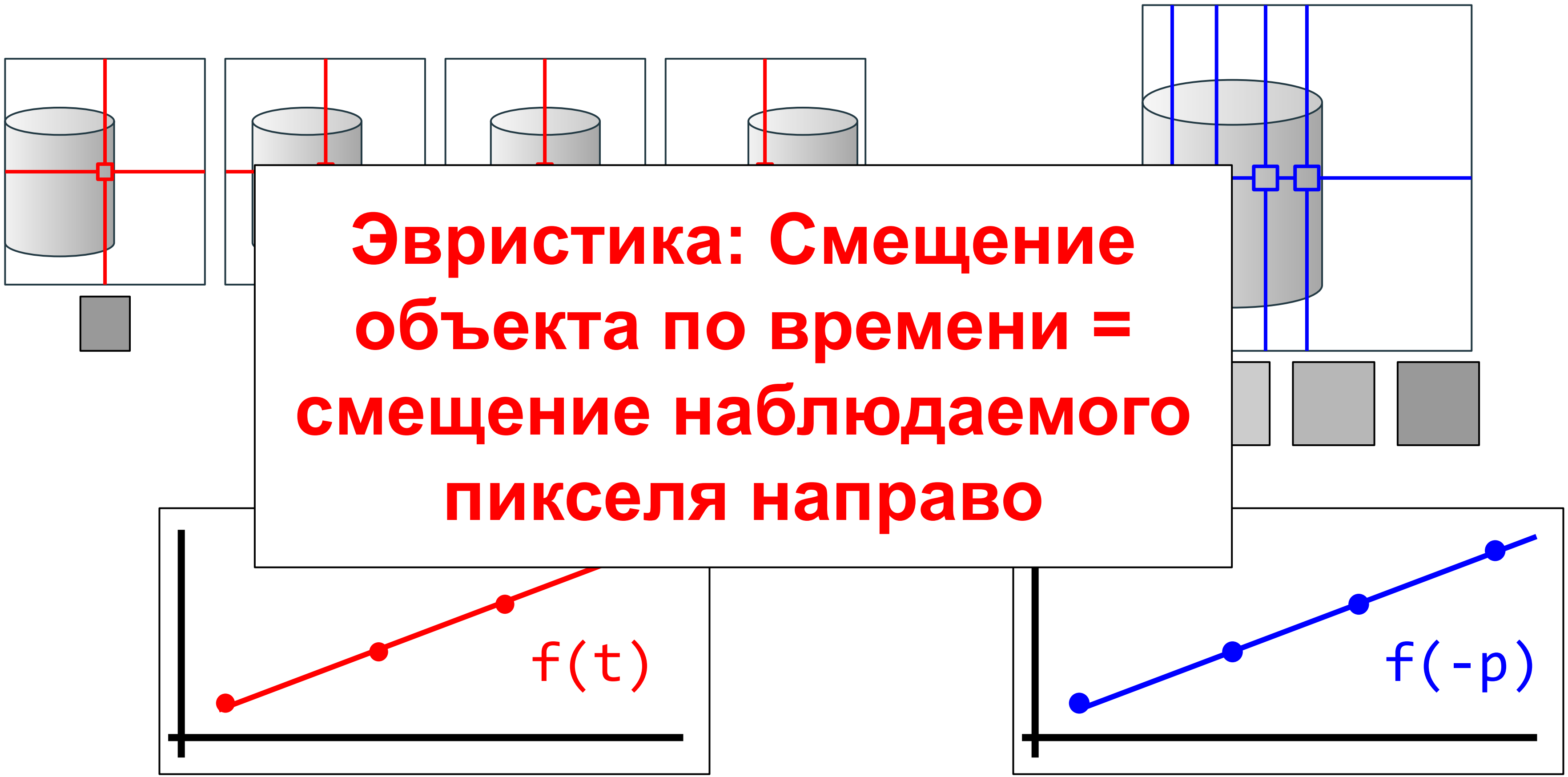




Наблюдение...



An observation...



Lucas-Kanade Optical Flow

Как использовать эту эвристику для расчета Optical flow?

Lucas-Kanade Optical Flow

Как использовать эту эвристику для расчета Optical flow?

B.D. Lucas, T. Kanade, “An Image Registration Technique with an Application to Stereo Vision”, in Proceedings of Image Understanding Workshop, 1981, pp. 121-130.

Lucas-Kanade Optical Flow

Предполагаем, что объект (пиксель) сместился на величину $\Delta\mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta\mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Lucas-Kanade Optical Flow

Предполагаем, что объект (пиксель) сместился на величину $\Delta \mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta \mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Наблюдаемый пиксель

Lucas-Kanade Optical Flow

Предполагаем, что объект (пиксель) сместился на величину $\Delta \mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta \mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Смещение

Lucas-Kanade Optical Flow

Предполагаем, что объект (пиксель) сместился на величину $\Delta \mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta \mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Первое изображение

Второе изображение

Lucas-Kanade Optical Flow

Предполагаем, что объект (пиксель) сместился на величину $\Delta\mathbf{p}$ за время Δt

$$\underbrace{f(\mathbf{p} - \Delta\mathbf{p}, t)}_{\text{Яркость}} = \underbrace{f(\mathbf{p}, t + \Delta t)}_{\text{Яркость}}$$

Lucas-Kanade Optical Flow

Предполагаем, что объект (пиксель) сместился на величину $\Delta\mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta\mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Аппроксимируем рядом Тейлора (до первого порядка)

Lucas-Kanade Optical Flow

Предполагаем, что объект (пиксель) сместился на величину $\Delta\mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta\mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Аппроксимируем рядом Тейлора (до первого порядка)

Что такое ряд Тейлора?

Ряд Тейлора

Описывает сложную функцию полиномом n-ого порядка

Ряд Тейлора

Описывает сложную функцию полиномом n-ого порядка

$$f(x) \approx \Sigma_i \, a_i x^i$$

Ряд Тейлора

Описывает сложную функцию полиномом n-ого порядка

$$f(x) \approx \textstyle\sum_i a_i x^i$$

Первый порядок: i=0,1

$$\approx a_1 x^1 + a_0 x^0$$

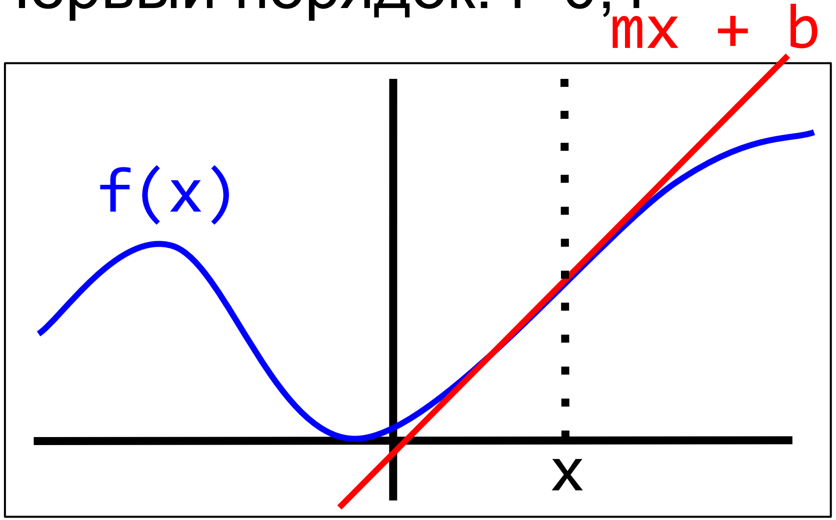
$$\approx mx + b$$

Quick & Dirty: Taylor Expansion

Описывает сложную функцию полиномом n-ого порядка

$$f(x) \approx \Sigma_i \, a_i x^i$$

Первый порядок: i=0,1



$$\approx a_1 x^1 + a_0 x^0$$

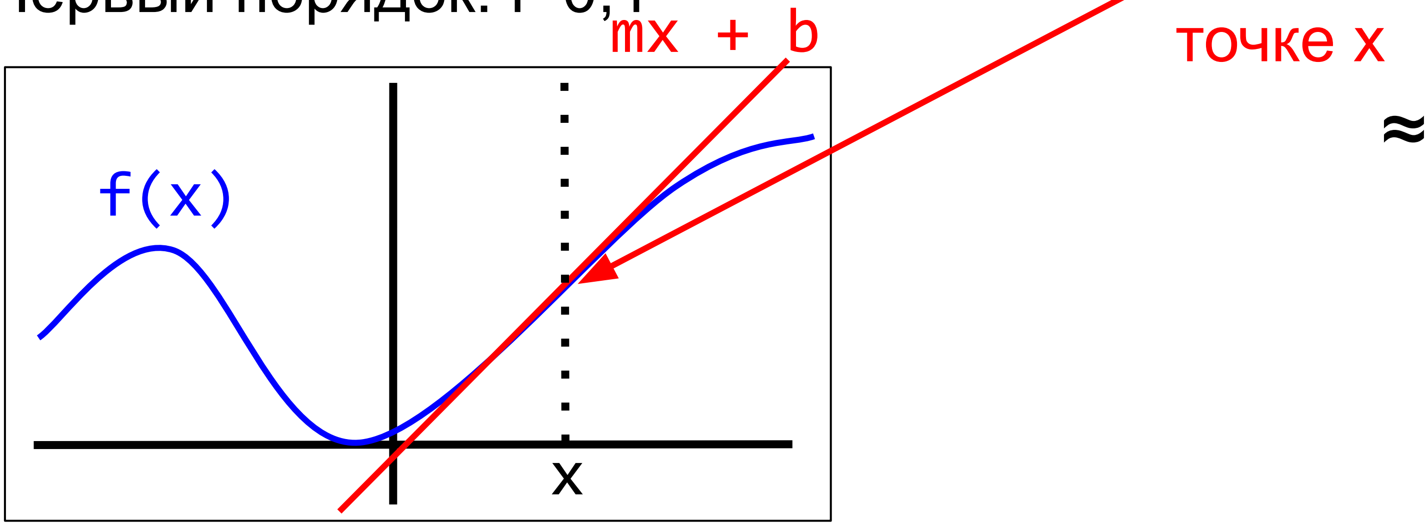
$$\approx mx + b$$

Quick & Dirty: Taylor Expansion

Описывает сложную функцию полиномом n-ого порядка

$$f(x) \approx \sum_i a_i x^i$$

Первый порядок: i=0,1



$$\approx a_1 x^1 + a_0 x^0$$

$$\approx mx + b$$

Lucas-Kanade Optical Flow

Предполагаем, что объект сместился на $\Delta\mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta\mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Раскладываем в ряд Тейлора до первого порядка

$$m(\mathbf{p} - \Delta\mathbf{p}) + b \approx f(\mathbf{p}, t + \Delta t)$$

Lucas-Kanade Optical Flow

Предполагаем, что объект сместился на $\Delta\mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta\mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Раскладываем в ряд Тейлора до первого порядка

$$m(\mathbf{p} - \Delta\mathbf{p}) + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Отнимем $f(\mathbf{p}, t)$

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(x, t) = mx + b$

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(x, t) = mx + b$

Подставляем `m`**`p`**` + b` вместо `f(`**`p`**`, t)`, и все сокращается!

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(x, t) = mx + b$

$$-m\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(x, t) = mx + b$

$$-m\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Это мы знаем

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(x, t) = mx + b$

$$-m\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Это то что мы хотим узнать

Это мы знаем

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(x, t) = mx + b$

А что это??

$$-m\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Это то что мы хотим узнать

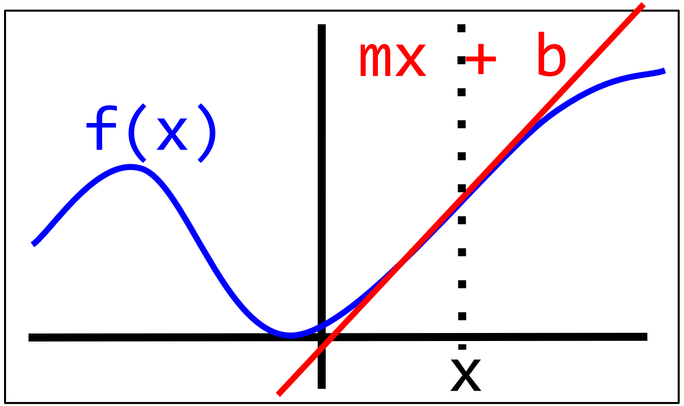
Это мы знаем

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(\mathbf{x}, t) = m\mathbf{x} + b$



$$-m\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

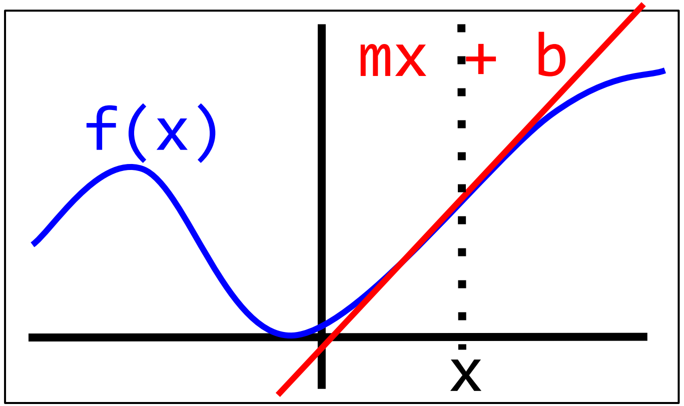
Наклон прямой

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(\mathbf{x}, t) = m\mathbf{x} + b$

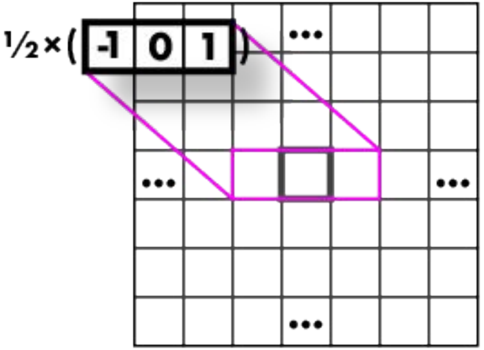


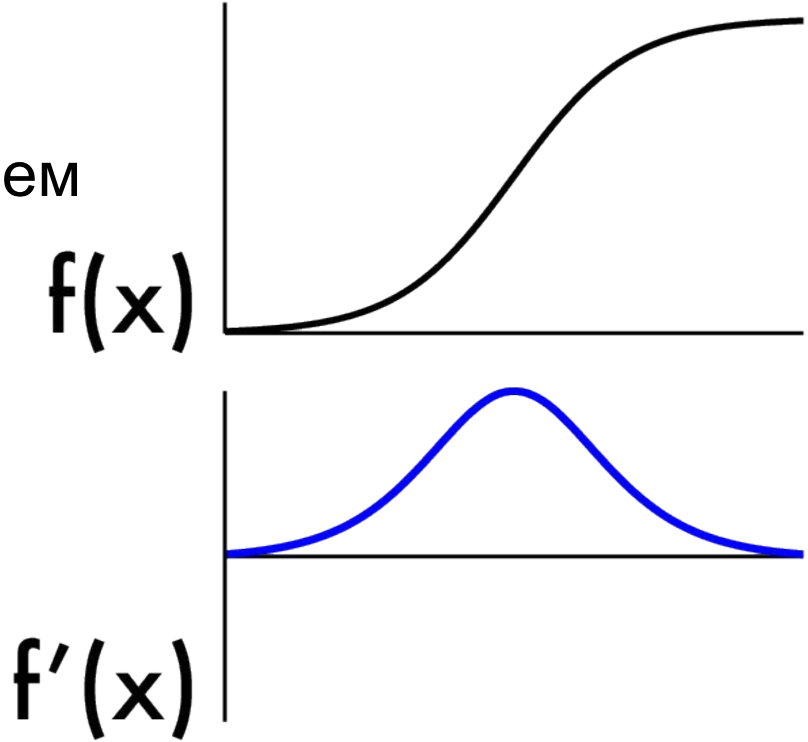
$$-m\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Наклон прямой = производная
= градиент

Вспомним: Image derivatives

- Вспомним:
  - $f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}.$
- У нас нет “настоящей” Функции -> численно оцениваем
- Давайте подставим h = 2
- Как это будет выглядеть?



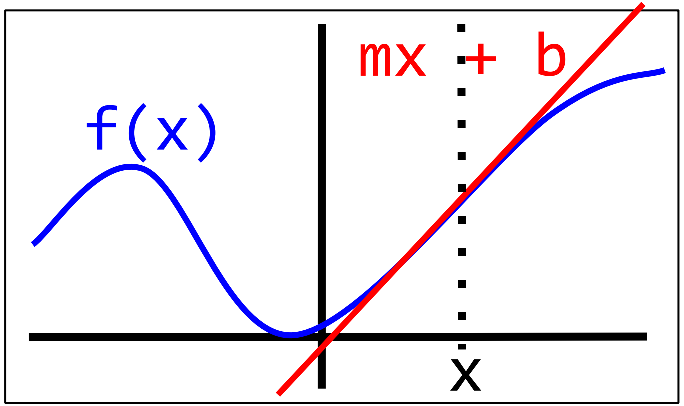


Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(\mathbf{x}, t) = m\mathbf{x} + b$



$$-m\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

$$m = d\mathbf{p} = \begin{bmatrix} dx \\ dy \end{bmatrix}$$

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(\mathbf{x}, t) = m\mathbf{x} + b$

$$-m\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

$$-d\mathbf{p}\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(\mathbf{x}, t) = m\mathbf{x} + b$

$$-m\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

$$-d\mathbf{p}\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Избавимся от векторной нотации

Lucas-Kanade Optical Flow

$$m\mathbf{p} - m\Delta\mathbf{p} + b \approx f(\mathbf{p}, t + \Delta t)$$

$$m\mathbf{p} - m\Delta\mathbf{p} + b - f(\mathbf{p}, t) \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

Вспомним, что $f(\mathbf{x}, t) = m\mathbf{x} + b$

$$-m\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

$$-d\mathbf{p}\Delta\mathbf{p} \approx f(\mathbf{p}, t + \Delta t) - f(\mathbf{p}, t)$$

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Lucas-Kanade Optical Flow

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Lucas-Kanade Optical Flow

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Немножко изменим обозначения:

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Lucas-Kanade Optical Flow

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Немножко изменим обозначения:

$$dx * u + dy * v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Lucas-Kanade Optical Flow

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Немножко изменим обозначения:

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Lucas-Kanade Optical Flow

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Немножко изменим обозначения:

$$dx * u + dy * v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Lucas-Kanade Optical Flow

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Немножко изменим обозначения:

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Lucas-Kanade Optical Flow

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Немножко изменим обозначения:

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Lucas-Kanade Optical Flow

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Немножко изменим обозначения:

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Решим относительно u,v

Lucas-Kanade Optical Flow

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Немножко изменим обозначения:

$$dx * u + dy * v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Решим относительно u,v

1 уравнение, 2 неизвестных

Lucas-Kanade Optical Flow

$$-dx\Delta x + -dy\Delta y \approx f((x,y), t + \Delta t) - f((x,y), t)$$

Немножко изменим обозначения:

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Решим относительно u,v

1 уравнение, 2 неизвестных

:(

Lucas-Kanade Optical Flow

Вернемся назад

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Lucas-Kanade Optical Flow

Вернемся назад

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

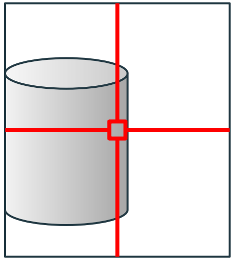
В чем смысл этого выражения?

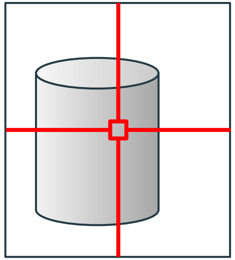
Lucas-Kanade Optical Flow

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Lucas-Kanade Optical Flow

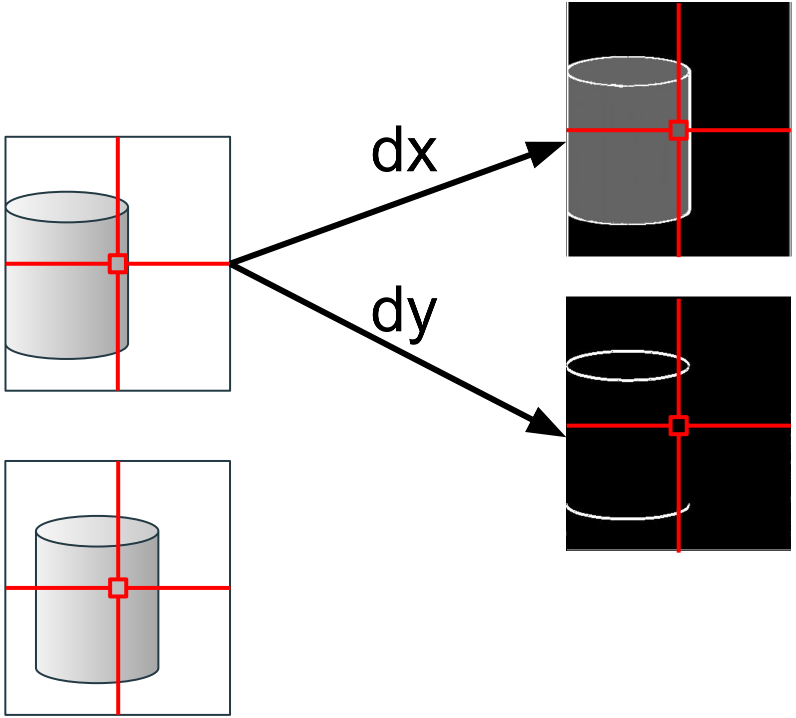
$$dx * u + dy * v = I_t[x,y] - I_{t+\Delta t}[x,y]$$





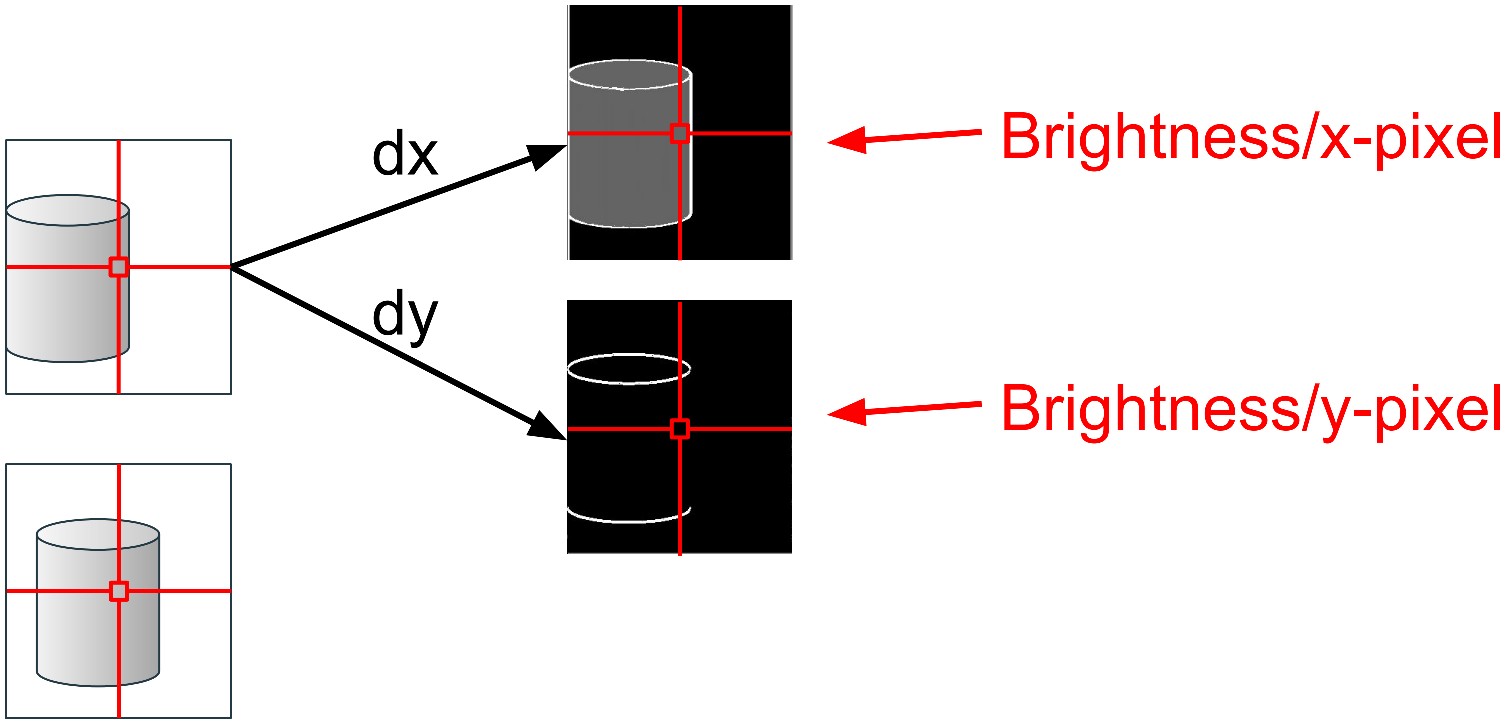
Lucas-Kanade Optical Flow

$$dx*\color{red}{u} + dy*\color{red}{v} = I_t[x,y] - I_{t+\Delta t}[x,y]$$



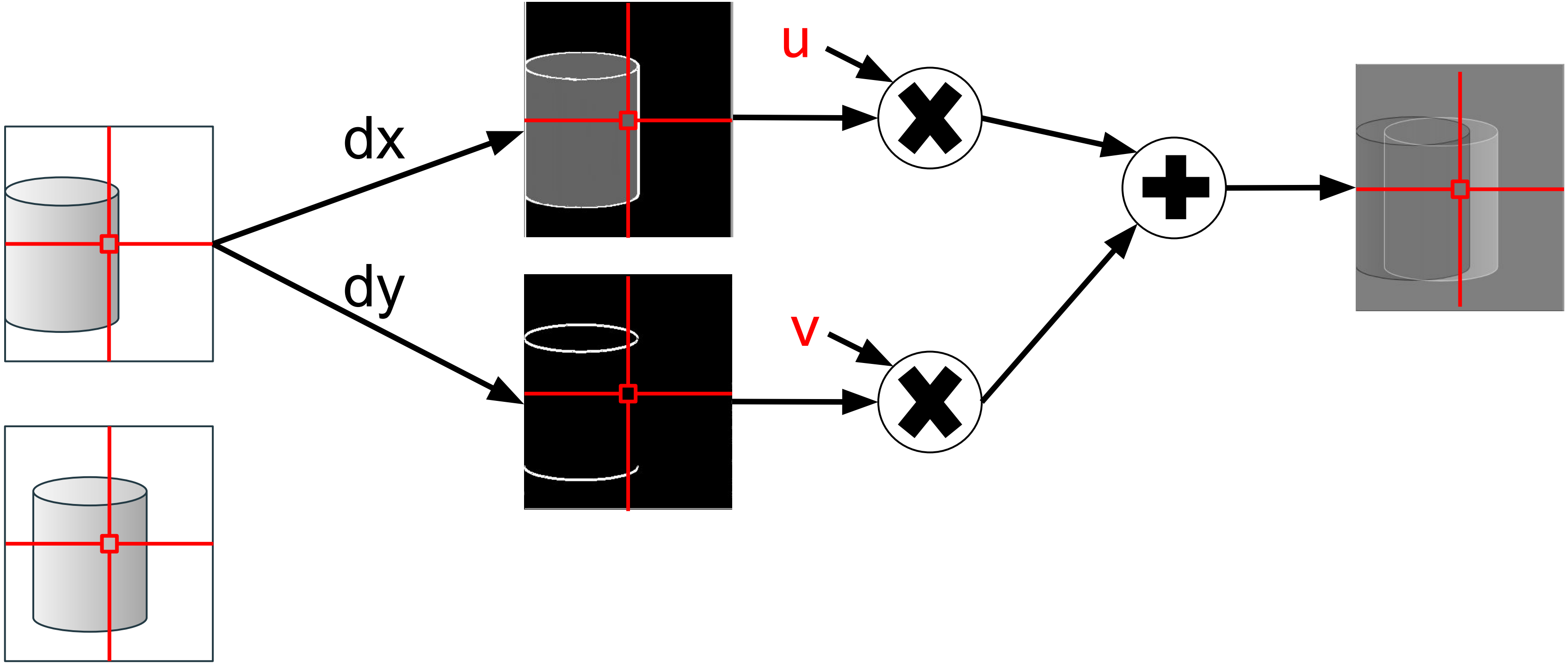
Lucas-Kanade Optical Flow

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$



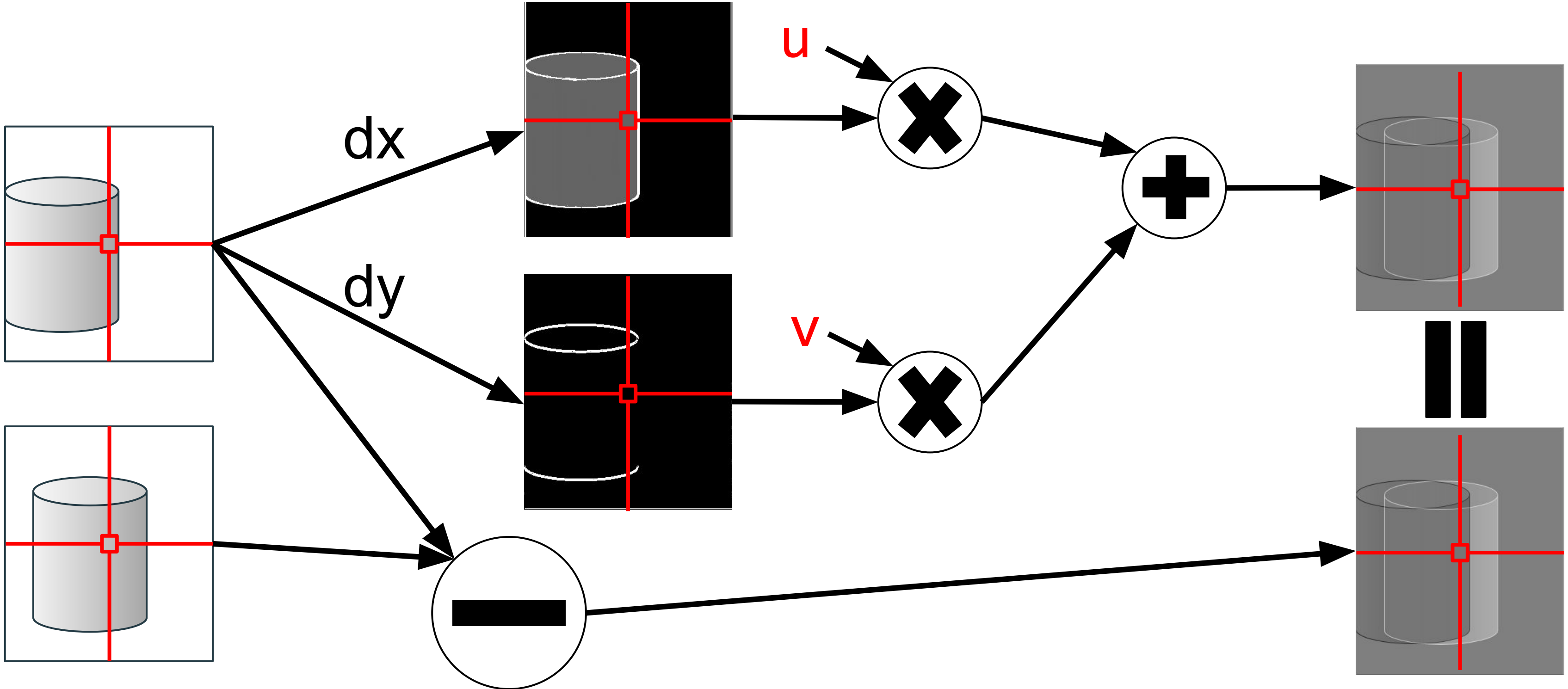
Lucas-Kanade Optical Flow

$$dx*\color{red}{u} + dy*\color{red}{v} = I_t[x,y] - I_{t+\Delta t}[x,y]$$



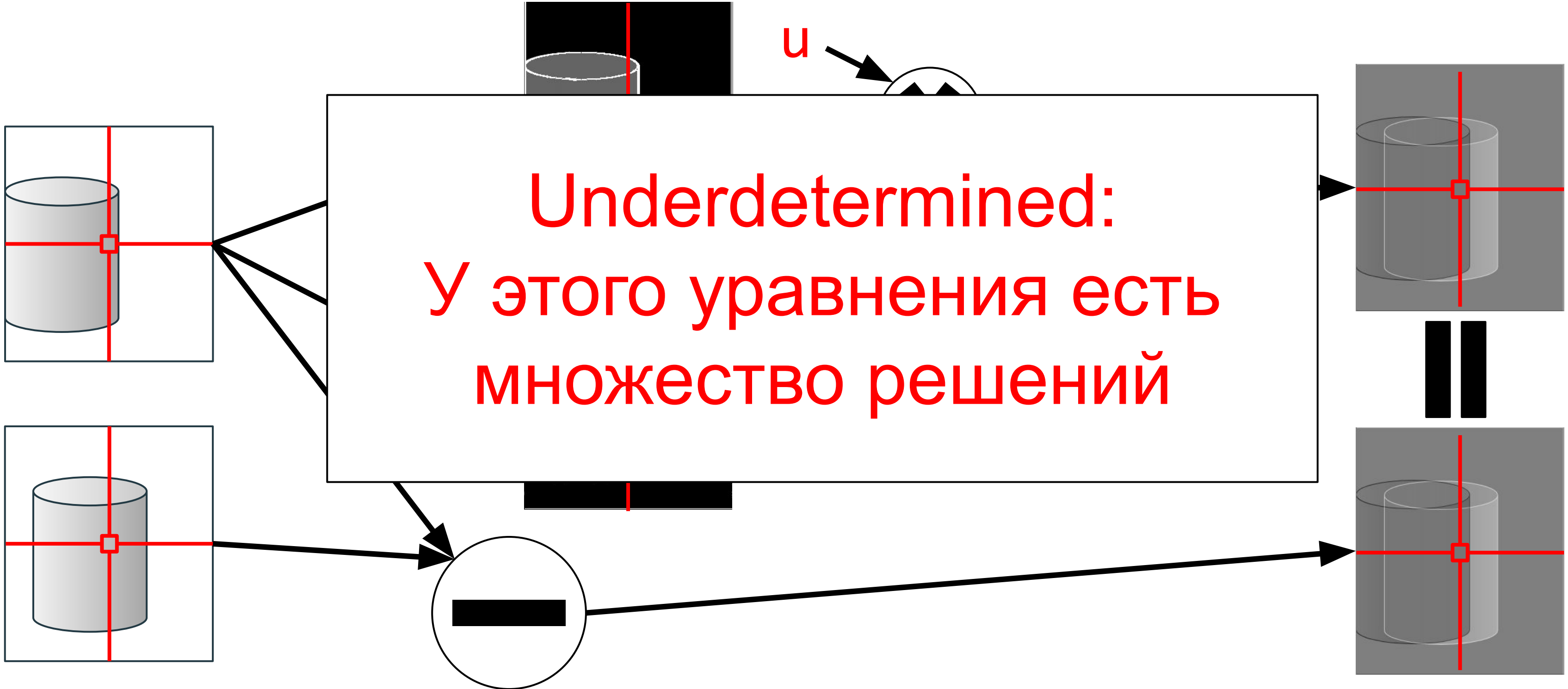
Lucas-Kanade Optical Flow

$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$



Lucas-Kanade Optical Flow

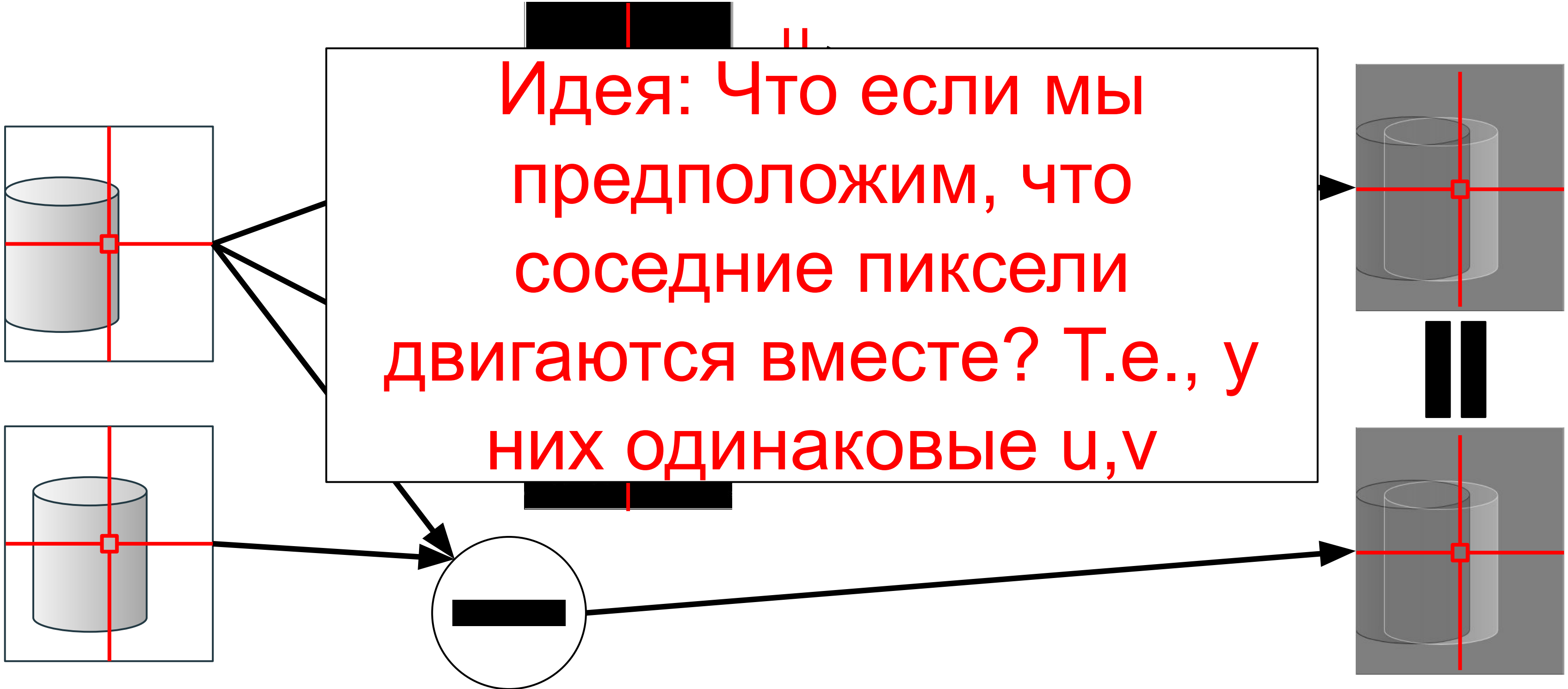
$$dx*\color{red}{u} + dy*\color{red}{v} = I_t[x,y] - I_{t+\Delta t}[x,y]$$



Underdetermined:
У этого уравнения есть множество решений

Lucas-Kanade Optical Flow

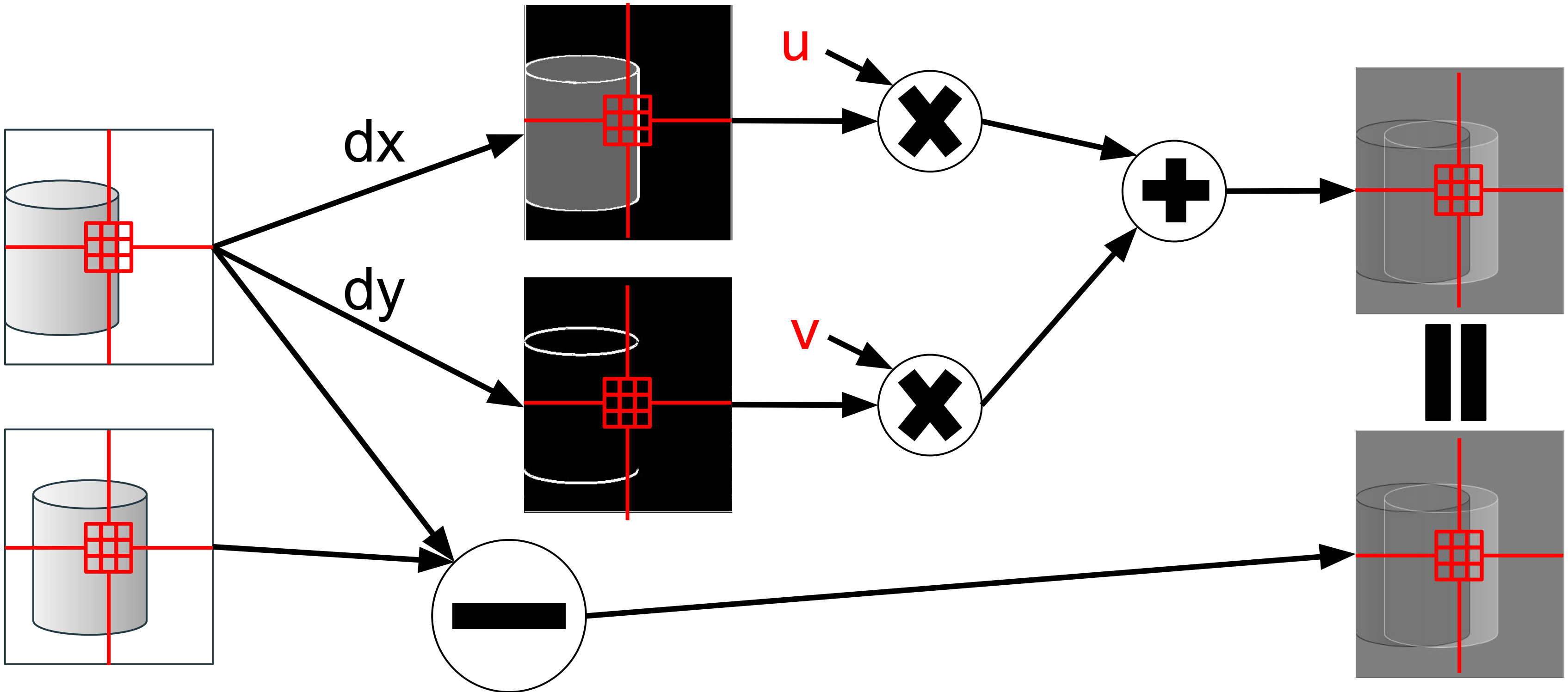
$$dx*u + dy*v = I_t[x,y] - I_{t+\Delta t}[x,y]$$



Идея: Что если мы предположим, что соседние пиксели двигаются вместе? Т.е., у них одинаковые u,v

Lucas-Kanade Optical Flow

$$dx*\color{red}{u} + dy*\color{red}{v} = I_t[x,y] - I_{t+\Delta t}[x,y]$$



Lucas-Kanade Optical Flow

$$dx_1 * u + dy_1 * v = I_t[x_1, y_1] - I_{t+\Delta t}[x_1, y_1]$$

$$dx_2 * u + dy_2 * v = I_t[x_2, y_2] - I_{t+\Delta t}[x_2, y_2]$$

...

$$dx_9 * u + dy_9 * v = I_t[x_9, y_9] - I_{t+\Delta t}[x_9, y_9]$$

9 уравнений, 2 неизвестных

Lucas-Kanade Optical Flow

$$dx_1 * u + dy_1 * v = I_t[x_1, y_1] - I_{t+\Delta t}[x_1, y_1]$$

$$dx_2 * u + dy_2 * v = I_t[x_2, y_2] - I_{t+\Delta t}[x_2, y_2]$$

$$\dots$$

$$dx_9 * u + dy_9 * v = I_t[x_9, y_9] - I_{t+\Delta t}[x_9, y_9]$$

9 уравнений, 2 неизвестных -> Overdetermined

Lucas-Kanade Optical Flow

$$S = \begin{bmatrix} dx_1 & dy_1 \\ dx_2 & dy_2 \\ \multicolumn{2}{c}{\cdots} \\ dx_9 & dy_9 \end{bmatrix} \qquad \mathbf{\Delta p} = \begin{bmatrix} u \\ v \end{bmatrix} \qquad T = \begin{bmatrix} I_t[x_1,y_1] - I_{t+\Delta t}[x_1,y_1] \\ I_t[x_2,y_2] - I_{t+\Delta t}[x_2,y_2] \\ \cdots \\ I_t[x_9,y_9] - I_{t+\Delta t}[x_9,y_9] \end{bmatrix}$$

$$S\mathbf{\Delta p} = T$$

Lucas-Kanade Optical Flow

$$S = \begin{bmatrix} dx_1 & dy_1 \\ dx_2 & dy_2 \\ \multicolumn{2}{c}{\cdots} \\ dx_9 & dy_9 \end{bmatrix} \qquad \mathbf{\Delta p} = \begin{bmatrix} u \\ v \end{bmatrix} \qquad T = \begin{bmatrix} I_t[x_1,y_1] - I_{t+\Delta t}[x_1,y_1] \\ I_t[x_2,y_2] - I_{t+\Delta t}[x_2,y_2] \\ \cdots \\ I_t[x_9,y_9] - I_{t+\Delta t}[x_9,y_9] \end{bmatrix}$$

$$S\mathbf{\Delta p} = T$$

Least-squares solution

$$||S\mathbf{\Delta p} - T||^2 = 0$$

$$\mathbf{\Delta p} = (S^TS)^{-1}S^TT$$

Lucas-Kanade Optical Flow

$$S = \begin{bmatrix} dx_1 & dy_1 \\ dx_2 & dy_2 \\ \multicolumn{2}{c}{\cdots} \\ dx_9 & dy_9 \end{bmatrix} \qquad \mathbf{\Delta p} = \begin{bmatrix} u \\ v \end{bmatrix} \qquad T = \begin{bmatrix} I_t[x_1,y_1] - I_{t+\Delta t}[x_1,y_1] \\ I_t[x_2,y_2] - I_{t+\Delta t}[x_2,y_2] \\ \cdots \\ I_t[x_9,y_9] - I_{t+\Delta t}[x_9,y_9] \end{bmatrix}$$

Что если это матрица необратима?

$$S\mathbf{\Delta p} = T$$

$$||S\mathbf{\Delta p} - T||^2 = 0$$

$$\mathbf{\Delta p} = (S^T S)^{-1} S^T T$$

Lucas-Kanade Optical Flow

$$S = \begin{bmatrix} dx_1 & dy_1 \\ dx_2 & dy_2 \\ \multicolumn{2}{c}{\dots} \\ dx_9 & dy_9 \end{bmatrix}$$

$$- I_{t+\Delta t}[x_1,y_1]$$
$$- I_{t+\Delta t}[x_2,y_2]$$
$$\dots$$
$$- I_{t+\Delta t}[x_9,y_9]$$

$$||S\mathbf{\Delta p} - T||^2 = 0$$

$$\mathbf{\Delta p} = (S^T S)^{-1} S^T T$$

Что если это матрица необратима?

E.G., нету структуры вокруг пикселя (x,y)

Lucas-Kanade Optical Flow

Проверяем на обратимость: $\Delta\mathbf{p} = (S^T S)^{-1} S^T T$

Lucas-Kanade Optical Flow

Проверяем на обратимость: $\mathbf{\Delta p} = (S^TS)^{-1}S^TT$

S^TS - симметричная, а значит можно разложить как:

$$S^TS = U \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} U^T$$

Lucas-Kanade Optical Flow

Проверяем на обратимость: $\mathbf{\Delta p} = (S^T S)^{-1} S^T T$

$S^T S$ - симметричная, а значит можно разложить как:

$$S^T S = U \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U^T$$

$$\det(S^T S - \lambda \mathbf{I}) = 0$$

Lucas-Kanade Optical Flow

Проверяем на обратимость: $\mathbf{\Delta p} = (S^TS)^{-1}S^TT$

S^TS - симметричная, а значит можно разложить как:

$$S^TS = U \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} U^T$$

$$\det(S^TS - \boldsymbol{\lambda}\mathbf{I}) = 0$$

$$\begin{vmatrix} \sum_i (dx_i)^2 - \lambda & \sum_i (dx_i)(dy_i) \\ \sum_i (dx_i)(dy_i) & \sum_i (dy_i)^2 - \lambda \end{vmatrix} = 0$$

Lucas-Kanade Optical Flow

Проверяем на обратимость: $\mathbf{\Delta p} = (S^T S)^{-1} S^T T$

$S^T S$ - симметричная, а значит можно разложить как:

$$S^T S = U \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} U^T$$

$$\det(S^T S - \lambda \mathbf{I}) = 0$$

$$\begin{vmatrix} \Sigma_i (dx_i)^2 - \lambda & \Sigma_i (dx_i)(dy_i) \\ \Sigma_i (dx_i)(dy_i) & \Sigma_i (dy_i)^2 - \lambda \end{vmatrix} = 0$$

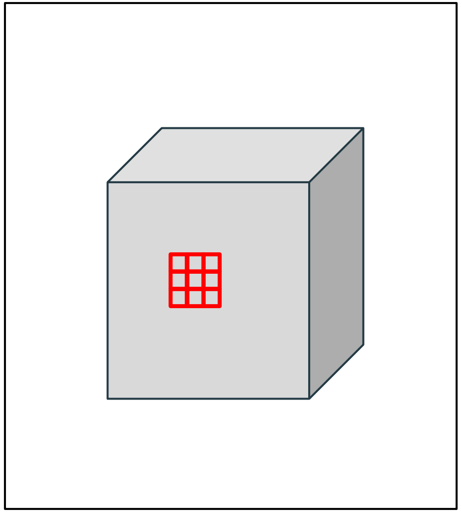
Если либо λ_1 или $\lambda_2 \leq 0$, значит S необратима и решения нету!

Lucas-Kanade Optical Flow

Как выбрать (x,y)? Что значит быть “good feature”?

Lucas-Kanade Optical Flow

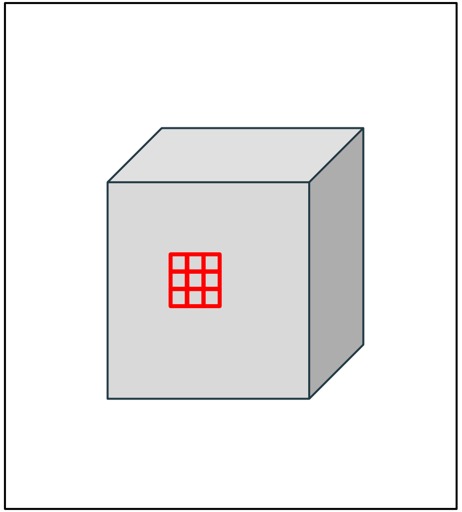
Как выбрать (x,y)? Что значит быть “good feature”?



Lucas-Kanade Optical Flow

Как выбрать (x,y)? Что значит быть “good feature”?

Bad

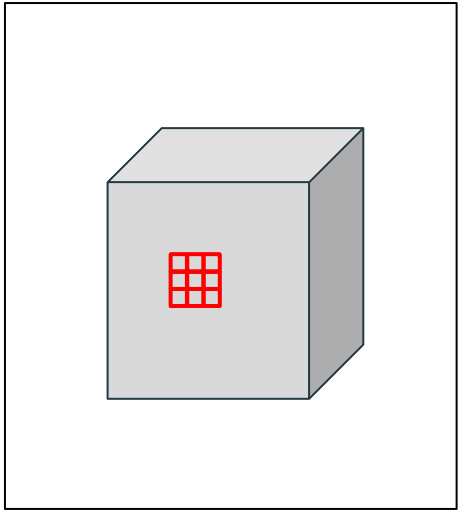


No gradient

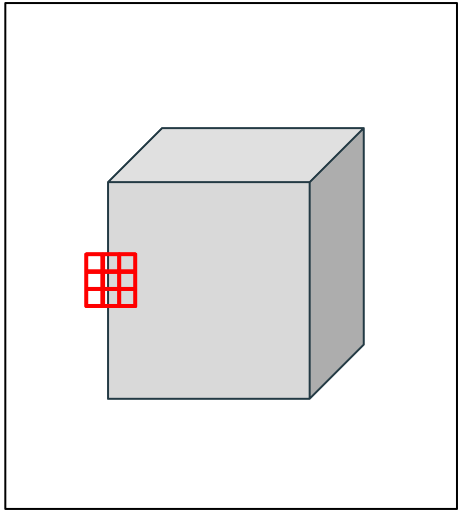
Lucas-Kanade Optical Flow

Как выбрать (x,y)? Что значит быть “good feature”?

Bad



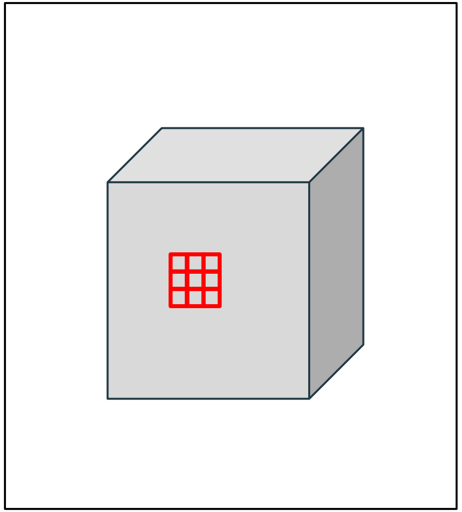
No gradient



Lucas-Kanade Optical Flow

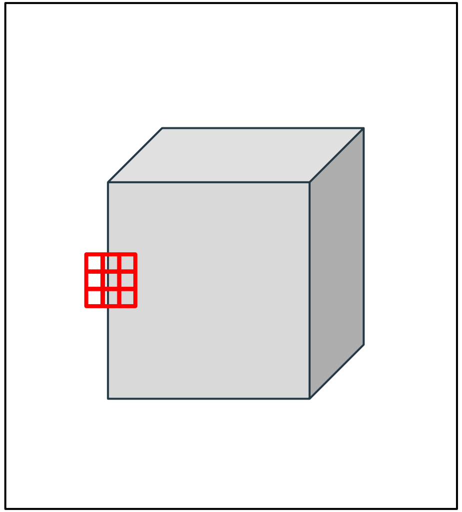
Как выбрать (x,y)? Что значит быть “good feature”?

Bad



No gradient

Slightly better

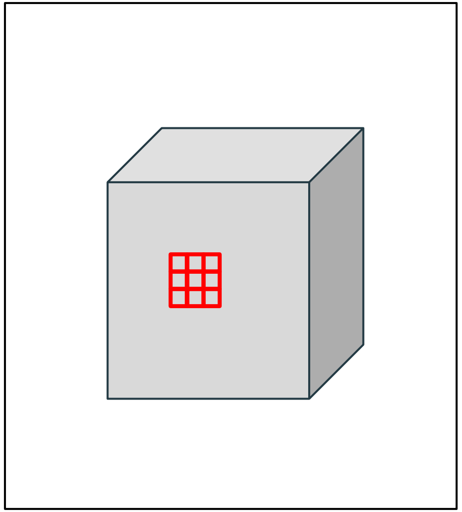


Can’t detect
vertical motion

Lucas-Kanade Optical Flow

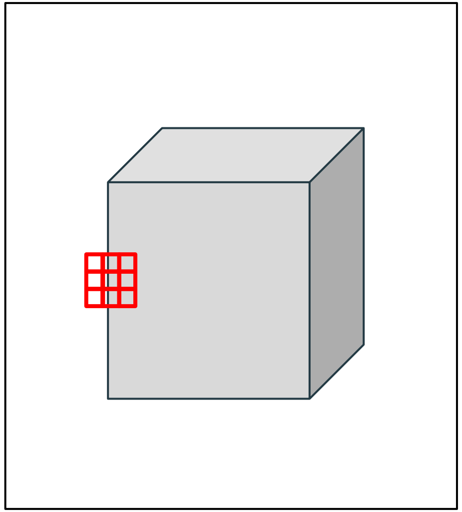
Как выбрать (x,y)? Что значит быть “good feature”?

Bad

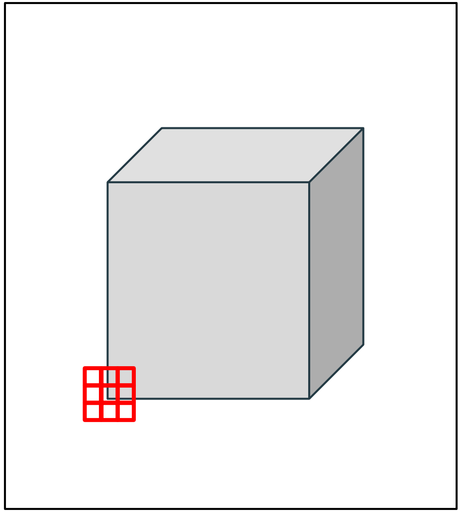


No gradient

Slightly better



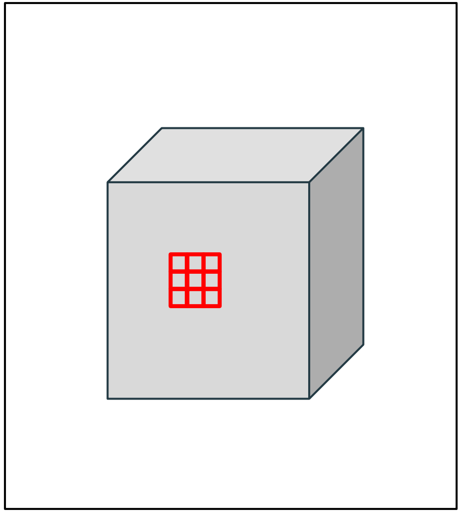
Can’t detect vertical motion



Lucas-Kanade Optical Flow

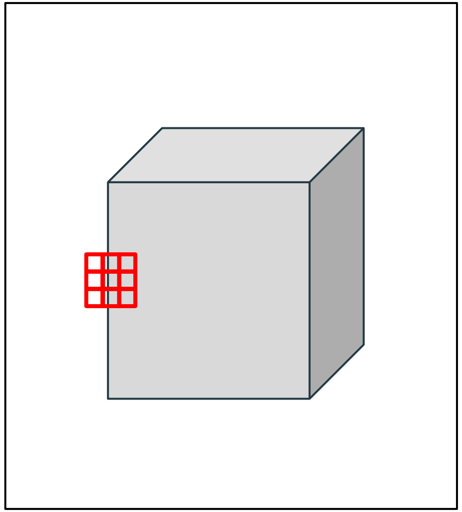
Как выбрать (x,y)? Что значит быть “good feature”?

Bad



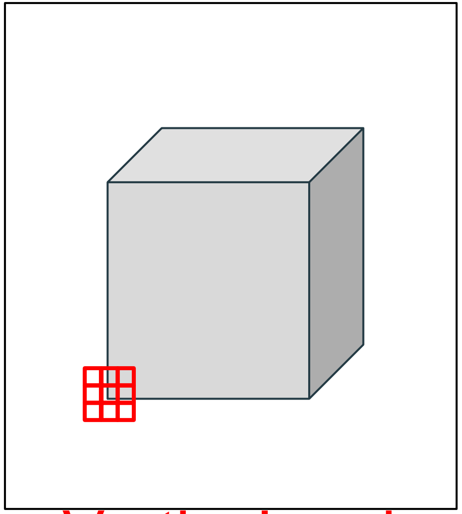
No gradient

Slightly better



Can’t detect vertical motion

Best

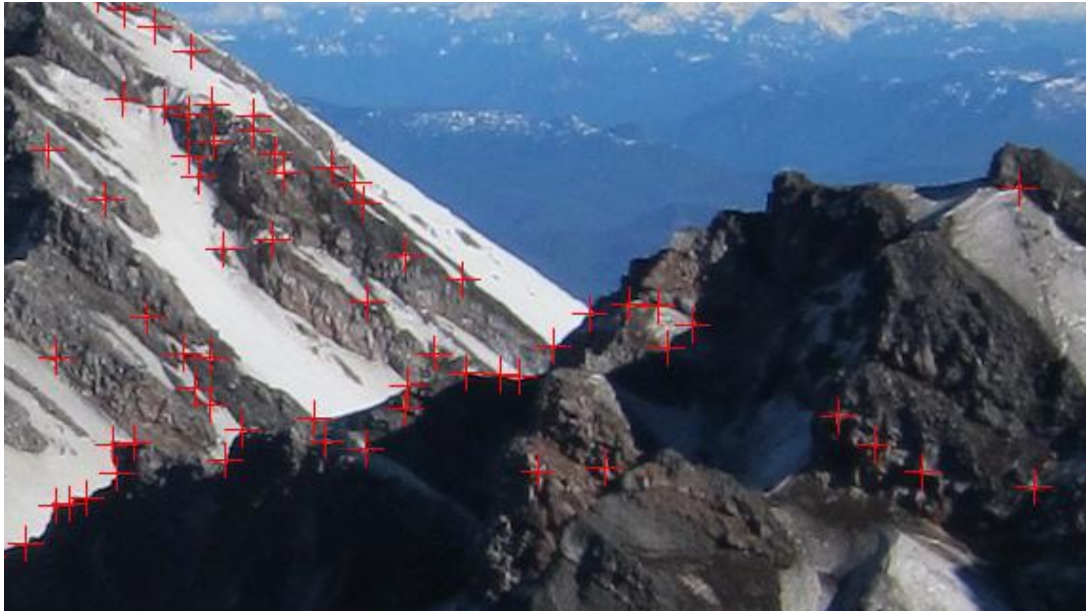


Vertical and horizontal motion

Lucas-Kanade Optical Flow

Как выбрать (x,y)? Что значит быть “good feature”?

УГЛЫ



Lucas-Kanade Optical Flow

Когда алгоритм фейлится?

Lucas-Kanade Optical Flow

Когда алгоритм фейлится?

-Изменения освещенности

-Большие движения

-Нет хороших фичей

-Проблема апертуры

Lucas-Kanade Optical Flow

Когда алгоритм фейлится?

-Изменения освещенности

-Большие движения

-Нет хороших фичей

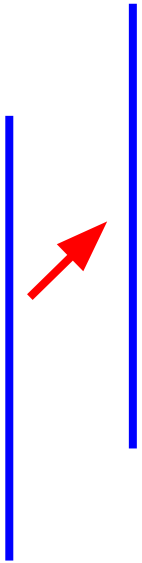
-Проблема апертуры

Lucas-Kanade Optical Flow

Проблема апертуры

Lucas-Kanade Optical Flow

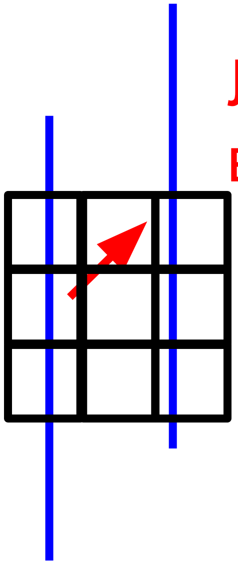
Проблема апертуры



Линия движется
вверх и направо

Lucas-Kanade Optical Flow

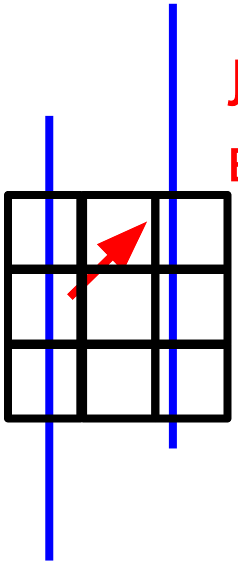
Проблема апертуры



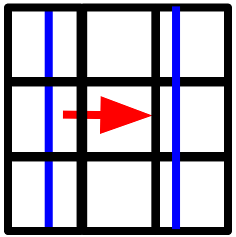
Линия движется
вверх и направо

Lucas-Kanade Optical Flow

Проблема апертуры



Линия движется
вверх и направо



Линия движется
только направо

Lucas-Kanade Optical Flow

Проблема апертуры - Barber pole illusion

Lucas-Kanade Optical Flow

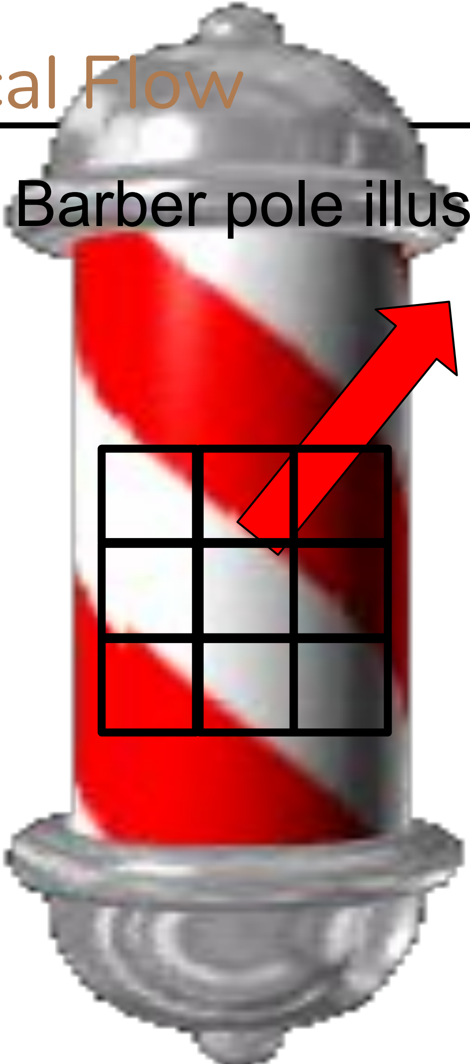
Проблема апертуры - Barber pole illusion

Lucas-Kanade Optical Flow

Проблема апертуры - Barber pole illusion

Lucas-Kanade Optical Flow

Проблема апертуры - Barber pole illusion



Смотрим через окошко - движется вверх и направо

Улучшения: Iterative LK

$$dx_1 * u + dy_1 * v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Улучшения: Iterative LK

$$dx_1 * u + dy_1 * v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

$$\Downarrow$$

Оцениваем один раз dx_1, dy_1
подставляем и снова переоцениваем

$$dx_2 * u + dy_2 * v = I_t[x,y] - I_{t+\Delta t}[x + dx_1, y + dy_1]$$

Улучшения: Iterative LK

$$dx_1 * u + dy_1 * v = I_t[x,y] - I_{t+\Delta t}[x,y]$$

Оцениваем dx_1, dy_1 подставляем и снова переоцениваем

$$dx_2 * u + dy_2 * v = I_t[x,y] - I_{t+\Delta t}[x + dx_1, y + dy_1]$$

Оцениваем один раз dx_2, dy_n подставляем и снова переоцениваем

...

$$dx_n * u + dy_n * v = I_t[x,y] - I_{t+\Delta t}[x + dx_{n-1}, y + dy_{n-1}]$$

Улучшения: Image Pyramids

Улучшения: Image Pyramids

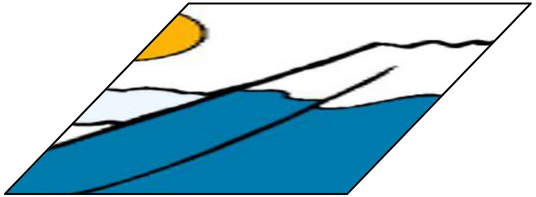


Image 1

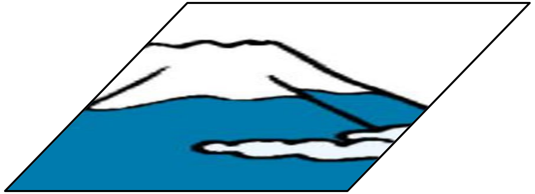


Image 2

Улучшения: Image Pyramids

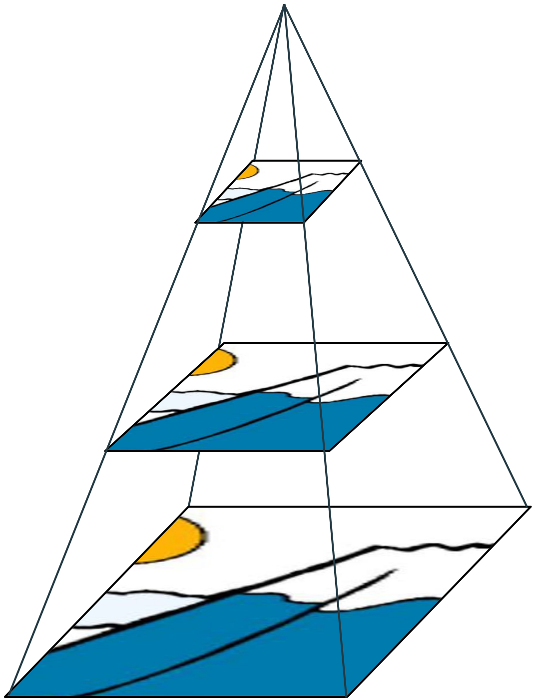


Image 1

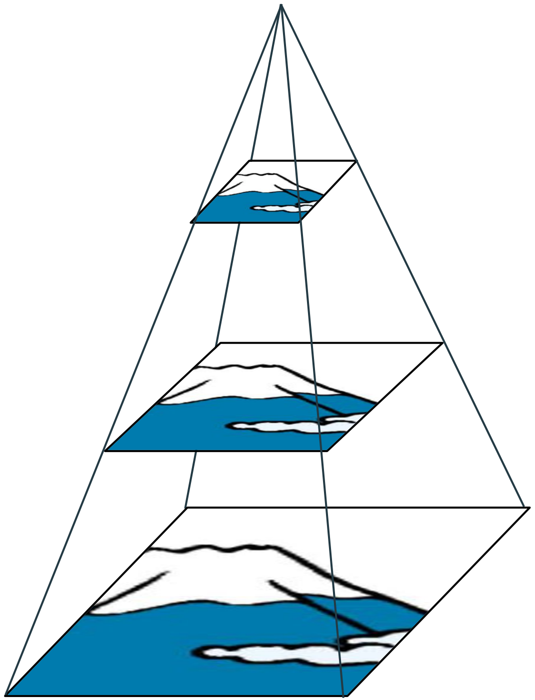
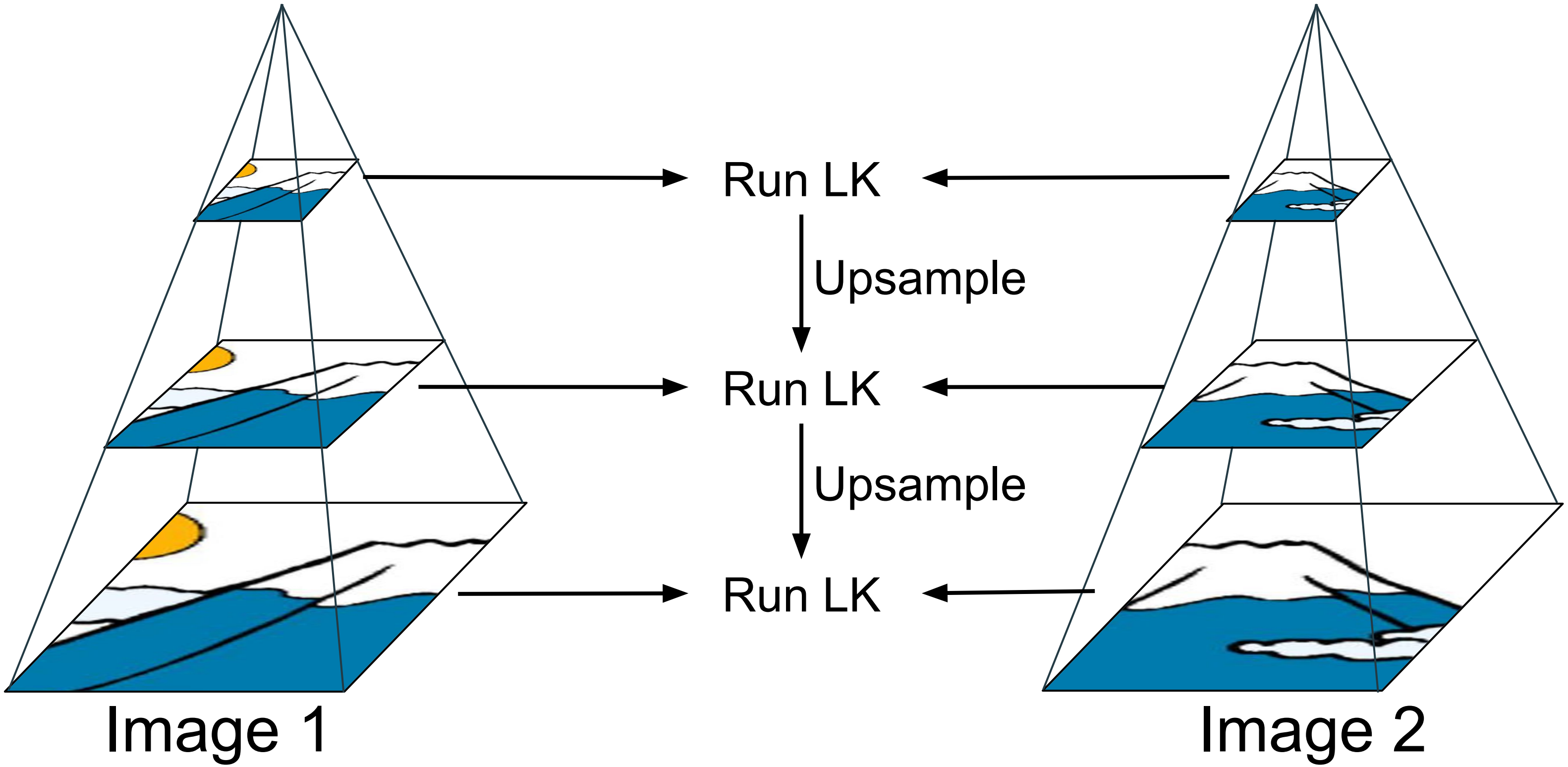


Image 2

Улучшения: Image Pyramids



Optical Flow

Sparse vs Dense

Sparse vs Dense

Считает OF для конкретных точек-фичей

Считает OF для всех пикселей

Sparse vs Dense

Считает OF для конкретных точек-фичей

Считает OF для всех пикселей

Lucas-Kanade - теоретически Dense, но на практике работает только на хороших точках-фичах, т.е., он sparse.

Как вычислять Dense Optical flow

Farnebäck Optical Flow

Farnebäck, Gunnar. "Two-frame motion estimation based on polynomial expansion." *Scandinavian conference on Image analysis*. Springer, Berlin, Heidelberg, 2003.

Farnebäck Optical Flow

Мы предположили, объект смещается на $\Delta\mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta\mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Farnebäck Optical Flow

Мы предположили, объект смещается на $\Delta \mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta \mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

в LK-методе мы аппроксимировали это рядом Тейлора первого порядка

Farnebäck Optical Flow

Мы предположили, объект смещается на $\Delta \mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta \mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

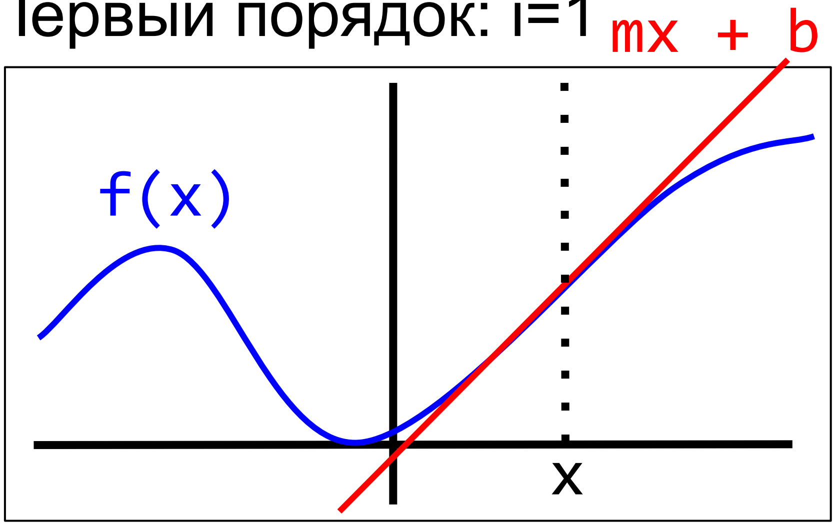
Вместо этого, давайте аппроксимировать рядом Тейлора второго порядка

Ряды Тейлора

Сложная функция ≈ полином n-ого порядка

$$f(x) \approx \Sigma_i \, a_i x^i$$

Первый порядок: i=1



$$\approx a_1 x^1 + a_0 x^0$$

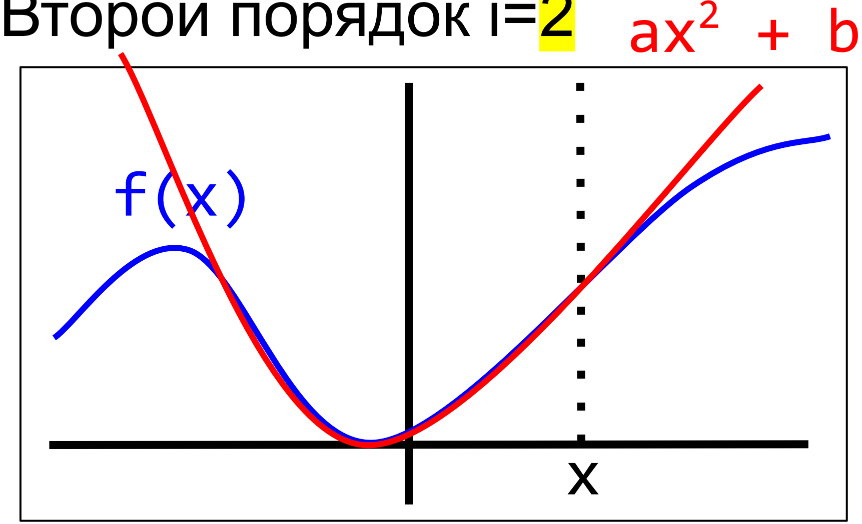
$$\approx mx + b$$

Ряды Тейлора

Сложная функция ≈ полином n-ого порядка

$$f(x) \approx \Sigma_i \, a_i x^i$$

Второй порядок i=2



$$\approx a_1 x^2 + a_1 x^1 + a_0 x^0$$

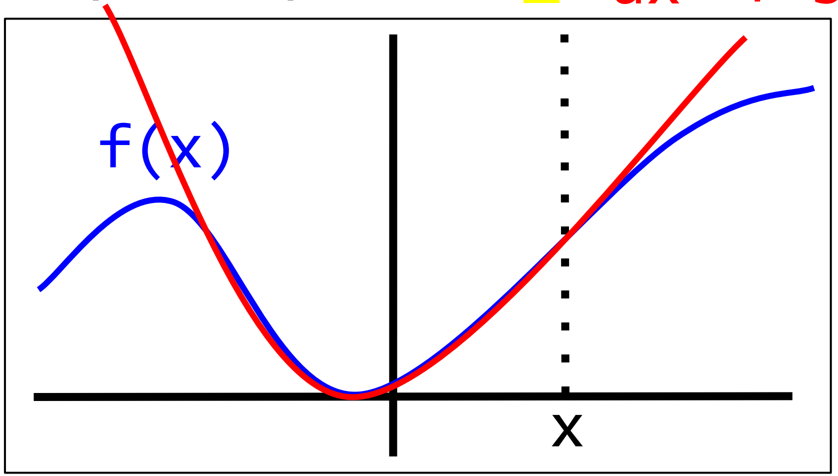
$$\approx ax^2 + bx + c$$

Ряды Тейлора

Сложная функция ≈ полином n-ого порядка

$$f(x) \approx \boldsymbol{\Sigma}_i \, a_i x^i$$

Второй порядок i=2 $ax^2 + bx + c$



$$\approx a_1 x^2 + a_1 x^1 + a_0 x^0$$

$$\approx ax^2 + bx + c$$

Vector notation

$$\approx \mathbf{x}^T A \mathbf{x} + b\mathbf{x} + c$$

Farnebäck Optical Flow

Мы предположили, объект смещается на $\Delta \mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta \mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Вместо этого, давайте аппроксимировать рядом Тейлора 2 порядка

Farnebäck Optical Flow

Мы предположили, объект смещается на $\boldsymbol{\Delta}\mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta\mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Вместо этого, давайте аппроксимировать рядом Тейлора 2 порядка

$$(\mathbf{p}-\Delta\mathbf{p})^T A_t(\mathbf{p}-\Delta\mathbf{p}) + b_t(\mathbf{p}-\Delta\mathbf{p}) + c_t = \mathbf{p}^T A_{t+\Delta t}\mathbf{p} + b_{t+\Delta t}\mathbf{p} + c_{t+\Delta t}$$

Farnebäck Optical Flow

Мы предположили, объект смещается на $\Delta\mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta\mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Вместо этого, давайте аппроксимировать рядом Тейлора 2 порядка

$$(\mathbf{p}-\Delta\mathbf{p})^T A_t (\mathbf{p}-\Delta\mathbf{p}) + b_t (\mathbf{p}-\Delta\mathbf{p}) + c_t = \mathbf{p}^T A_{t+\Delta t}\mathbf{p} + b_{t+\Delta t}\mathbf{p} + c_{t+\Delta t}$$

Параметры полинома из первого изображения

Параметры полинома из второго изображения

Farnebäck Optical Flow

Мы предположили, объект смещается на $\Delta \mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta \mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Вместо этого, давайте аппроксимировать рядом Тейлора 2 порядка

$$(\mathbf{p}-\Delta\mathbf{p})^T A_t (\mathbf{p}-\Delta\mathbf{p}) + b_t(\mathbf{p}-\Delta\mathbf{p}) + c_t = \mathbf{p}^T A_{t+\Delta t}\mathbf{p} + b_{t+\Delta t}\mathbf{p} + c_{t+\Delta t}$$

$$\mathbf{p}^T A_t \mathbf{p} + (b_t - 2A_t\Delta\mathbf{p})^T\mathbf{p} + \Delta\mathbf{p}^T A_t \Delta\mathbf{p} - b_t\Delta\mathbf{p} + c_t = \mathbf{p}^T A_{t+\Delta t}\mathbf{p} + \mathbf{p} b_{t+\Delta t} + c_{t+\Delta t}$$

Farnebäck Optical Flow

Мы предположили, объект смещается на $\Delta\mathbf{p}$ за время Δt

$$f(\mathbf{p} - \Delta\mathbf{p}, t) = f(\mathbf{p}, t + \Delta t)$$

Вместо этого, давайте аппроксимировать рядом Тейлора 2 порядка

$$(\mathbf{p}-\Delta\mathbf{p})^T A_t(\mathbf{p}-\Delta\mathbf{p}) + b_t(\mathbf{p}-\Delta\mathbf{p}) + c_t = \mathbf{p}^T A_{t+\Delta t}\mathbf{p} + b_{t+\Delta t}\mathbf{p} + c_{t+\Delta t}$$

$$\mathbf{p}^T A_t \mathbf{p} + (b_t - 2A_t\Delta\mathbf{p})^T\mathbf{p} + \Delta\mathbf{p}^T A_t \Delta\mathbf{p} - b_t\Delta\mathbf{p} + c_t = \mathbf{p}^T A_{t+\Delta t}\mathbf{p} + \mathbf{p} b_{t+\Delta t} + c_{t+\Delta t}$$

$$\boxed{A_t = A_{t+\Delta t}}$$

$$\boxed{b_{t+\Delta t} = b_t - 2A_t\Delta\mathbf{p}}$$

$$\boxed{c_{t+\Delta t} = \Delta\mathbf{p}^T A_t \Delta\mathbf{p} - b_t\Delta\mathbf{p} + c_t}$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Решение для $\mathbf{\Delta p}$

$$\mathbf{\Delta p = } -\tfrac{1}{2}(A_t)^{-1}(b_{t+\Delta t} - b_t)$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Решение для $\mathbf{\Delta p}$

$$\mathbf{\Delta p} = -½(A_t)^{-1}(b_{t+\Delta t} - b_t)$$

A, b, и c могут быть найдены поточечно используя соседние пиксели как и в предыдущем алгоритма

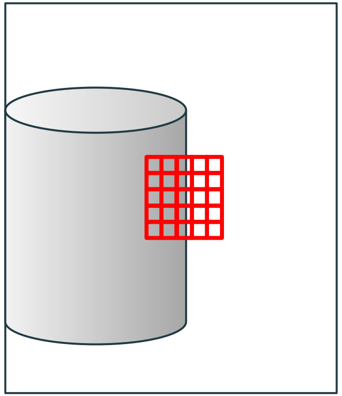
Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Решение для $\mathbf{\Delta p}$

$$\mathbf{\Delta p} = -\tfrac{1}{2}(A_t)^{-1}(b_{t+\Delta t} - b_t)$$

A, b, и c могут быть найдены поточечно используя соседние пиксели как и в предыдущем алгоритма



Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Решение для $\mathbf{\Delta p}$

$$\mathbf{\Delta p = } -½(A_t)^{-1}(b_{t+\Delta t} - b_t)$$

A, b, и c могут быть найдены поточечно используя соседние пиксели как и в предыдущем алгоритма

$$S = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \multicolumn{2}{c}{\cdots} \\ x_k & y_k \end{bmatrix} \quad T = \begin{bmatrix} I[x_1,y_1] \\ I[x_2,y_2] \\ \cdots \\ I[x_k,y_k] \end{bmatrix}$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Решение для $\mathbf{\Delta p}$

$$\mathbf{\Delta p = } -\tfrac{1}{2}(A_t)^{-1}(b_{t+\Delta t} - b_t)$$

A, b, и с могут быть найдены поточечно используя соседние пиксели как и в предыдущем алгоритма

$$S = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ & \cdots \\ x_k & y_k \end{bmatrix} \quad T = \begin{bmatrix} I[x_1,y_1] \\ I[x_2,y_2] \\ \cdots \\ I[x_k,y_k] \end{bmatrix}$$

$$SAS^T + bS^T + c = T$$

$$||SAS^T + bS^T + c - T||^2 = 0$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Решение для $\mathbf{\Delta p}$

$$\mathbf{\Delta p} = -\tfrac{1}{2}(A_t)^{-1}(b_{t+\Delta t} - b_t)$$

A, b, и c могут быть найдены поточечно используя соседние пиксели как и в предыдущем алгоритма

$$S = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \multicolumn{2}{c}{\cdots} \\ x_k & y_k \end{bmatrix} \quad T = \begin{bmatrix} I[x_1,y_1] \\ I[x_2,y_2] \\ \cdots \\ I[x_k,y_k] \end{bmatrix}$$

$$SAS^T + bS^T + c = T$$

$$||SAS^T + bS^T + c - T||^2 = 0$$

Quadratic Regression

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Решение для $\mathbf{\Delta p}$

$$\mathbf{\Delta p} = -\tfrac{1}{2}(A_t)^{-1}(b_{t+\Delta t} - b_t)$$

Такая же проблема как раньше: что если матрица необратима?

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\boldsymbol{\Delta p}$$

Попробуем другой подход:

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Попробуем другой подход:

$$A_t\mathbf{\Delta p} = -½(b_{t+\Delta t} - b_t)$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\boldsymbol{\Delta p}$$

Попробуем другой подход:

$$\boxed{A_t = A_{t+\Delta t}}$$

$$A_t\boldsymbol{\Delta p} = -\tfrac{1}{2}(b_{t+\Delta t} - b_t)$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Попробуем другой подход:

$$A_t\mathbf{\Delta p} = -\tfrac{1}{2}(b_{t+\Delta t} - b_t)$$

$$\boxed{A_t = A_{t+\Delta t}}$$

В реальности не равны

$$\boxed{A_t \neq A_{t+\Delta t}}$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\boldsymbol{\Delta p}$$

Попробуем другой подход:

$$A_t\boldsymbol{\Delta p} = -½(b_{t+\Delta t} - b_t)$$

$\boxed{A_t = A_{t+\Delta t}}$

$\downarrow$ В реальности не равны

$\boxed{A_t \neq A_{t+\Delta t}}$

$\downarrow$ Усредняем

$\boxed{A = ½(A_t + A_{t+\Delta t})}$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Попробуем другой подход:

$$A_t\mathbf{\Delta p} = -\tfrac{1}{2}(b_{t+\Delta t} - b_t)$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Попробуем другой подход:

$$A_t\mathbf{\Delta p} = -½(b_{t+\Delta t} - b_t)$$

$$\boxed{\Delta b = -½(b_{t+\Delta t} - b_t)}$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\boldsymbol{\Delta p}$$

Попробуем другой подход:

$$A\boldsymbol{\Delta p} = \Delta b$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Попробуем другой подход:

$$A\mathbf{\Delta p} = \Delta b$$

Предполагаем, что соседи двигаются вместе

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Попробуем другой подход:

$$A\mathbf{\Delta p} = \Delta b$$

Предполагаем, что соседи двигаются вместе

$$A_i\mathbf{\Delta p} = \Delta b_i$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Попробуем другой подход:

$$A\mathbf{\Delta p} = \Delta b$$

Предполагаем, что соседи двигаются вместе

$$A_i\mathbf{\Delta p} = \Delta b_i$$

МНК

$$\mathbf{\Sigma}_i\, w_i \,||A_i\mathbf{\Delta p} - \Delta b_i||^2$$

Farnebäck Optical Flow

$$b_{t+\Delta t} = b_t - 2A_t\mathbf{\Delta p}$$

Попробуем другой подход:

$$A\mathbf{\Delta p} = \Delta b$$

Предполагаем, что соседи двигаются вместе

$$A_i\mathbf{\Delta p} = \Delta b_i$$

МНК

$$\Sigma_i\, w_i||A_i\mathbf{\Delta p} - \Delta b_i||^2$$

Гауссовы веса (мы больше учитываем центральные пиксели чем краевые)

Farnebäck Optical Flow

Все еще предполагаем, что $\Delta \mathbf{p}$ одинаковая для всех соседних пикселей, хотя на самом деле это не так...

Farnebäck Optical Flow

Все еще предполагаем, что $\mathbf{\Delta p}$ одинаковая для всех соседних пикселей, хотя на самом деле это не так...

ПРЕДПОЛОЖИМ: что $\mathbf{\Delta p}$ - аффинное преобразование всех соседних пикселей

Farnebäck Optical Flow

Все еще предполагаем, что $\mathbf{\Delta p}$ одинаковая для всех соседних пикселей, хотя на самом деле это не так...

ПРЕДПОЛОЖИМ: что $\mathbf{\Delta p}$ - аффинное преобразование всех соседних пикселей

$$\mathbf{\Delta p} = \begin{bmatrix} a_1 + a_2x + a_3y + a_7x^2 + a_8xy \\ a_4 + a_5x + a_6y + a_7xy + a_8y^2 \end{bmatrix}$$

Farnebäck Optical Flow

Все еще предполагаем, что $\mathbf{\Delta p}$ одинаковая для всех соседних пикселей, хотя на самом деле это не так...

ПРЕДПОЛОЖИМ: что $\mathbf{\Delta p}$ - аффинное преобразование всех соседних пикселей

$$\mathbf{\Delta p} = \begin{bmatrix} a_1 + a_2 x + a_3 y + a_7 x^2 + a_8 xy \\ a_4 + a_5 x + a_6 y + a_7 xy + a_8 y^2 \end{bmatrix}$$

$$\mathbf{\Delta p} = S\mathbf{d}$$

$$S = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 \end{bmatrix}$$

$$\mathbf{d} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix}$$

Farnebäck Optical Flow

$$\boldsymbol{\Sigma}_i \; w_i || A_i \boldsymbol{\Delta p} - \boldsymbol{\Delta} b_i ||^2$$

Farnebäck Optical Flow

$$\boldsymbol{\Sigma}_i\ w_i||A_i\boldsymbol{\Delta}\mathbf{p} - \boldsymbol{\Delta}b_i||^2$$

Подставлем S**d** вместо**Δp**.

$$\boldsymbol{\Sigma}_i\ w_i||A_iS\mathbf{d} - \boldsymbol{\Delta}b_i||^2$$

Farnebäck Optical Flow

$$\boldsymbol{\Sigma}_i\, w_i \,||A_i \boldsymbol{\Delta}\mathbf{p} - \boldsymbol{\Delta} b_i||^2$$

Подставлем S**d** вместо**Δp.**

$$\boldsymbol{\Sigma}_i\, w_i \,||A_i S\mathbf{d} - \boldsymbol{\Delta} b_i||^2$$

Решаем относительно **d**

$$\mathbf{d} = \left(\boldsymbol{\Sigma}_i w_i (S_i)^T (A_i)^T S_i\right)^{-1} \boldsymbol{\Sigma}_i w_i (S_i)^T (A_i)^T \boldsymbol{\Delta} b_i$$

Farnebäck Optical Flow

$$\boldsymbol{\Sigma}_i\, w_i \,||A_i \boldsymbol{\Delta} \mathbf{p} - \boldsymbol{\Delta} b_i||^2$$

Подставлем S**d** вместо**Δp**.

$$\boldsymbol{\Sigma}_i\, w_i \,||A_i S\mathbf{d} - \boldsymbol{\Delta} b_i||^2$$

Решаем относительно **d**

$$\mathbf{d} = \left(\boldsymbol{\Sigma}_i w_i (S_i)^T (A_i)^T S_i\right)^{-1} \boldsymbol{\Sigma}_i w_i (S_i)^T (A_i)^T \boldsymbol{\Delta} b_i$$

Такая же проблема обратимости, но на практике проблем меньше

Image captioning

Common Objects in COntext (COCO)

80 objects

117,261 train/val images

902,435 object instances

Segmentation masks for each instance



http://cocodataset.org/#home

COCO dataset also has captions!

5 captions per image

Детекция/сегментация - это просто поиск паттернов

Чтобы описать изображение, надо его понять

Необходимо моделировать и визуальную информацию, и язык

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.

A horse carrying a large load of hay and two people sitting on it.

Bunk bed with a narrow shelf sitting underneath it.

NLP using neural networks

Изображения статичны, у языка есть в некоторым смысле компонента времени

Символы/слова появляются последовательно, нужно прочитать предыдущие слова, что понять текущее слово (похоже на последовательность кадров в видео)

Как обрабатывать time-series data нейронными сетями?

A long time ago in a ______ far, far away....

A long time ago in a galaxy far, far away….

Neural networks + language

Наивный подход

Input: string

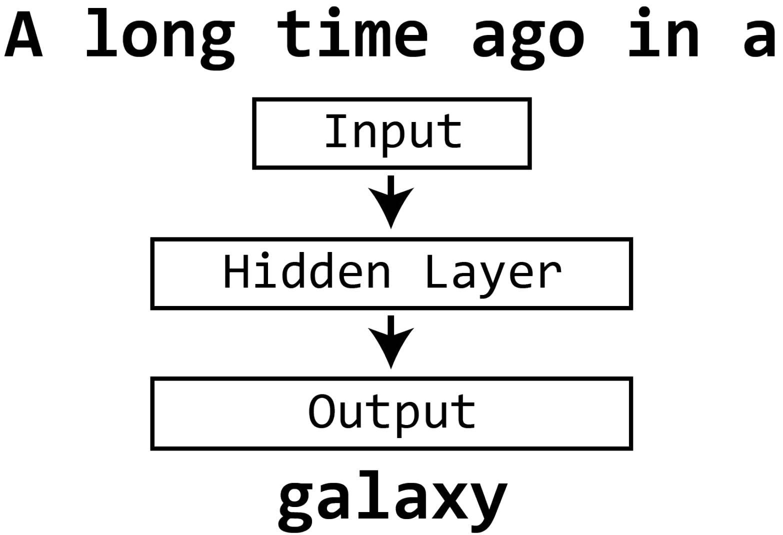
Output: string

Какое численное представление?

~3000 слов

1-hot encoding всех строк из 6 слов

это вектор длина 18,000



Что насчет предложений из 7 или 8 или 24 слов?

Различные архитектуры каждый раз? А что если мы хотим другой аутпут?

Recurrent neural network

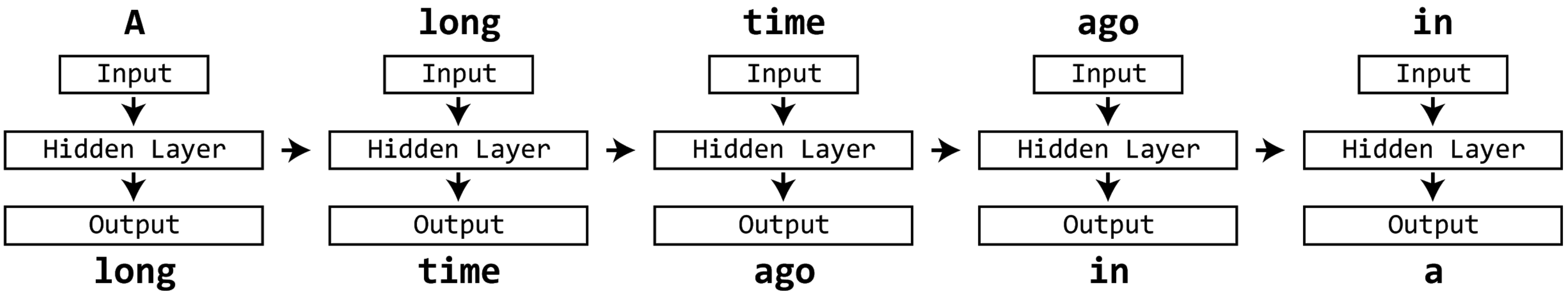
Позволяют обрабатывать последовательные данные

Идея:

Читаем один токен (слово/буква) за раз

Получаем аутпут

Обновляем внутреннюю память

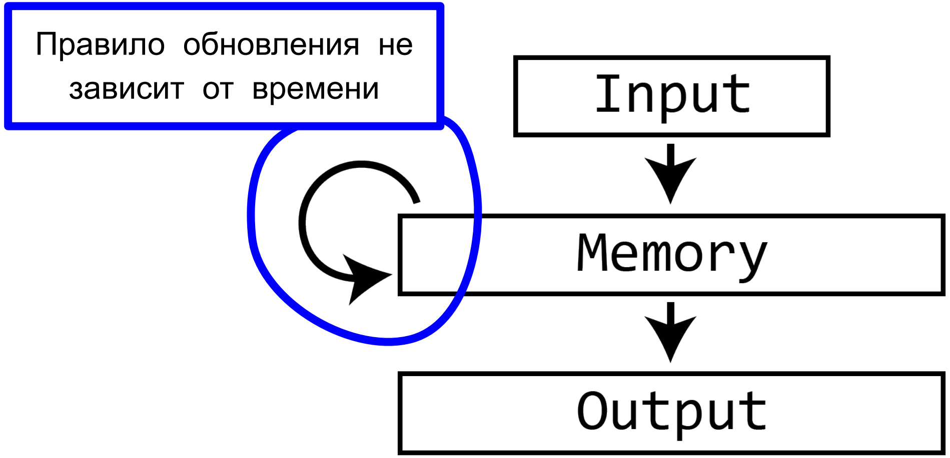


Recurrent neural network

Позволяют обрабатывать последовательные данные

Идея:

- Читаем один токен (слово/буква) за раз
- Получаем аутпут
- Обновляем внутреннюю память



Recurrent neural network

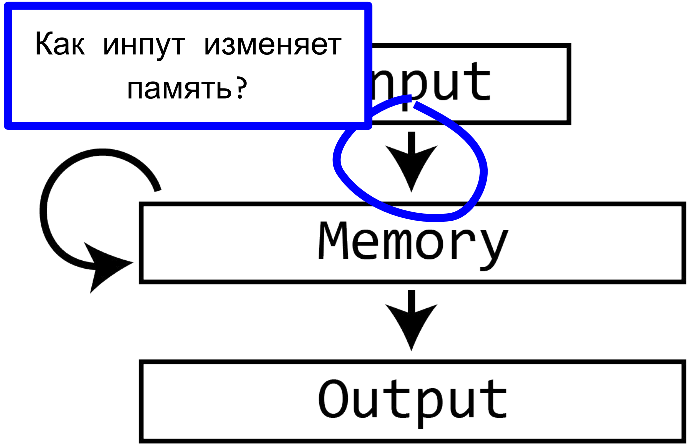
Позволяют обрабатывать последовательные данные

Идея:

Читаем один токен (слово/буква) за раз

Получаем аутпут

Обновляем внутреннюю память



Recurrent neural network

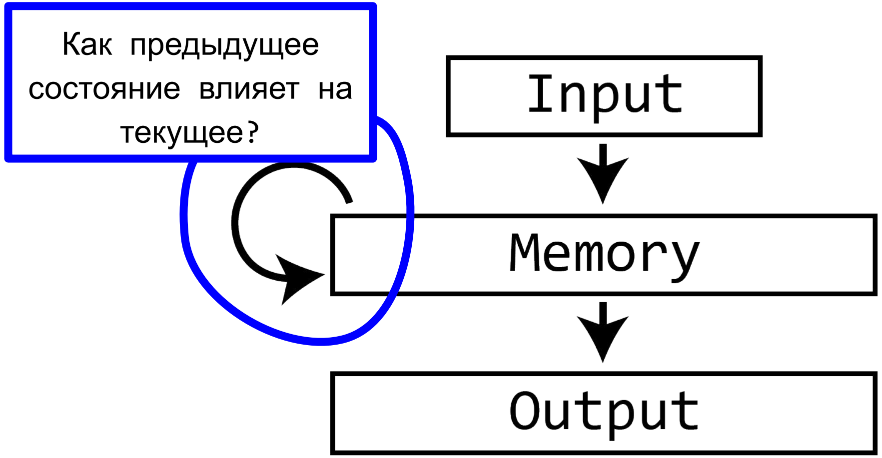
Позволяют обрабатывать последовательные данные

Идея:

Читаем один токен (слово/буква) за раз

Получаем аутпут

Обновляем внутреннюю память

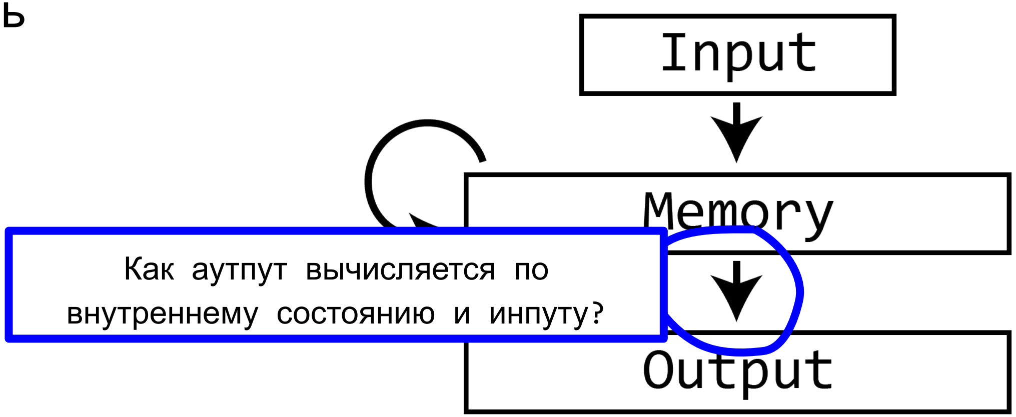


Recurrent neural network

Позволяют обрабатывать последовательные данные

Идея:

- Читаем один токен (слово/буква) за раз
- Получаем аутпут
- Обновляем внутреннюю память

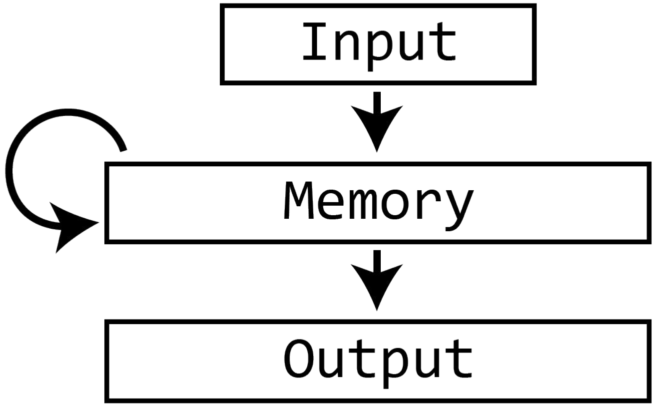


Vanilla RNN

Имея input $\mathbf{x}_t$, предыдущую память $\mathbf{h}_{t-1}$, получаем output $\mathbf{y}_t$

$$\mathbf{h}_t = \varphi(\mathbf{w}\mathbf{x}_t + \mathbf{v}\mathbf{h}_{t-1})$$

$$\mathbf{y}_t = \mathbf{h}_t$$



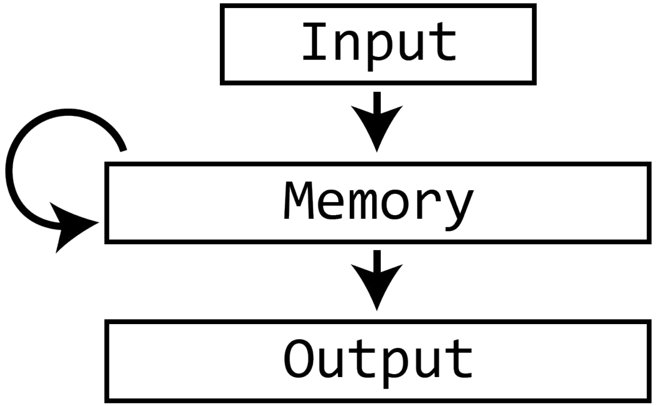
Vanilla RNN

Имея input $\mathbf{x}_t$, предыдущую память $\mathbf{h}_{t-1}$, получаем output $\mathbf{y}_t$

$\mathbf{h}_t = \varphi(\mathbf{w}\mathbf{x}_t + \mathbf{v}\mathbf{h}_{t-1})$

$\mathbf{y}_t = \mathbf{h}_t$

Аутпут - то же самое, что и текущая память



Vanilla RNN

Имея input $\mathbf{x}_t$, предыдущую память $\mathbf{h}_{t-1}$, получаем output $\mathbf{y}_t$

$\mathbf{h}_t = \varphi(\mathbf{w}\mathbf{x}_t + \mathbf{v}\mathbf{h}_{t-1})$

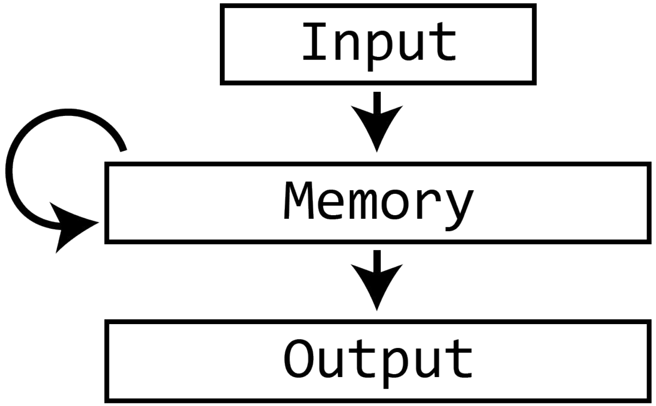
$\mathbf{y}_t = \mathbf{h}_t$

Аутпут - то же самое, что и текущая память

w -матрица весов воздействующих на инпут

V - матрица весов воздействующих на память

φ какая-то функция активации



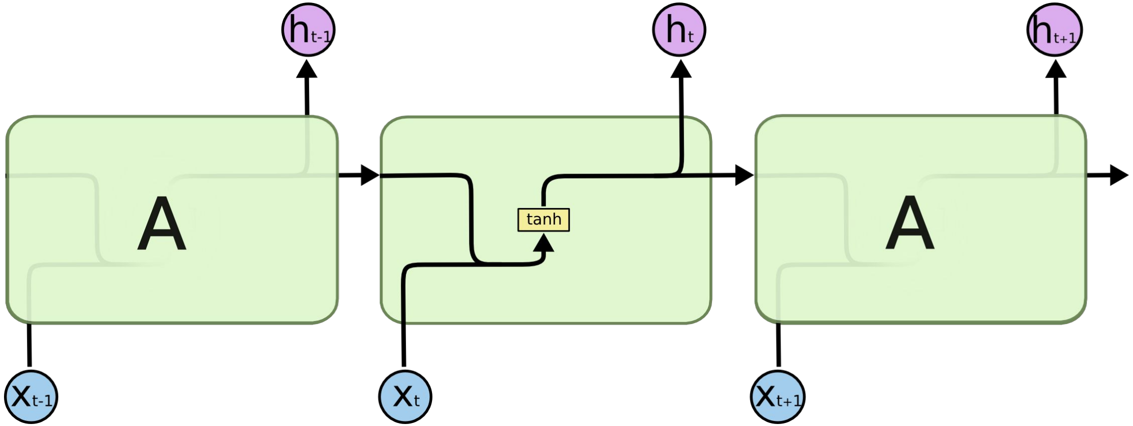
Vanilla RNN

Имея инпут $\mathbf{x_t}$, предыдущую память $\mathbf{h_{t-1}}$, получить аутпут $\mathbf{y_t}$

На практике, объединяем x_t и h_{t-1} и используем один набор весов

$\mathbf{h_t} = \varphi(\mathbf{w}\cdot[\mathbf{x_t},\mathbf{h_{t-1}}])$

$\mathbf{y_t} = \mathbf{h_t}$



http://colah.github.io/posts/2015-08-Understanding-LSTMs

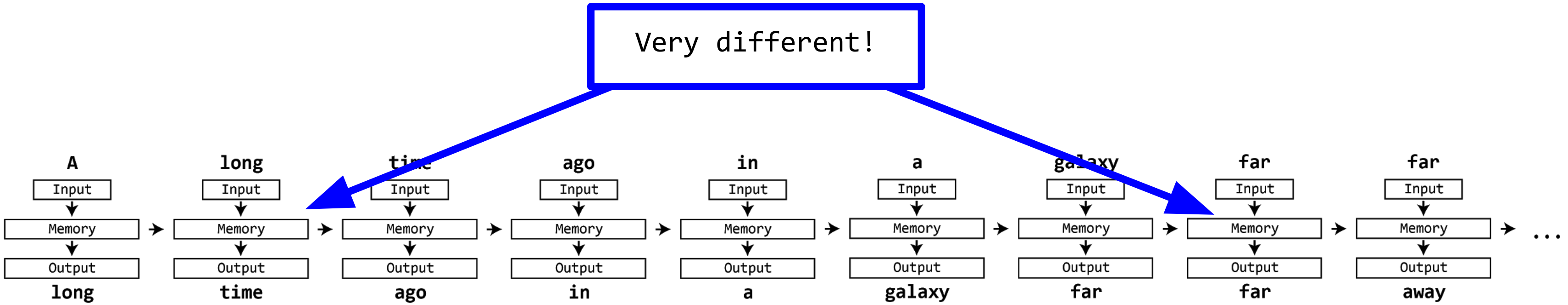
Vanilla RNN

Имея инпут $\mathbf{x}_t$, предыдущую память $\mathbf{h}_{t-1}$, получить аутпут $\mathbf{y}_t$

$$\mathbf{h}_t = \varphi(\mathbf{w}\cdot[\mathbf{x}_t,\mathbf{h}_{t-1}])$$

$$\mathbf{y}_t = \mathbf{h}_t$$

Проблемы: неулавливает дальние взаимосвязи, память полность пересчитывается на каждой итерации, трудно что-то запомнить на долгое время



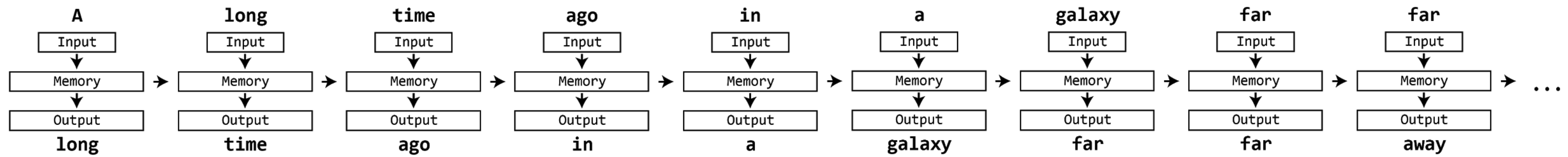
Идея: выборочно перезаписывать память

Вместо того, чтобы каждый раз полностью переписывать память, будем читать и записывать в нее информацию как это делает “компьютер”

Вычисляем gating function чтобы решить какую часть памяти оставляем и какую изменяем

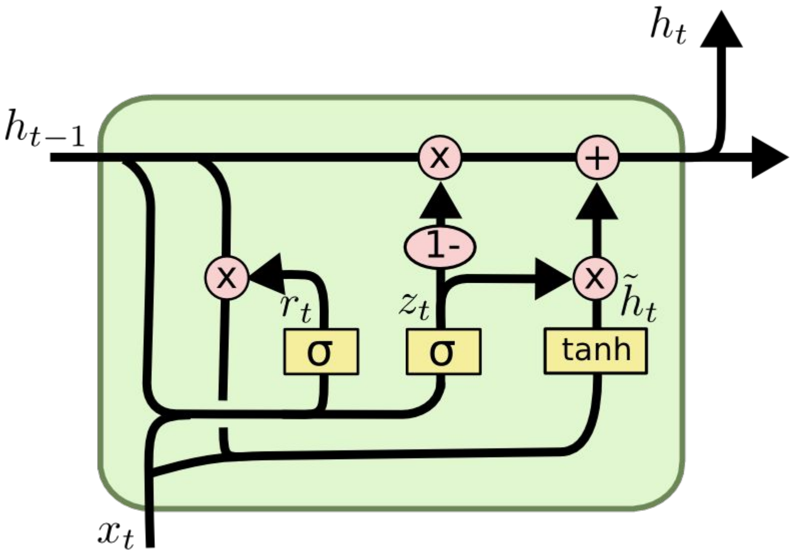
Это позволяет нам производить изменения, а также помнить важные вещи длительное время.

Самые популярные архитектуры: GRU, LSTM, ...



GRU: Gated recurrent units

Разберемся по порядку



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

GRU: Gated recurrent units

Интуитивное объяснение, как работает GRU

Смотрим на память и на вход, решаем какая часть памяти важная

Возможно, мы не были уверены, что будет в предложении дальше, но теперь с новым инпутом, мы знаем, поэтому можем забыть прочие варианты

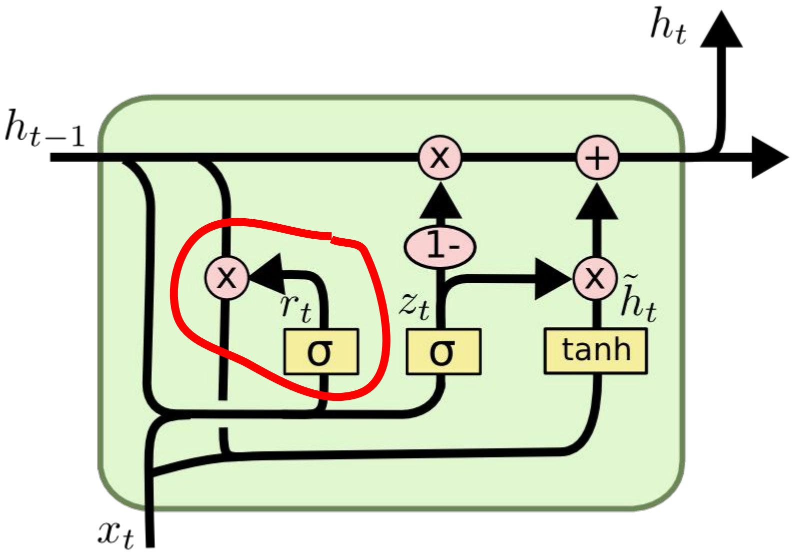
Посчитать обновление памяти, которое будем добавлять

Решить какие части памяти сохранить, а какие обновить

Аутпут это память, которую мы сохранили + апдейт, который мы посчитали

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Reset gate: ignore some memory



$$z_t = \sigma$$

Определяем, на какие участки памяти обратить внимание, какие игнорировать

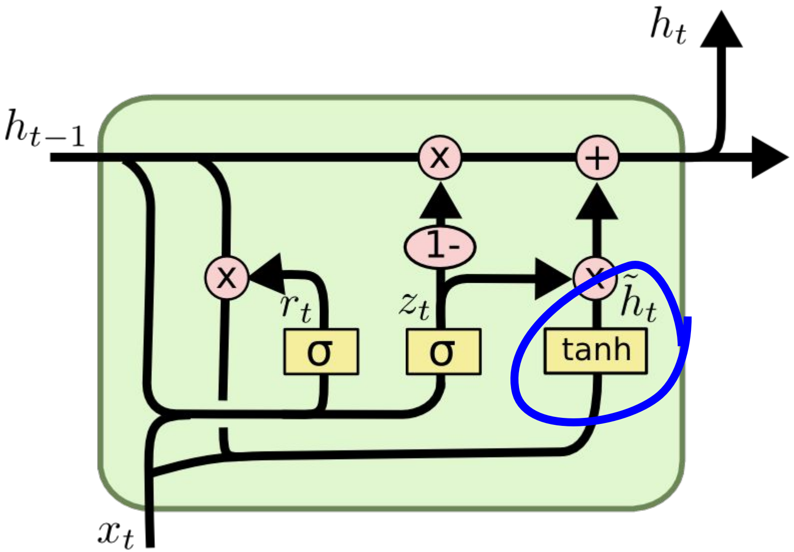
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Calculate update



$$z_t = \sigma \left(W_z \cdot [h_{t-1}, x_t] \right)$$

$$r_t = \sigma \, (W$$

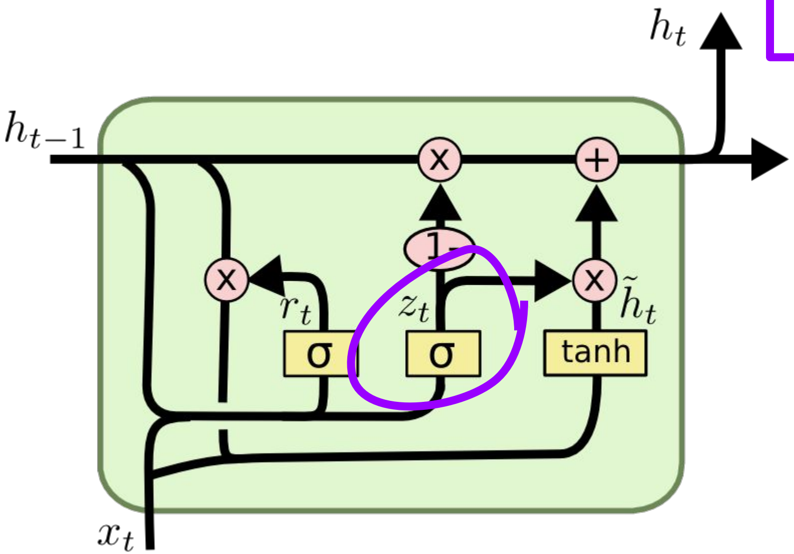
Считаем апдейт, используя инпут и важные участки памяти

$$\tilde{h}_t = \tanh \left(W \cdot [r_t * h_{t-1}, x_t] \right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Update gate: what to save/update



Считаем, какие участки памяти оставить, какие удалить

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

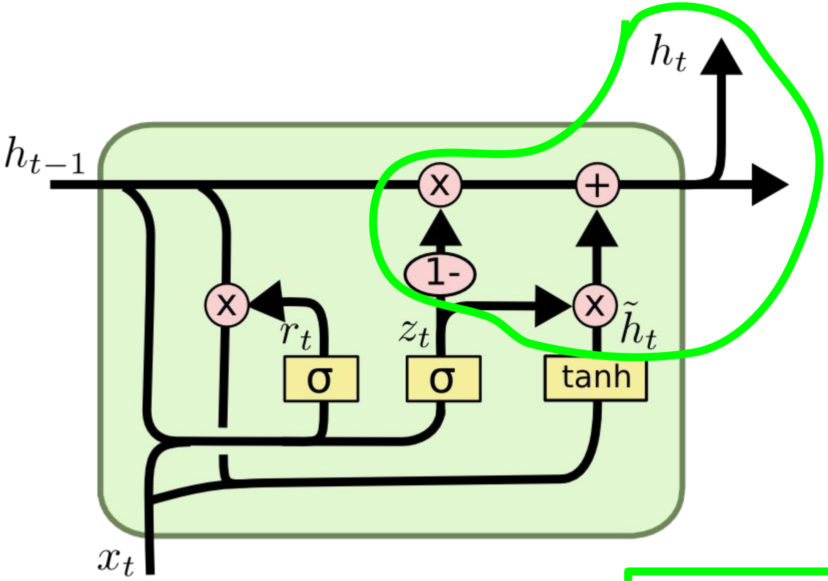
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Output: weighted sum



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

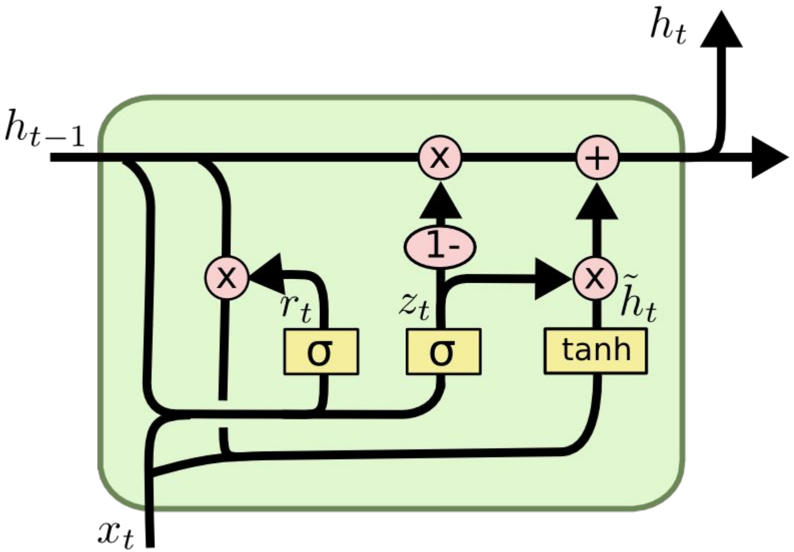
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Аутпут это взвешенная сумма предыдущего аутпута и нового аутпута

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

GRU: Gated recurrent units



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

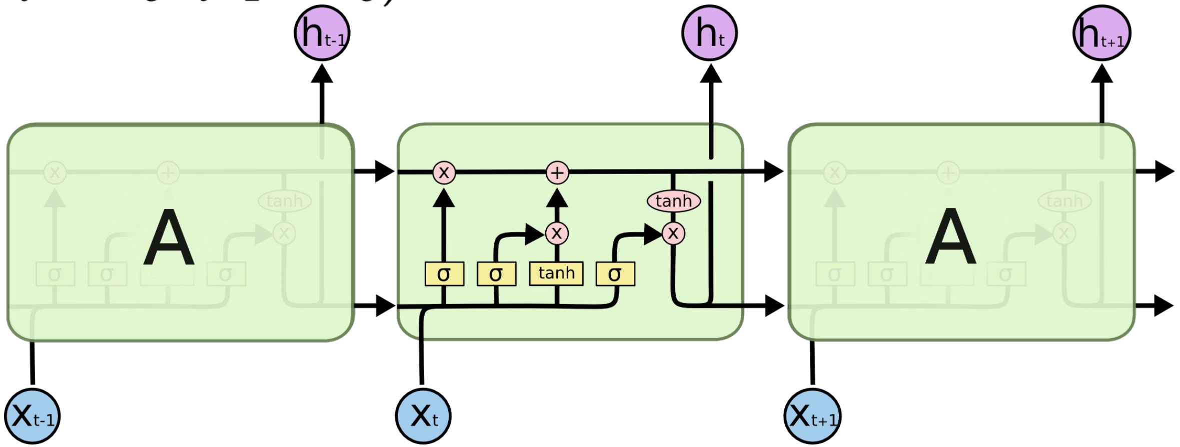
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

https://arxiv.org/pdf/1406.1078.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM: Long short-term memory

Букв становится все больше и больше...

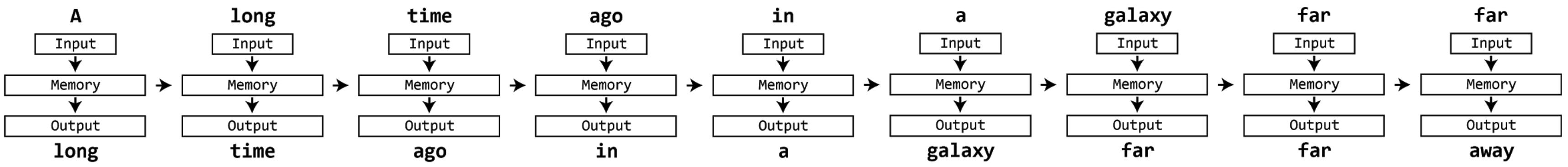
$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$h_t = o_t \circ \sigma_h(c_t)$$



http://www.bioinf.jku.at/publications/older/2604.pdf, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

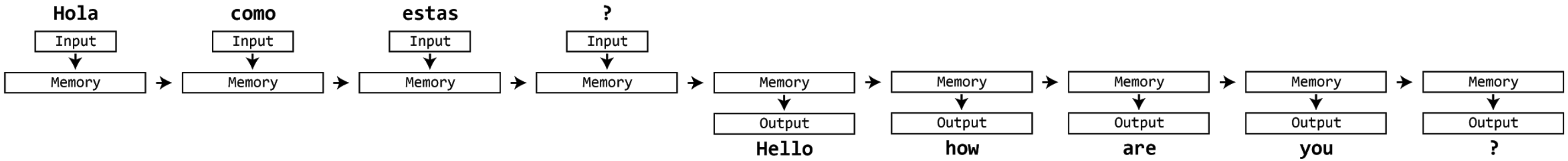
Language modeling: what’s next

Имея строку из слов/букв, определяем, что будет дальше



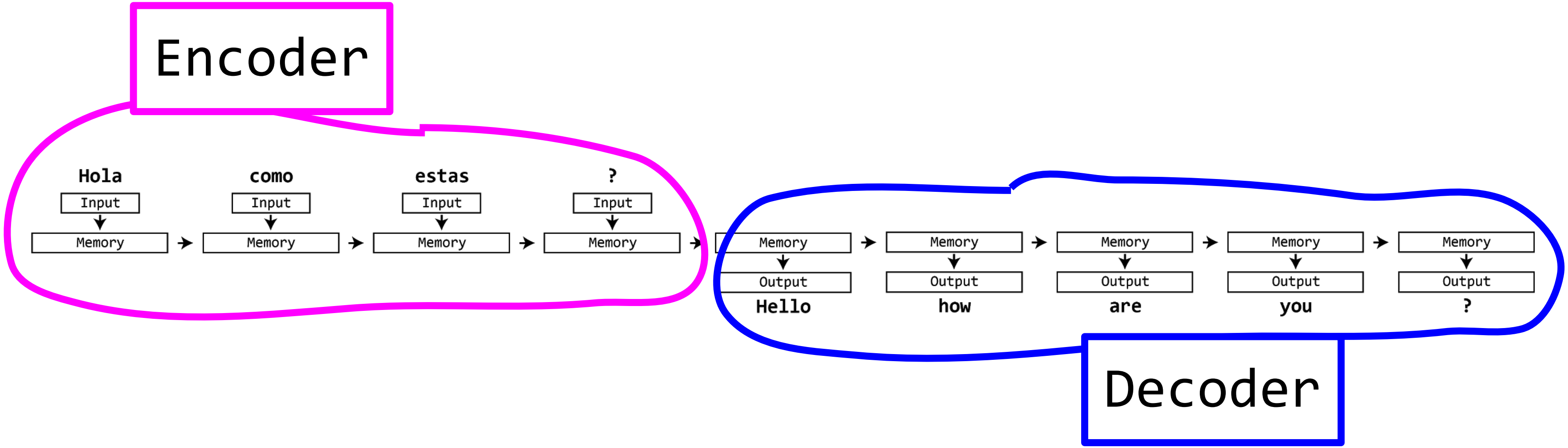
Translation and Q/A

Имея строку из слов/букв, рассчитываем ответ



Translation and Q/A

Имея строку из слов/букв, рассчитываем ответ



Translation and Q/A

Имея строку из слов/букв, рассчитываем ответ

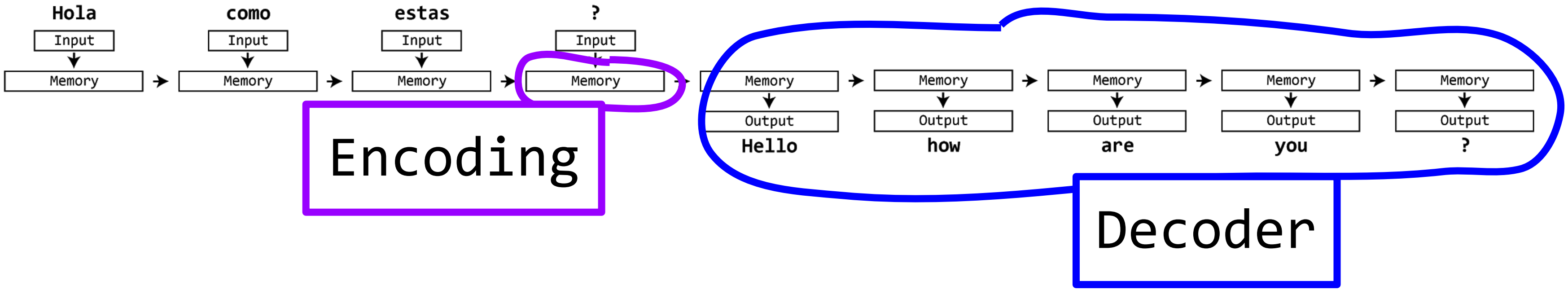


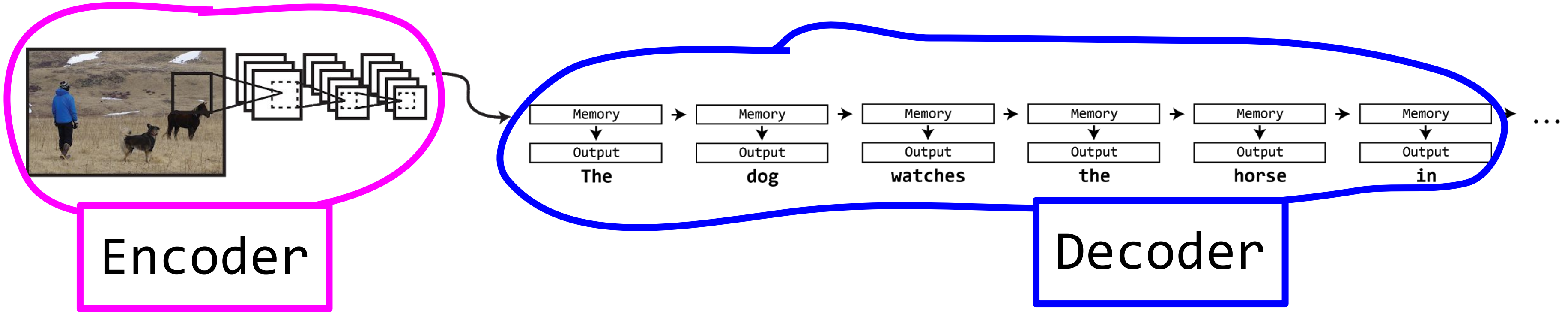
Image captioning

Имея изображения:

Извлекаем фичи с помощью CNN

Скармливаем фичи RNN

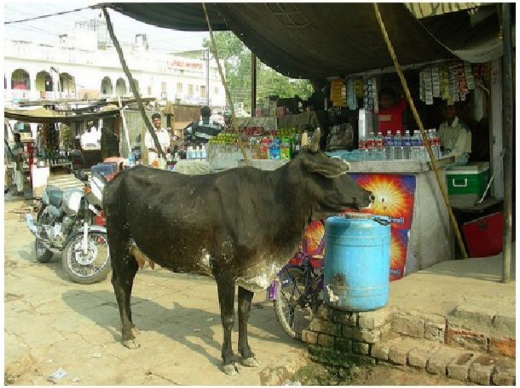
Генерируем предложение!



The Good

a group of people standing around a room with remotes
logprob: -9.17

a young boy is holding a baseball bat
logprob: -7.61



a cow is standing in the middle of a street
logprob: -8.84

a cat is sitting on a toilet seat
logprob: -7.79

a display case filled with lots of different types of donuts
logprob: -7.78

a group of people sitting at a table with wine glasses
logprob: -6.71

https://cs.stanford.edu/people/karpa

The Bad

a man standing next to a clock on a wall
logprob: -10.08

a young boy is holding a baseball bat
logprob: -7.65

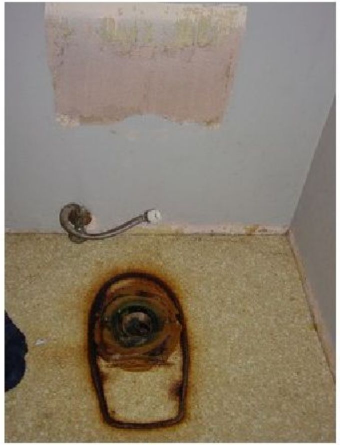
a cat is sitting on a couch with a remote control
logprob: -12.45

a baby laying on a bed with a stuffed bear
logprob: -8.66

a table with a plate of food and a cup of coffee
logprob: -9.93

a young boy is playing frisbee in the park
logprob: -9.52

The Ugly



a toilet with a seat up in a bathroom
logprob: -13.44

a woman holding a teddy bear in front of a mirror
logprob: -9.65

a horse is standing in the middle of a road
logprob: -10.34

https://cs.stanford.edu/people/karpathy/sfmltalk.pdf

kNN

Зачастую kNN подход оказывается самым лучшим подходом

Датасет очень большой, поэтому такой подход работает хорошо

Но есть другая проблема: как мы определяем, что такое “хорошое” описание изображения? Еще сложнее чем с детекцией или сегментацией.

Автоматические скоринг: BLEU, METEOR, etc.

https://arxiv.org/pdf/1505.04467.pdf

Visual Question Answering

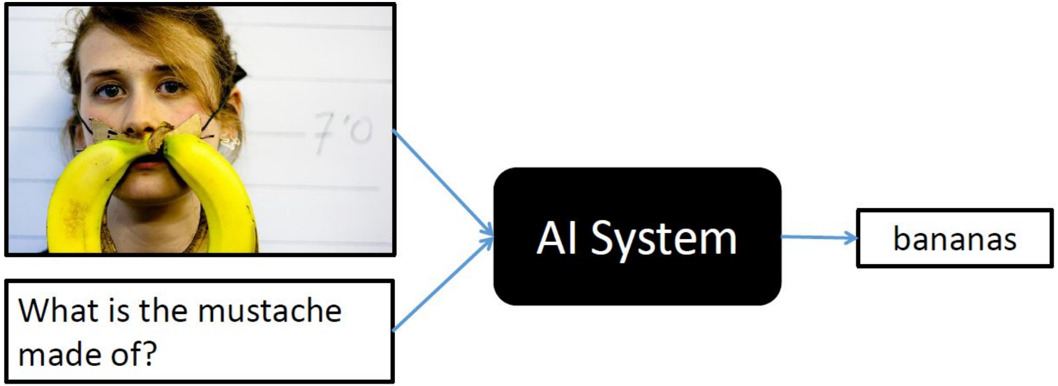
Имея изображение и вопрос:

Отвечаем на него!

Сложнее чем image captioning?

Требует больше понимания?

Зато легче оценивать!



http://www.visualqa.org/challenge.html

Let’s code