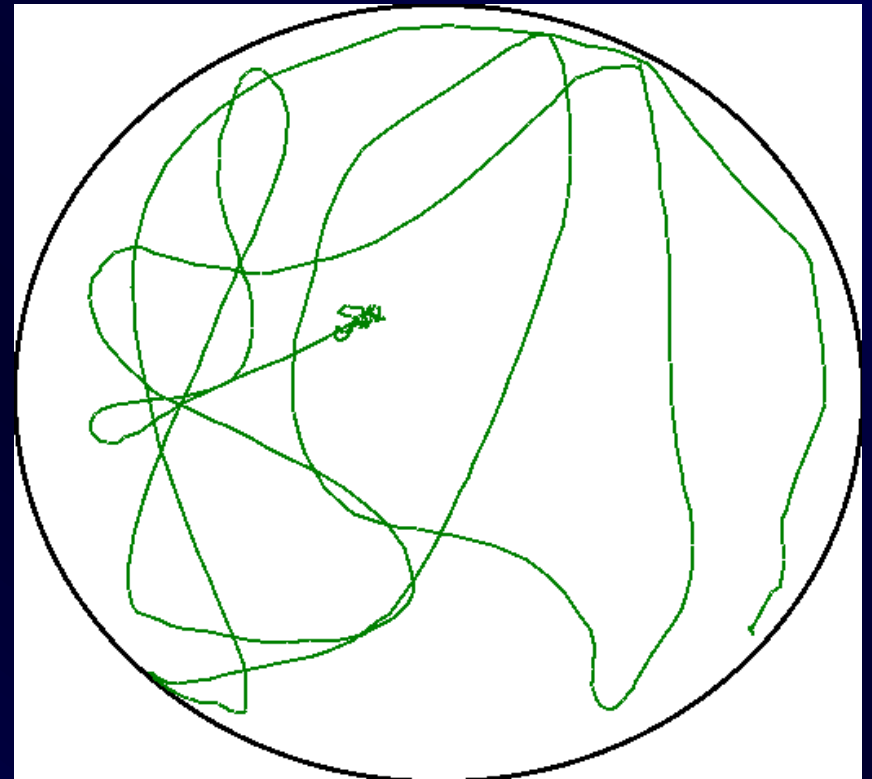


Анализ Видео

Из прошлой лекции...

- Локальные особенности
- Дескрипторы
- Скользящее окно
- Градиенты

Трассировка



Работа с видео OpenCV

```
cv::VideoCapture cap;  
//Открыть файл  
cap.open('filename');  
cv::Mat img;  
//Загрузить в память изображение  
cap >> img;
```

Работа с видео Matlab

//Открыть файл и получить указатель

```
hReader = mmreader('filename');
```

//Кол-во кадров в файле

```
numFrames = get(hReader, 'numberOfFrames');
```

//Прочитать i-ый кадр

```
x = read(hReader,i);
```

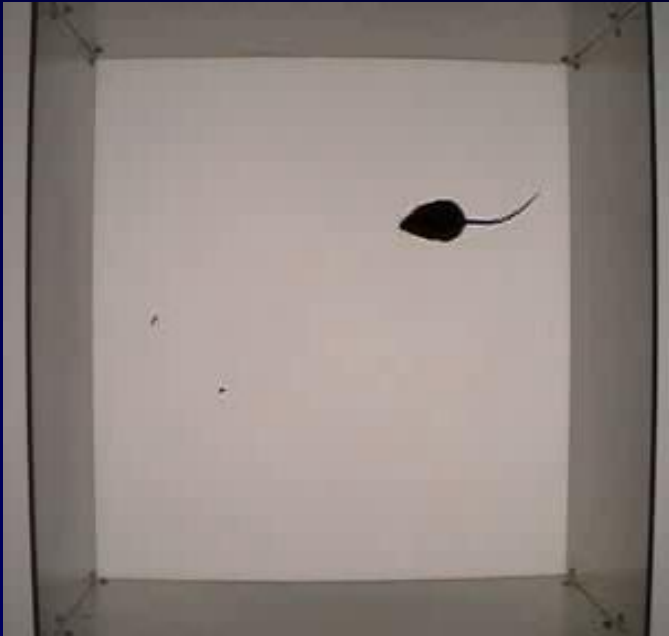
ВАЖНО: Необходимо чтобы были установлены нужные кодеки. Есть бесплатные наборы кодеков например
K-Lite codec pack.

Задача

- Наблюдение за объектом
- Захват объекта с помощью ограничивающего прямоугольника или с помощью попиксельной маске
- Трассировка. Вход первое положение объекта, выход -положение объекта в других кадрах

Система наблюдения

1. Неподвижная камера
2. Постоянный фон

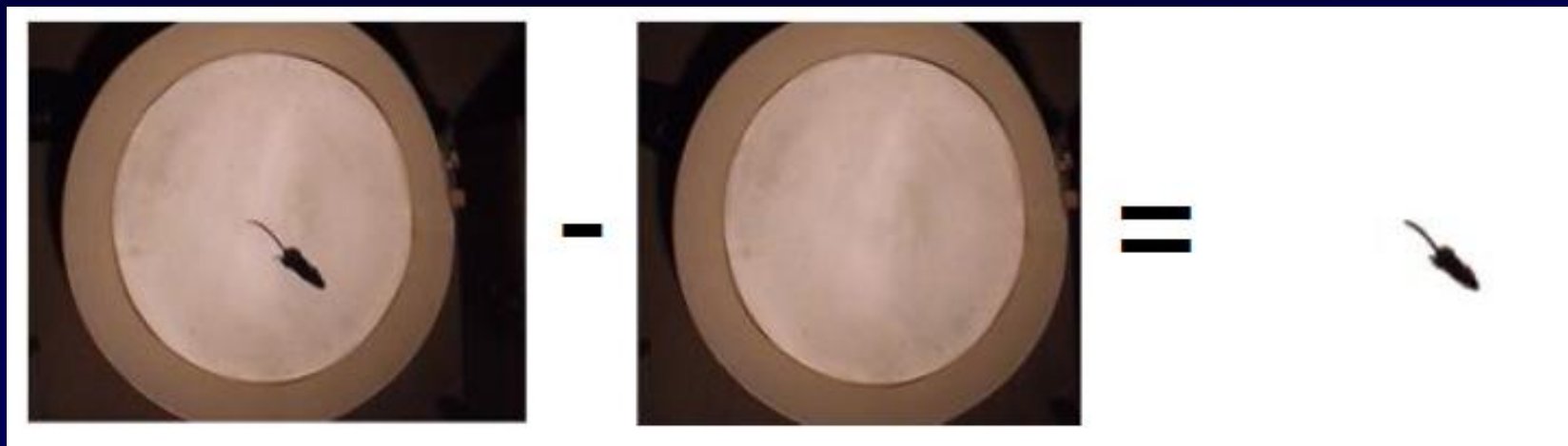


Простой алгоритм трассировки

- Вычитание фона:
 - Взяли снимок без объектов, только фон
 - Из снимка с объектом вычитаем фон
 - Строим передний план
 - Считаем параметры связанных компонент
 - Фильтрация медианная или морфологическая

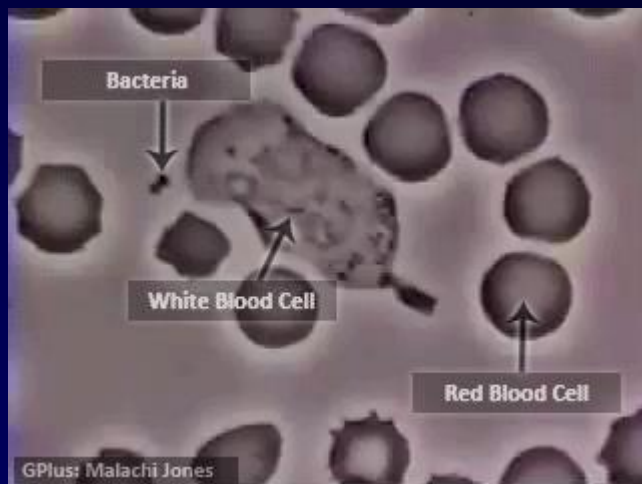
Пример животные

- Идеальные лабораторные условия



Пример эритроциты

- Движущийся фон
- Множество объектов



Реальные условия

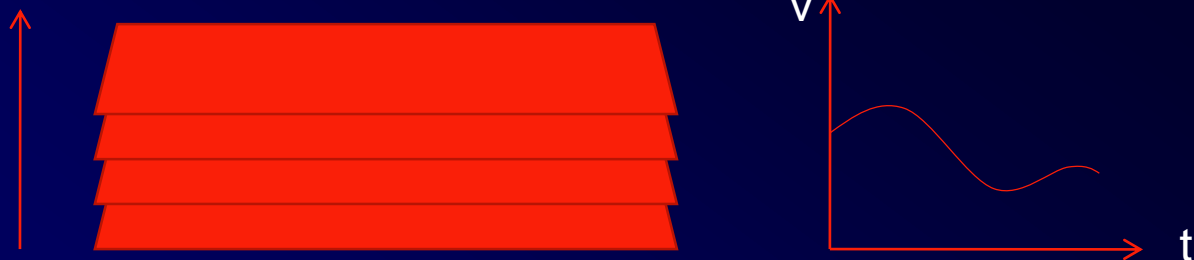
- Фон существенно меняется



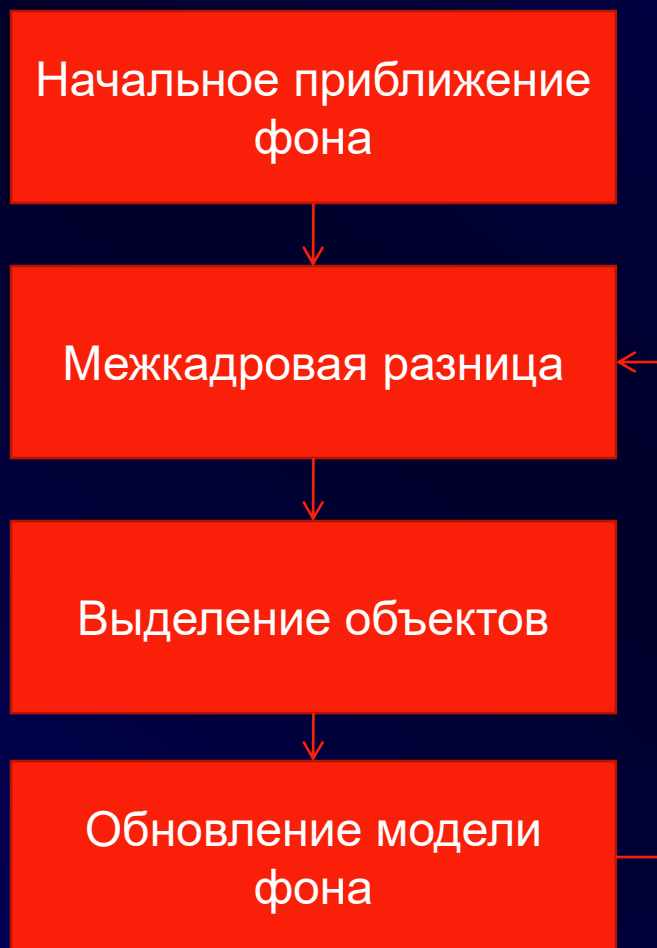
Slide credit: Антон Конушин

Изменения яркости

- У каждого пикселя индивидуальная функция яркости от времени
- Строим модель фона
- Если пиксель пришедший не удовлетворяет модели фона, значит он относится к переднему плану



Блок схема алгоритма



Среднее изображение фон

- Простейшая модель фона – одно изображение чистого фона
- Возьмем N кадров и попиксельно усредним интенсивности (цвета)

$$I_0(x, t) = \frac{1}{T} \sum_{t=1}^T I(x, t)$$



Медианная модель фона



- Когда не работает усреднение:
 - Всегда в кадре движущиеся объекты
 - Случайные и резкие изменения яркости (блики, засветка)
- Вместо усреднения берем медиану

Как построить медиану для 1000 кадров?

Медианная фильтрация или среднее



Средний цвет



Медиана

Обновление фона

- Обновление матожидания

$$\mu_{t+1} = \alpha\mu_t + (1 - \alpha)x_{t+1}$$

- Обновление дисперсии

$$\sigma_{t+1}^2 = \alpha(\sigma_{t+1}^2 + (\mu_{t+1} - \mu_t)^2) + (1 - \alpha)(x_{t+1} - \mu_{t+1})^2$$

α – скорость обновления (обучения)

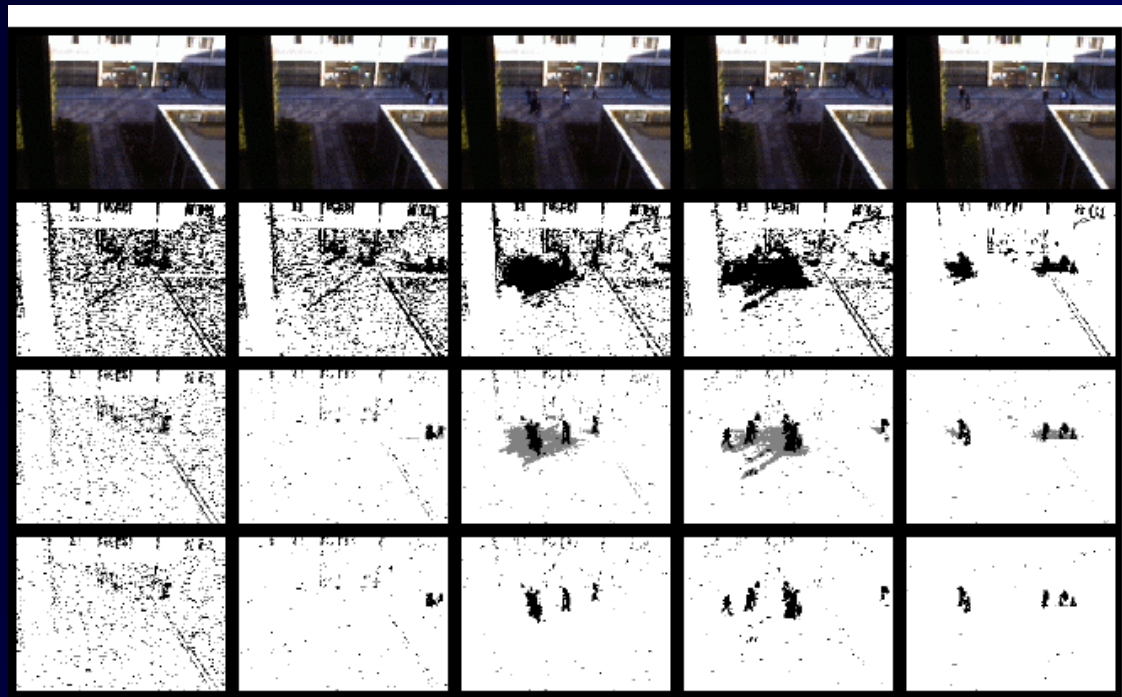
Модуль шума в точке изображения

- В точке шум в яркости можно описать с помощью нормального распределения (среднего значения и дисперсии)
- Тогда вероятность что пиксель принадлежит фону будет равна:

$$\rho(x | \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Модель фона как смесь гауссианов

BackgroundSubtractorMOG



P. KadewTraKuPong and R. Bowden, *An improved adaptive background mixture model for real-time tracking with shadow detection*, Proc. 2nd European Workshop on Advanced Video-Based Surveillance Systems, 2001

Трассировка множества объектов



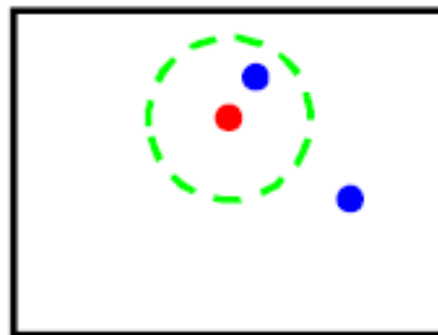
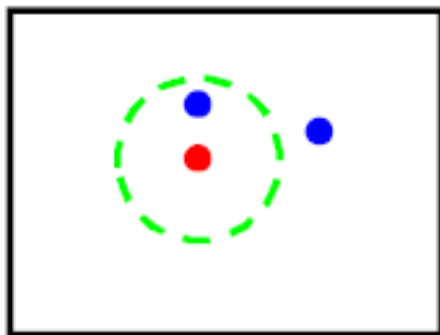
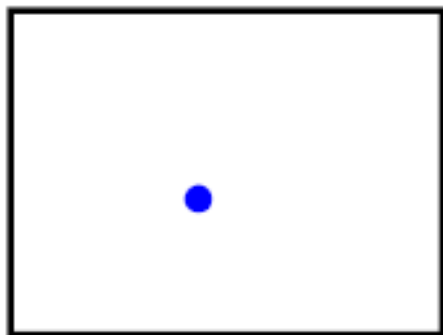
Сопровождение объектов



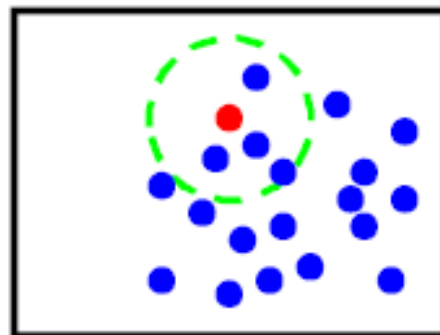
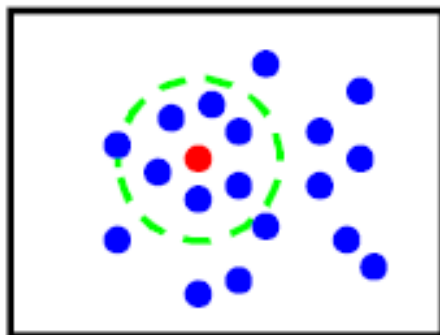
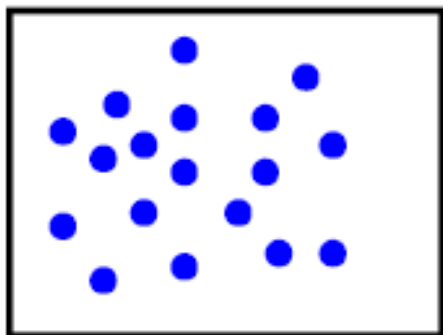
Сопровождение объектов

- Выделили «объекты интереса» на каждом кадре с помощью вычитания фона в виде набора блобов
- Какому объекту (следу) какой блоб соответствует?
- Фактически – задача сопоставления или ассоциации данных (Data association)
- «Tracking by detection»
- Можем каждый объект описать набором признаков

Простейшая стратегия

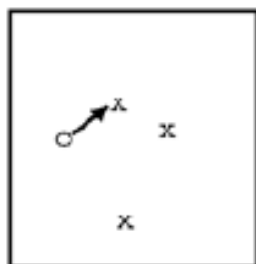


Сопоставим ближайшее наблюдение следу

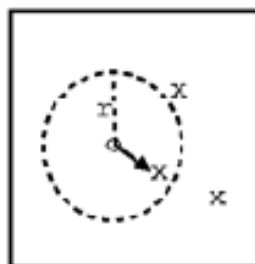


Простейшая стратегия в более сложных случаях не срабатывает, поэтому исследовались разные ограничения

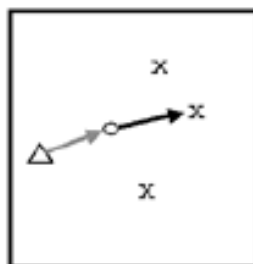
Ограничения



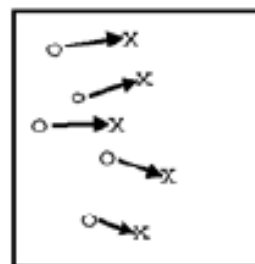
(a)



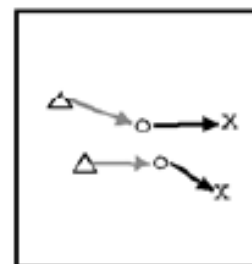
(b)



(c)



(d)



(e)

- (a) близость
- (b) максимальная скорость
- (c) малое изменение вектора скорости
- (d) общее движение
- (e) «жесткость»

Slide credit: Антон Конушин

Трассировка через обнаружения



Что делать если фон изменяется?



Перерыв

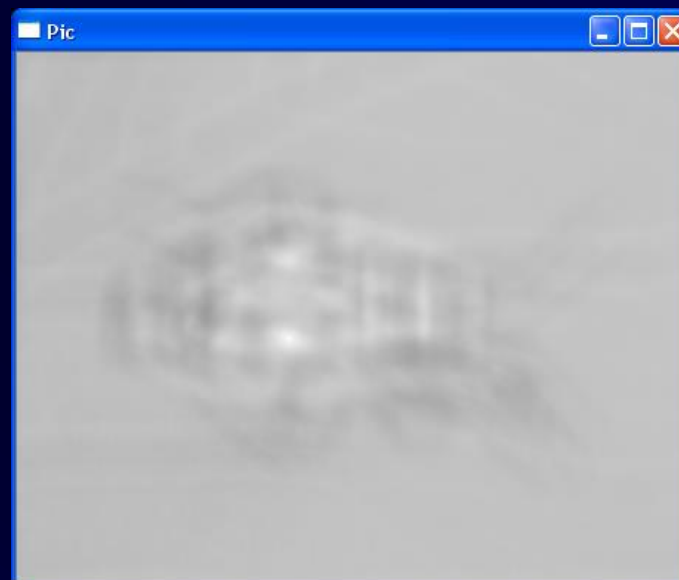
- Вопрос?

Функции правдоподобия

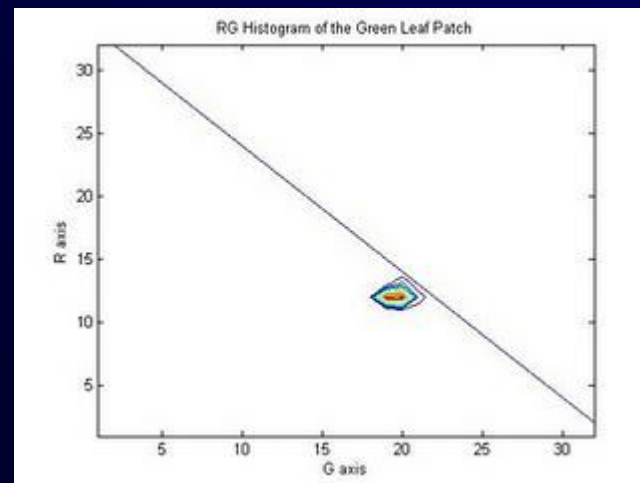
- Функции дающие ответ на вопрос насколько вероятно что точка принадлежит объекту

$$L(\mathbf{x} \mid \theta) = \sum_{i=1}^n \ln f_X(x_i \mid \theta)$$

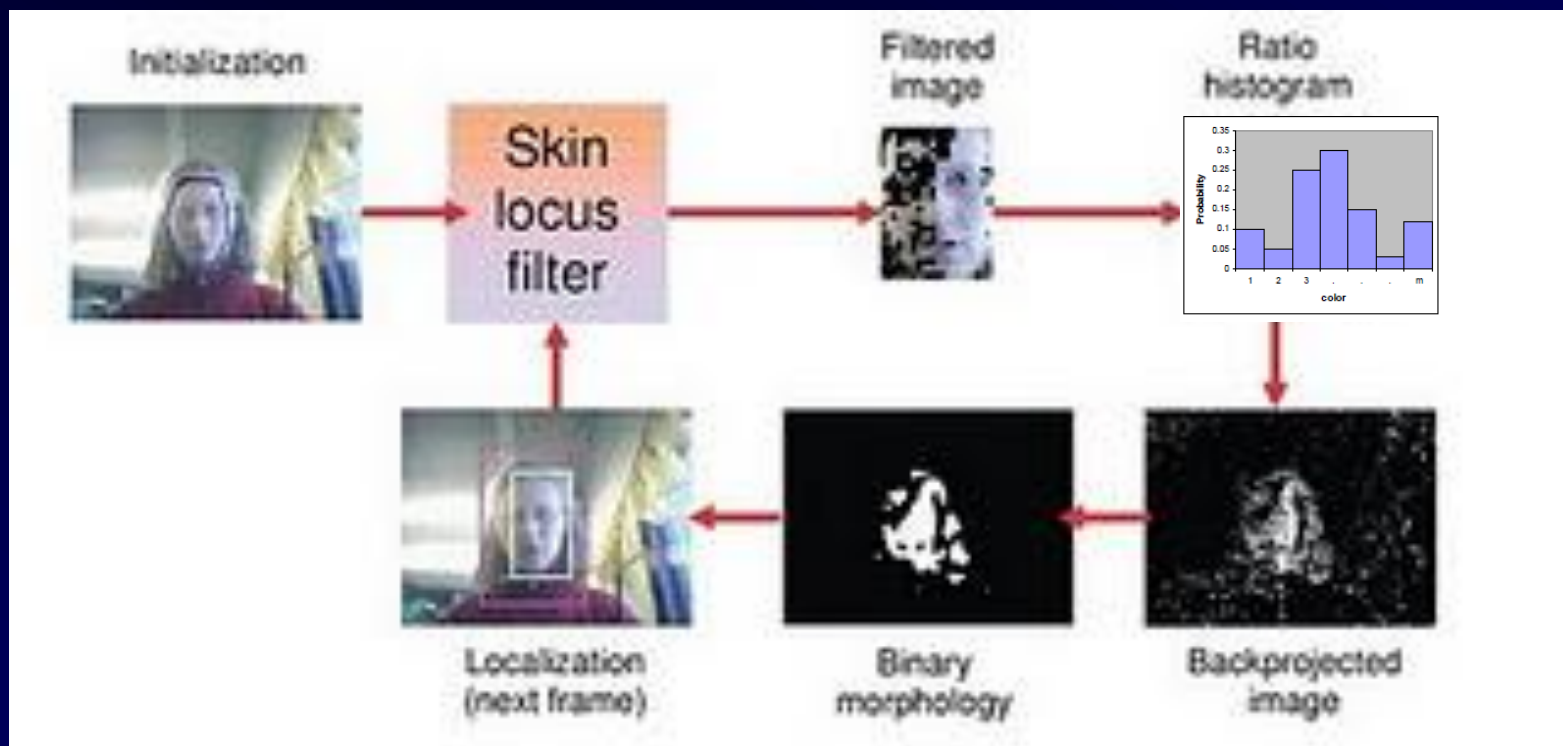
Прямое сопоставление (корреляция)



Обратная проекция гистограммы



Трассировка смещением среднего



Трассировка смещением среднего

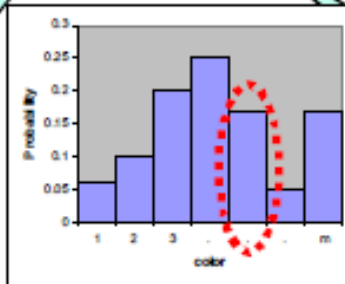
Вероятность признака в модели

Вероятность признака в изображении

$$q_u = C \sum_{b(x_i)=u} k(\|x_i\|^2)$$

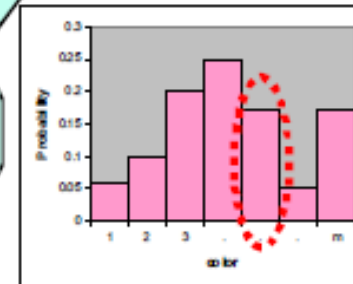
$$p_u(y) = C_h \sum_{b(x_i)=u} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)$$

Нормализация



Вес пиксела

Нормализация

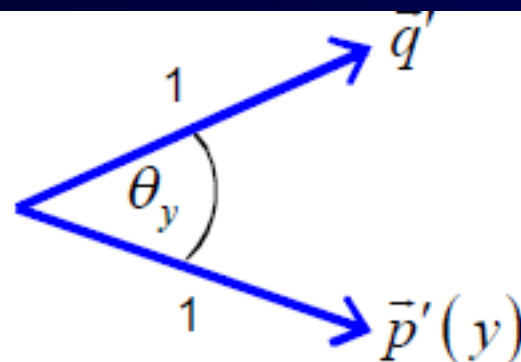


Вес пиксела

Расстояние между нормированными векторами

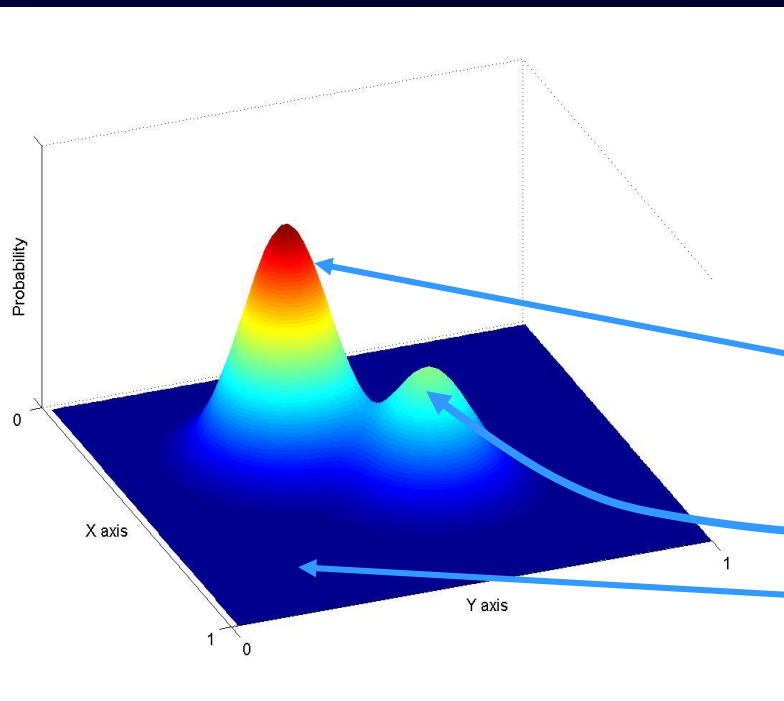
$$\bar{q}' = (\sqrt{q_1}, \dots, \sqrt{q_m})$$

$$\bar{p}'(y) = (\sqrt{p_1(y)}, \dots, \sqrt{p_m(y)})$$

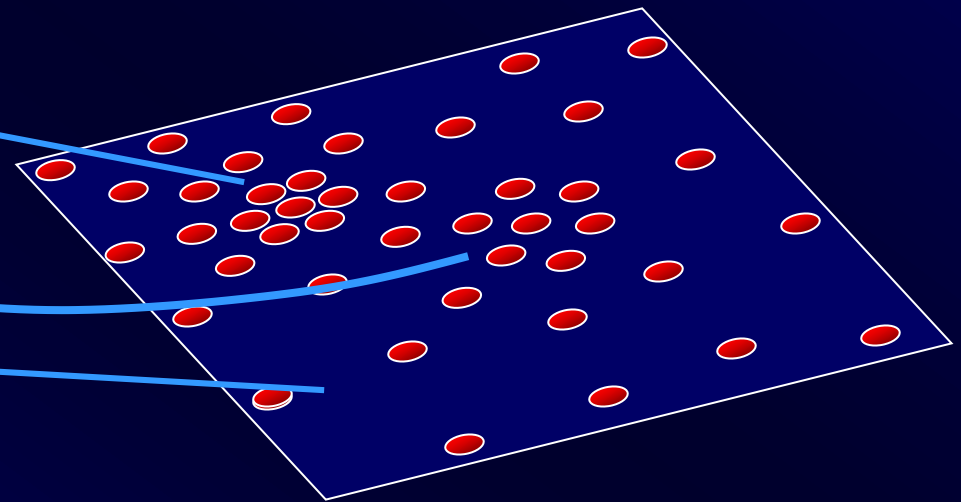


$$f(y) = \cos \theta_y = \frac{p'(y)^T q'}{\|p'(y)\| \cdot \|q'\|} = \sum_{u=1}^m \sqrt{p_u(y) q_u}$$

Переход к непрерывной модели



Прогнозированная плотность

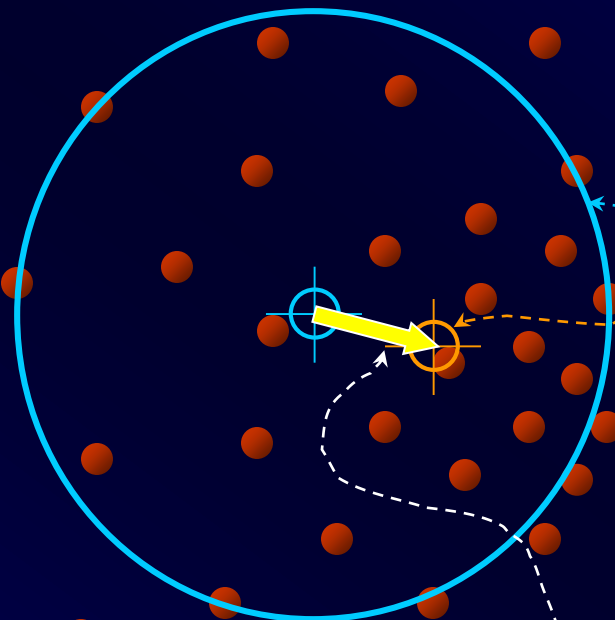


Реальные измерения

Интересующий
регион

Центр масс

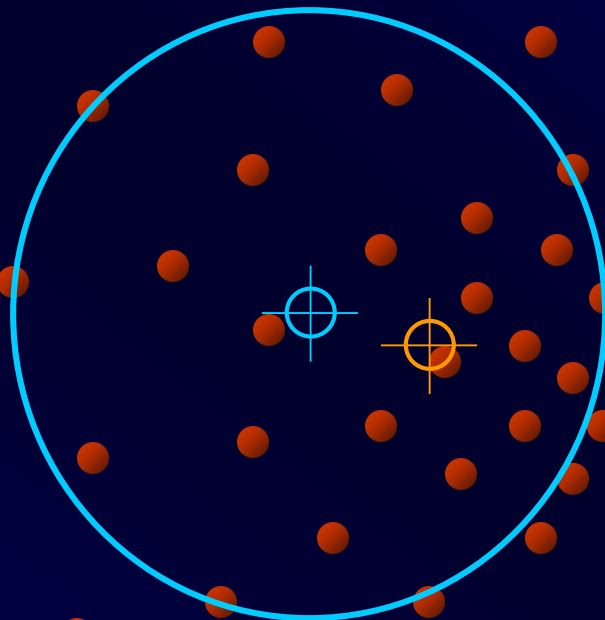
Смещение



Интересующий
регион

Центр масс

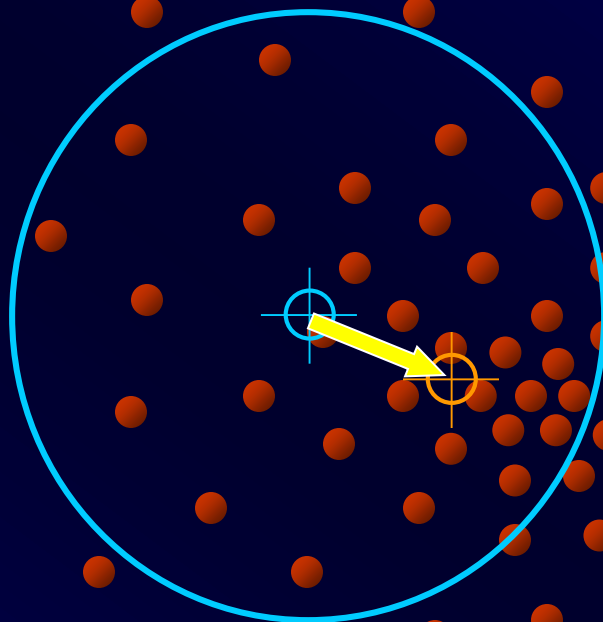
Смещение



Интересующий
регион

Центр масс

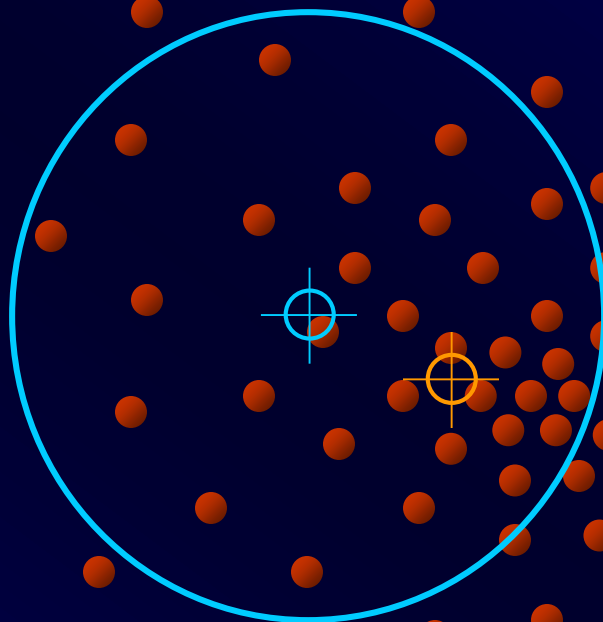
Смещение



Интересующий
регион

Центр масс

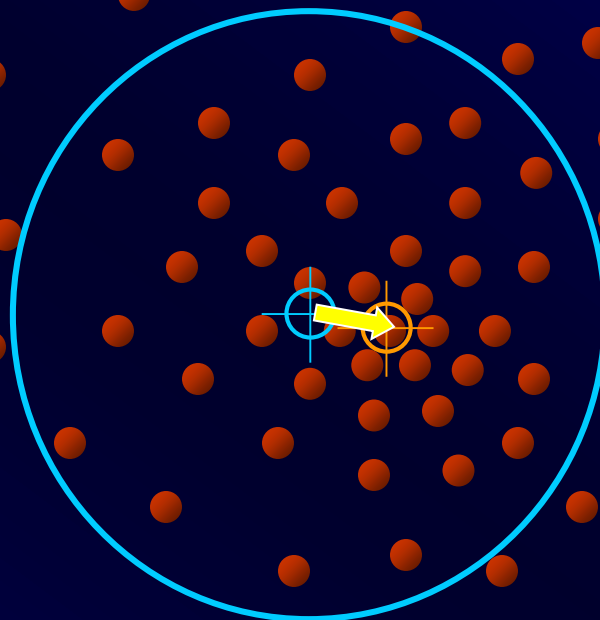
Смещение



Интересующий
регион

Центр масс

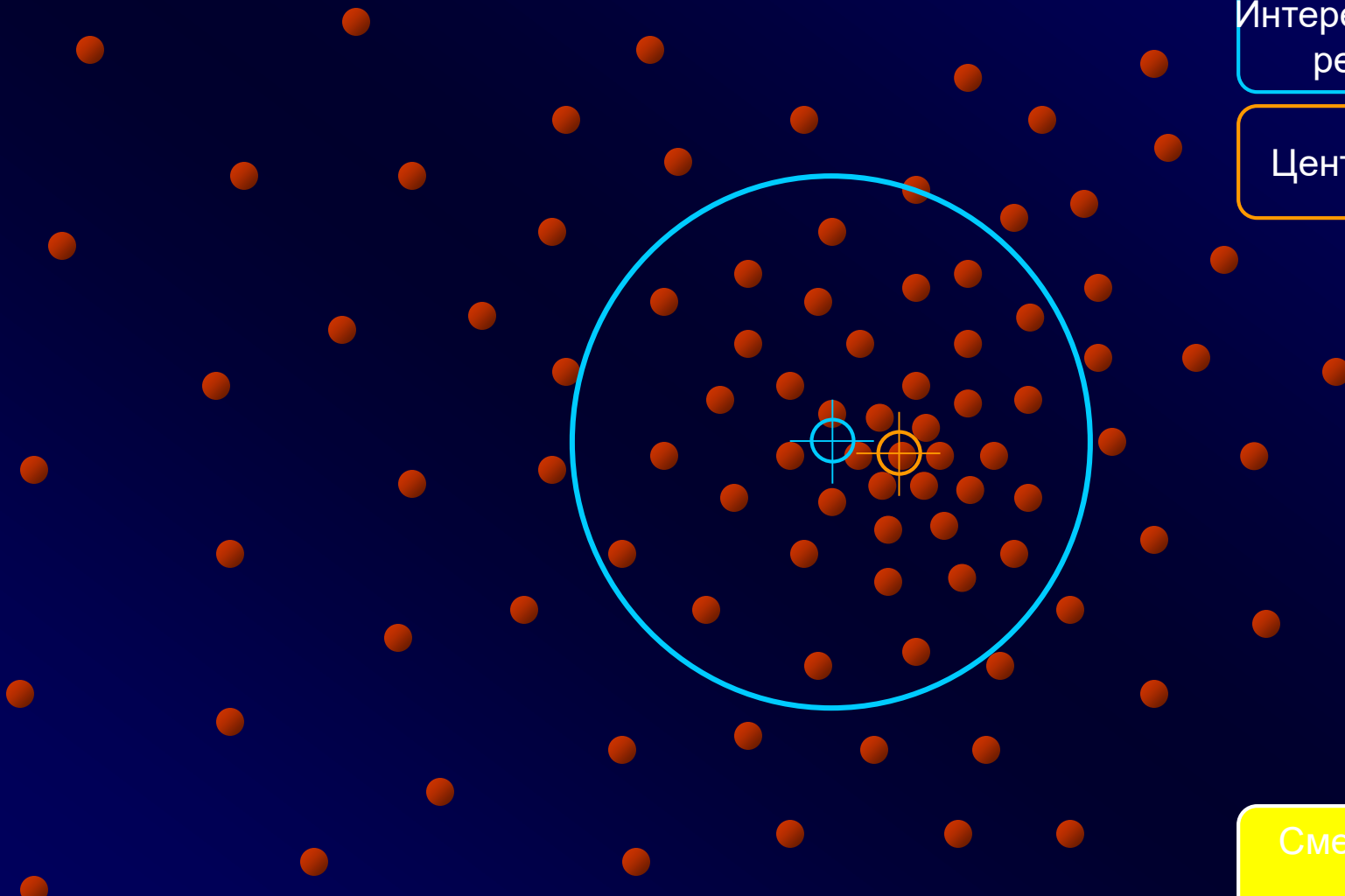
Смещение



Интересующий
регион

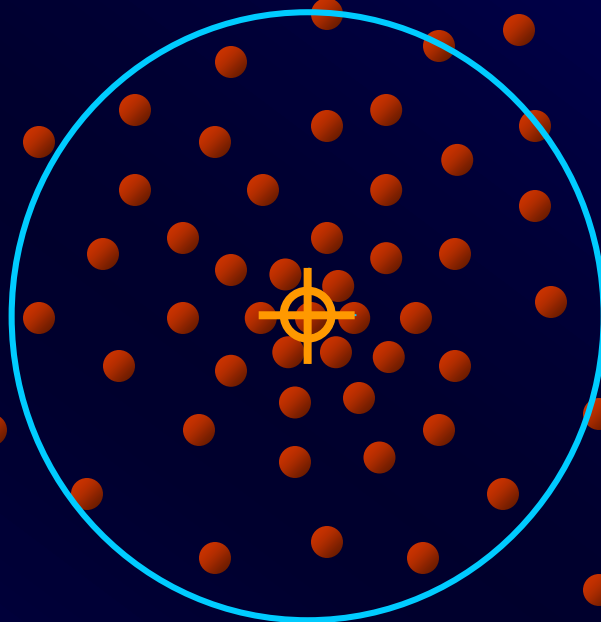
Центр масс

Смещение

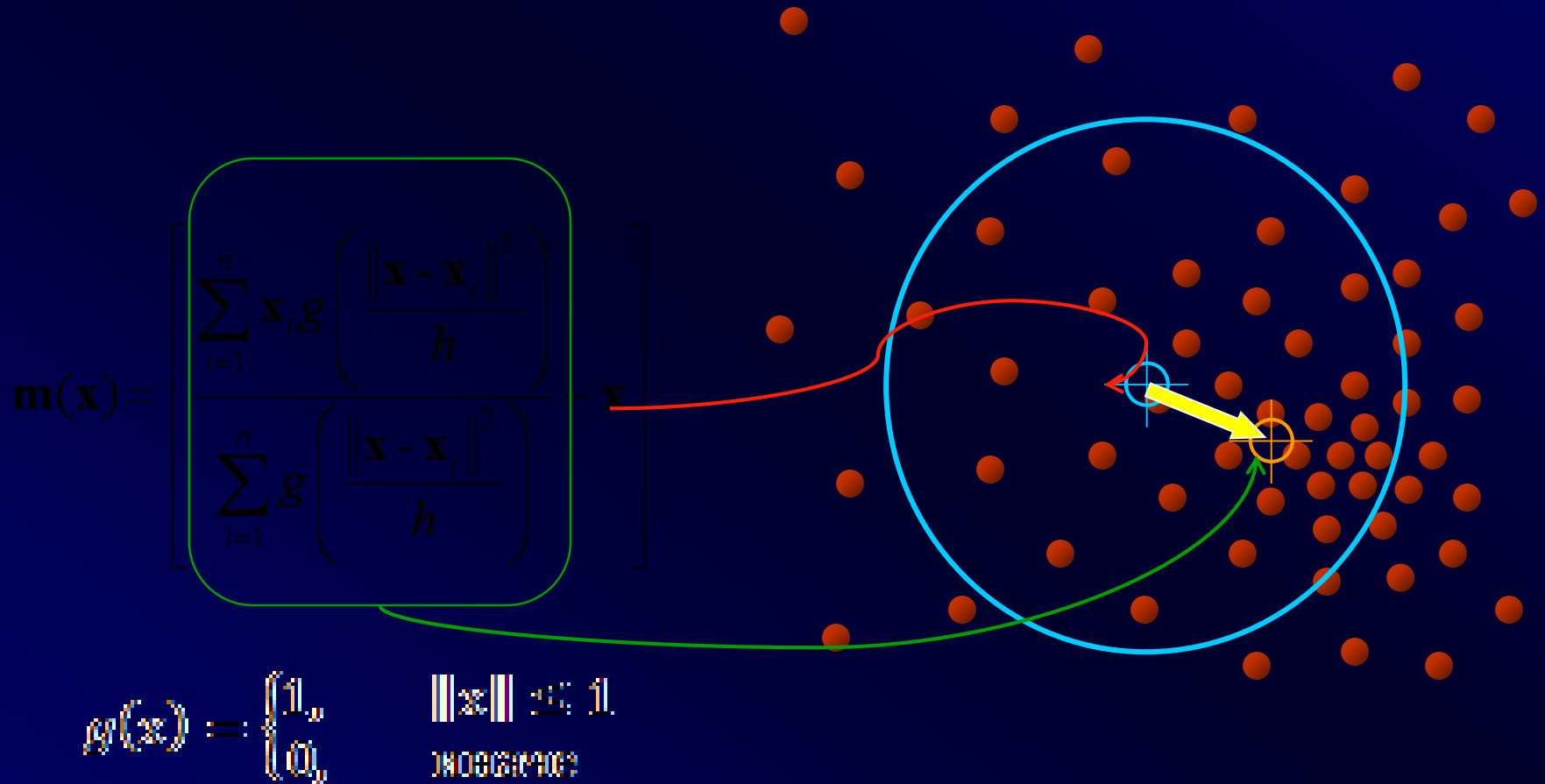


Интересующий
регион

Центр масс



Трассировка смещением среднего



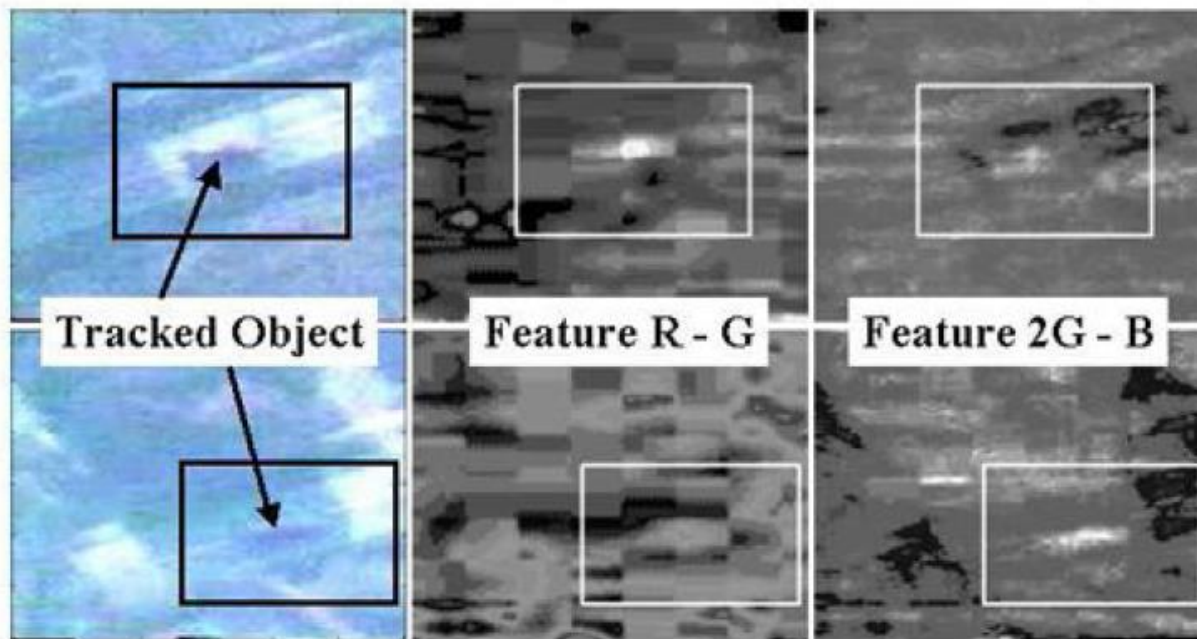
Реализация в OpenCV

- `int meanShift(InputArray probImage, Rect& window, TermCriteria criteria)`
- `probImage` – изображение правдоподобия
- `Window` – выходное окно описывающее объект
- `Criteria` – критерий остановки

Пример

- <http://www.youtube.com/watch?v=vGDy20VxwAA&feature=related>

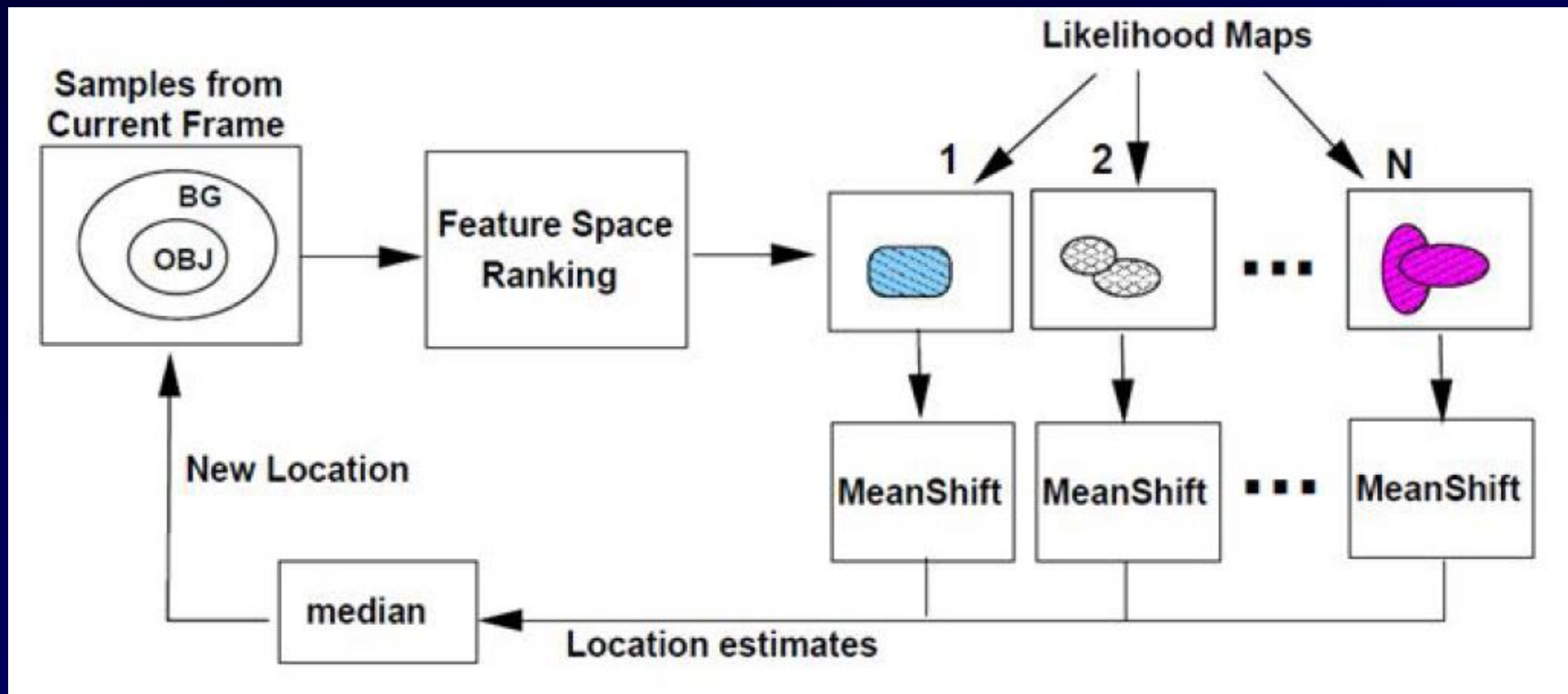
Выбор признаков



- В зависимости от разных факторов, объект может быть различим в разных каналах, по разным признакам

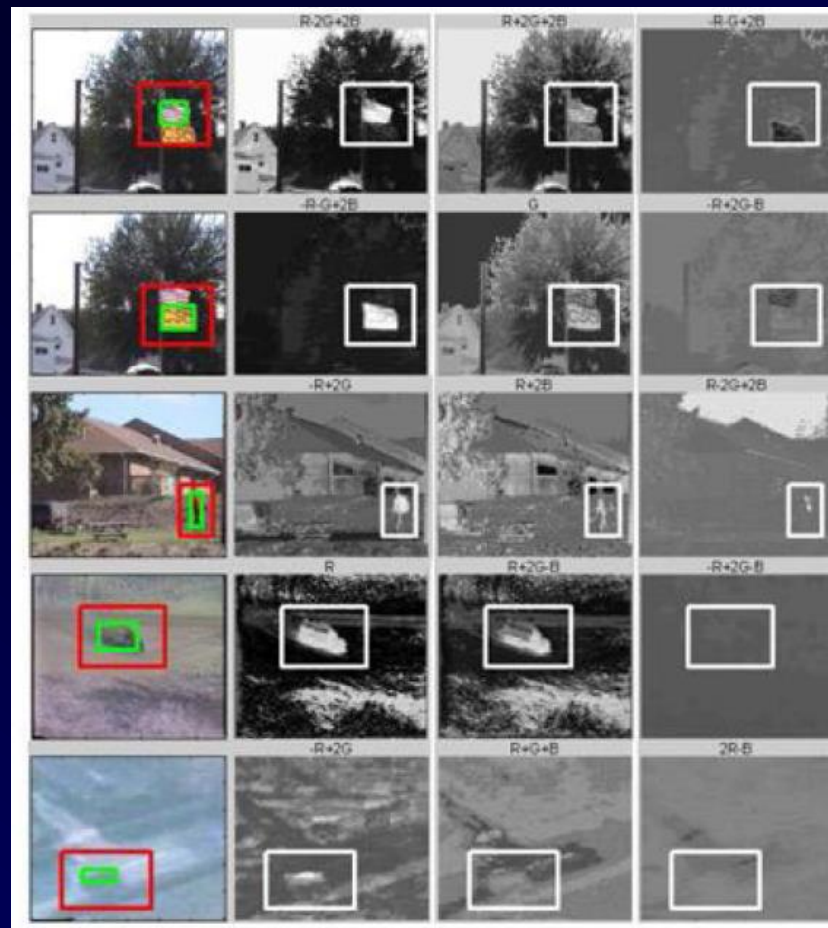
COLLINS, R. AND LIU, Y.. On-line selection of discriminative tracking features. ICCV 2003

Выбор признаков



Slide credit: Антон Конушин

$$F_1 \equiv \{w_1R + w_2G + w_3B \mid w_* \in [-2, -1, 0, 1, 2]\}$$



Predator (совмещение обучения с трассировкой)

- <http://www.youtube.com/user/ekalic2>



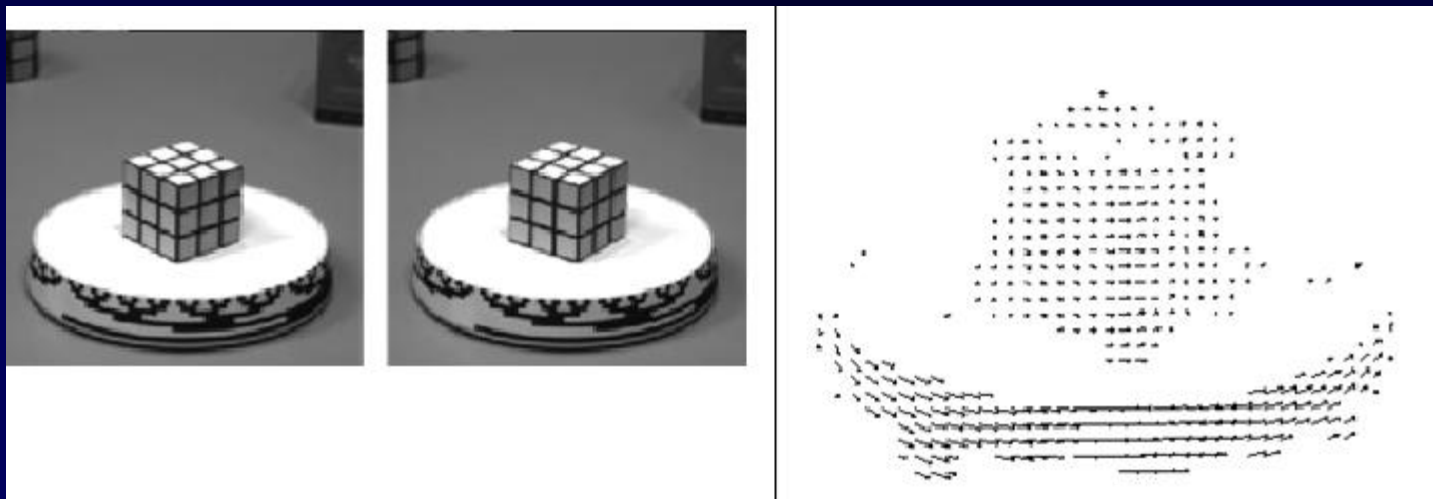
Zdenek Kalal et al. Forward-Backward Error: Automatic Detection of Tracking Failures , ICPR 2010

Итог по трассировке

- Что лучше обнаружение и сопоставление или трассировка?

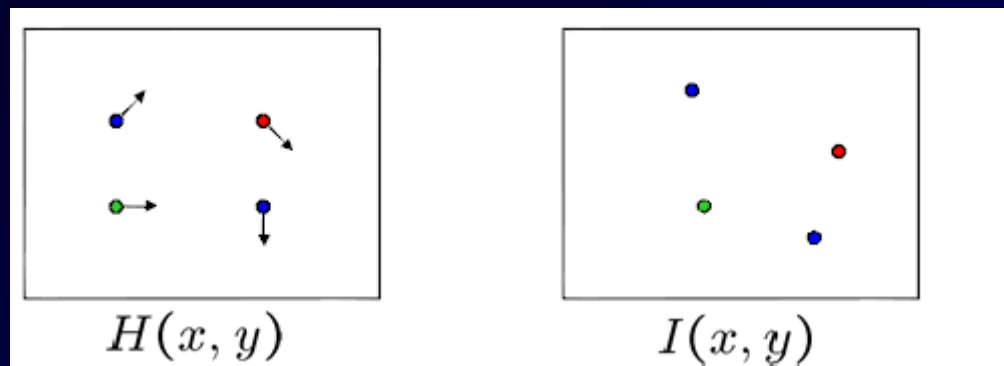
Оптический поток

- Векторное поле движения пикселей между кадрами
- Задача - аналог задачи сопоставления изображений (dense matching)
- Один из базовых инструментов анализа изображений



Спасибо. Антон Конушин

Задача



Как оценить движение пикселей от H в изображение I ?

Пусть дан пиксель H , найти близкие пиксели того же цвета в I

- **Константный цвет:** точка в H выглядит также, как и в I
 - Для изображения в градациях серого, это постоянная яркость
- **Малое движение:** точки не уезжают далеко между кадрами

Slide credit: Антон Конушин

Уравнение оптического потока

- Объединим два ограничения

$$\begin{aligned} 0 &= I(x + u, y + v) - H(x, y) & I_x &= \frac{\partial I}{\partial x} \\ &\approx I(x, y) + I_x u + I_y v - H(x, y) \\ &\approx (I(x, y) - H(x, y)) + I_x u + I_y v \\ &\approx I_t + I_x u + I_y v \\ &\approx I_t + \nabla I \cdot [u \ v] \end{aligned}$$

В пределе u и v стремятся к нулю, и получаем равенство:

$$0 = I_t + \nabla I \cdot \left[\frac{\partial x}{\partial t} \ \frac{\partial y}{\partial t} \right]$$

Сколько неизвестных и уравнений для каждого пикселя?

Slide credit: Антон Конушин

Дополнительные ограничения

- Как можно получить больше уравнений?
 - Пусть оптический поток меняется плавно
 - Пусть для всех пикселей p из окрестности (x,y) смещение (u,v) постоянно

$$\begin{array}{c} \left[\begin{array}{cc} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{array} \right] \begin{array}{c} \left[\begin{array}{c} u \\ v \end{array} \right] \end{array} = - \begin{array}{c} \left[\begin{array}{c} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{array} \right] \end{array} \\ \begin{array}{ccc} A & d & b \\ 25 \times 2 & 2 \times 1 & 25 \times 1 \end{array} \end{array}$$

- В чем теперь проблема?

Алгоритм Лукаса-Канаде

- Проблема: больше уравнений, чем неизвестных!

$$\underset{25 \times 2}{A} \underset{2 \times 1}{d} = \underset{25 \times 1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

- Получаем задачу наименьших квадратов
- Можем решить её через нормальные уравнения

$$\underset{2 \times 2}{(A^T A)} \underset{2 \times 1}{d} = \underset{2 \times 1}{A^T b} \quad \Rightarrow \quad d = (A^T A)^{-1} A^T b$$

$$\underset{A^T A}{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}} \begin{bmatrix} u \\ v \end{bmatrix} = - \underset{A^T b}{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}$$

- Суммируем по всем пикселям в окне $K \times K$
- Это метод был предложен Лукасом и Канаде в 1981 году

Slide credit: Антон Конушин

Условия на разрешимость

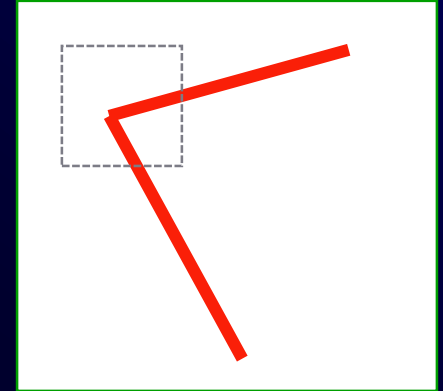
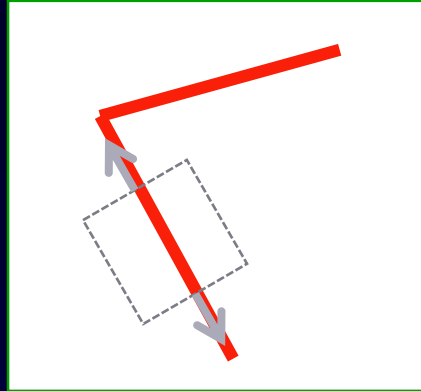
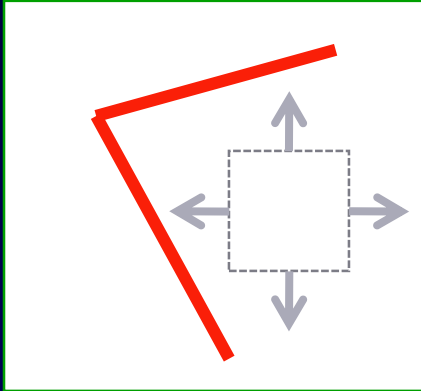
- Решение задачи оптического потока $d = (u, v)$ может быть найдено в виде

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T$$

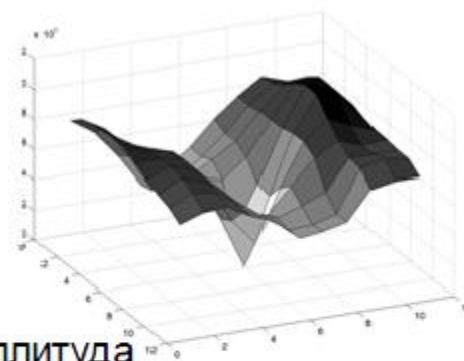
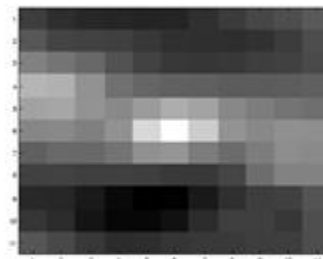
$$d = (A^T A)^{-1} A^T b \qquad A^T b = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

- Когда задача разрешима?
 - $A^T A$ должна быть обратимой
 - $A^T A$ не должна быть слишком близка к нулю
 - С.значения λ_1 и λ_2 матрицы $A^T A$ не должны быть малы
 - $A^T A$ должна быть хорошо определима
 - λ_1 / λ_2 не должно быть слишком велико
 - (λ_1 = наибольшее с.значение)
- $A^T A$ разрешима, когда нет апертурной проблемы

Slide credit: Антон Конушин



Контрастная область



$$\sum \nabla I (\nabla I)^T$$

- градиенты разные, большая амплитуда
- большое λ_1 , большое λ_2

Slide credit: АНТОН Конушин

Реализация

- GoodFeatureToTrack
 - Выбор особых точек (фактически, Harris)
- cvCalcOpticalFlowPyrLK
 - Иерархическое расширение метода Lucas-Kanade для оптического потока

Оценка движения



Optical Flow

- Оптический поток между парой изображений есть векторное поле, задающее естественную трансформацию первого изображения во второе.



Optical Flow

Применение

1. Видеоморфинг и другие спецэффекты
2. Компрессирование видеоинформации
3. Машинное зрение
4. Получение и визуализация научно-технических данных
5. Другие приложения

Оценка оптического потока

Последовательности упорядоченных изображений позволяют оценивать движение либо как мгновенную скорость изображения, либо как дискретное смещение.

Анализ методов вычисления оптического потока проведен в работе John L. Barron, David J. Fleet и Steven Beauchemin. Они рассматривают методы как с точки зрения точности, так и с точки зрения плотности получаемого векторного поля.

Методы, основанные на оптическом потоке, вычисляют движение между двумя кадрами, взятыми в момент времени t и $t + \delta t$, в каждом пикселе. Эти методы называются дифференциальными, так как они основаны на приближении сигнала отрезком ряда Тейлора; таким образом, они используют частные производные по времени и пространственным координатам.

Оценка оптического потока

В случае размерности $2D+t$ (случаи большей размерности аналогичны) пиксель в позиции (x, y, t) с интенсивностью $I(x, y, t)$ за один кадр будет перемещен на δx , δy и δt ,

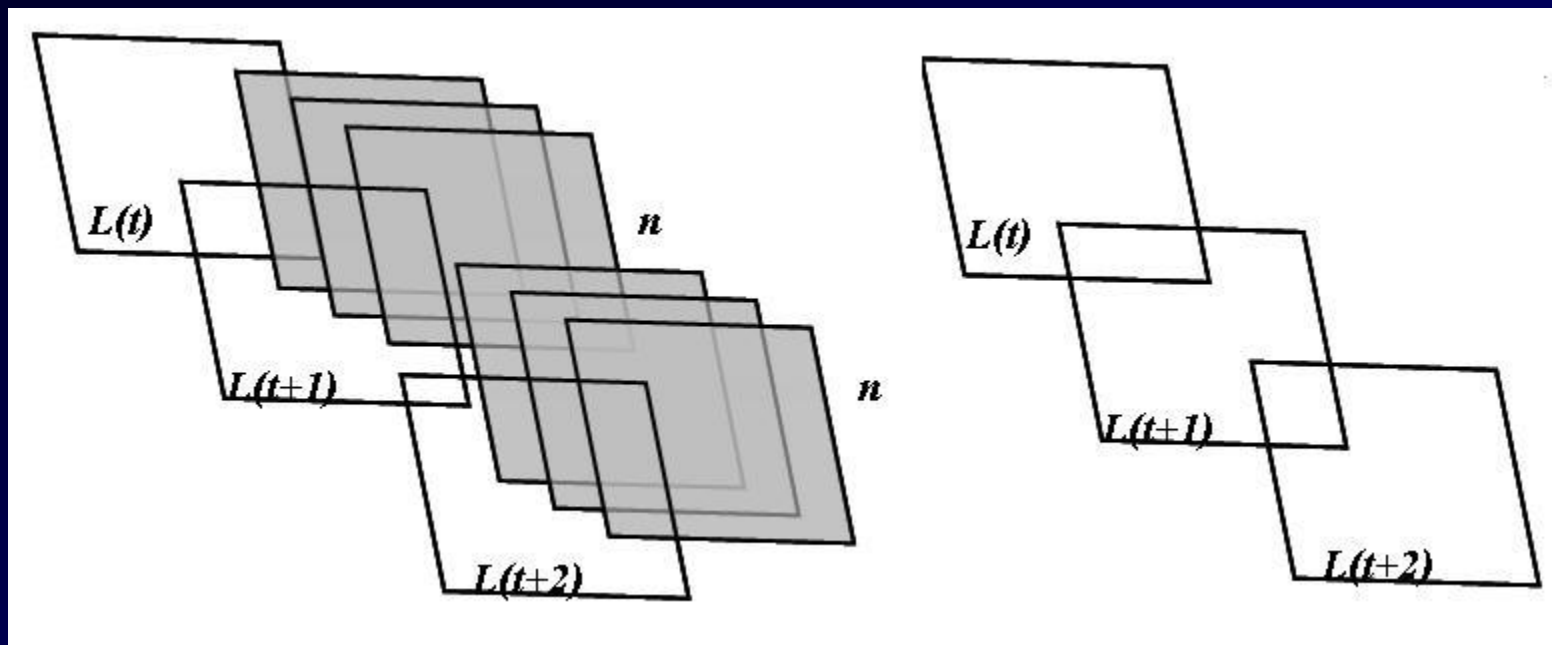
Считая, что перемещение мало, и используя ряд Тейлора, получаем:

$$I(x + \delta x, y + \delta y, t + \delta t) \approx I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t$$

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0 \quad I_x V_x + I_y V_y = -I_t$$

V - компоненты скорости оптического потока

Построение интерполированных кадров



$$L_{out}(z) = \begin{cases} L_{in}(t), & z = n \cdot t, \quad t \in \{0, 1, 2, 3, \dots\} \\ L_{in}(t), & z = n \cdot t + l, \quad l \in \{1, 2, 3, \dots, n-1\} \end{cases},$$

$$z \in \{0, 1, 2, 3, \dots\}, \quad n \in \{2, 3, 4, \dots\},$$

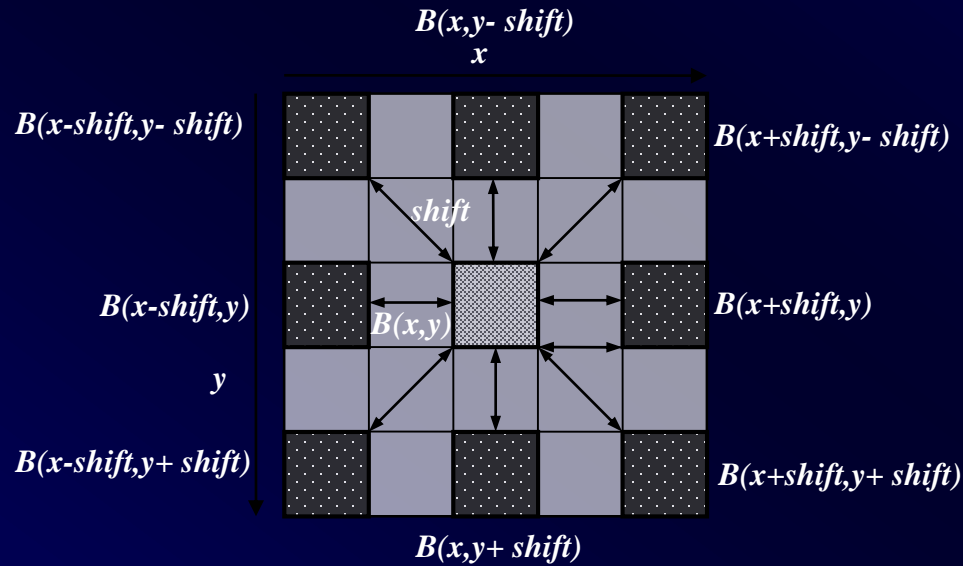
где n – параметр, определяющий, во сколько раз увеличить число кадров;
 z – общее число кадров.

Обобщенная схема построения интерполированных кадров

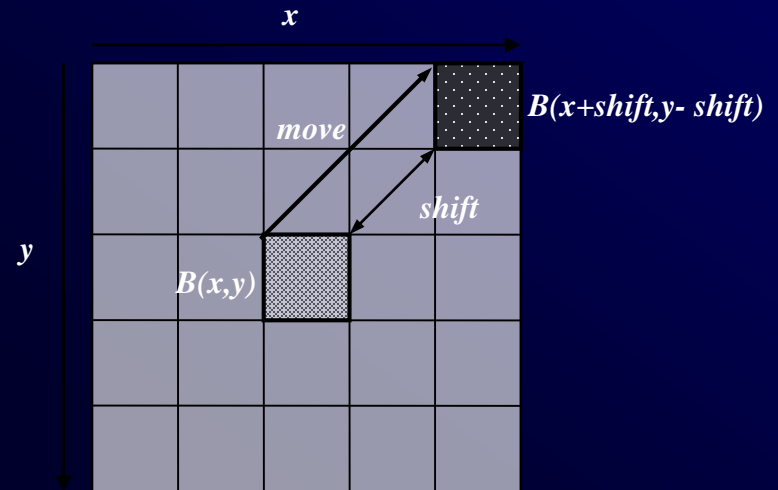


Оценки движения при построении интерполированных кадров

Область поиска движения



Вектор движения



Поле векторов движения

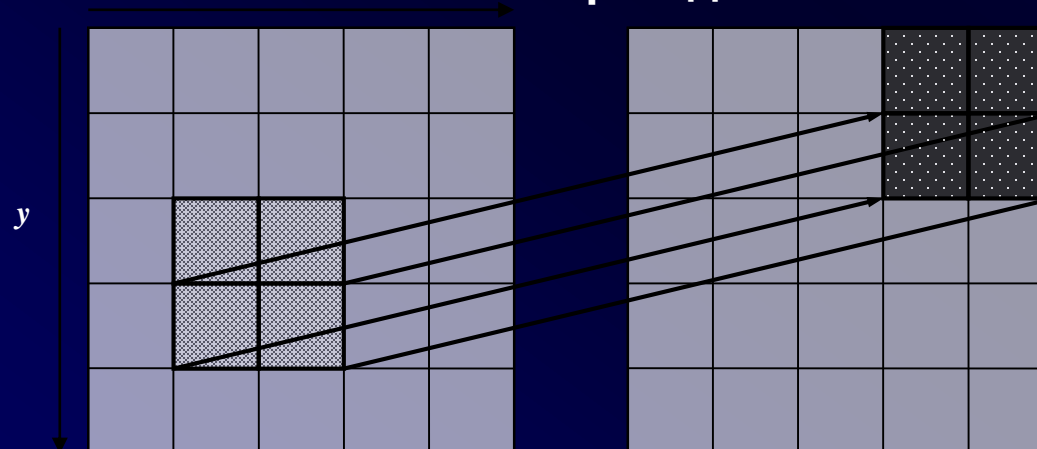
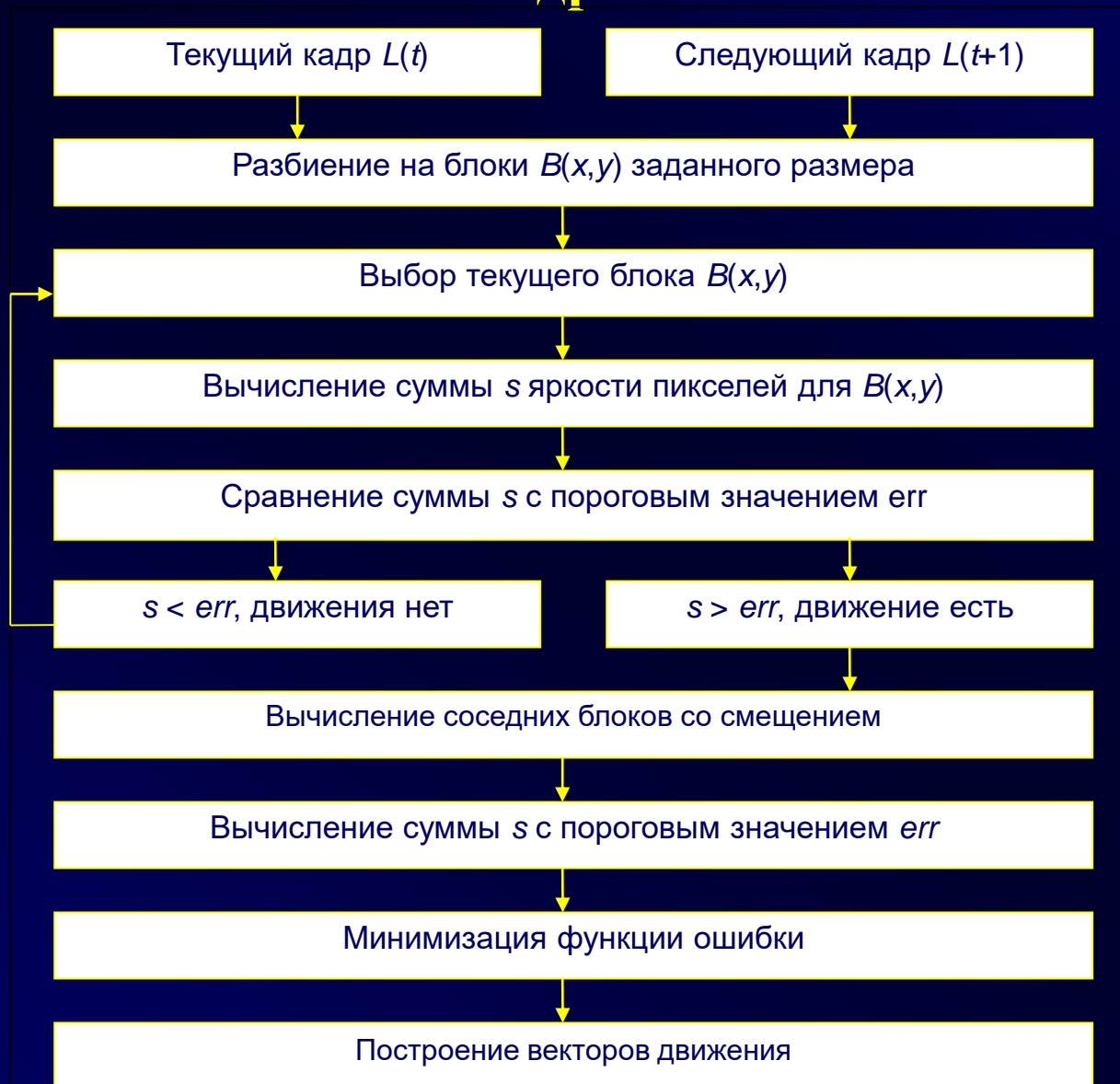


Схема оценки движения при построении интерполированных кадров



Интерполяция кадров

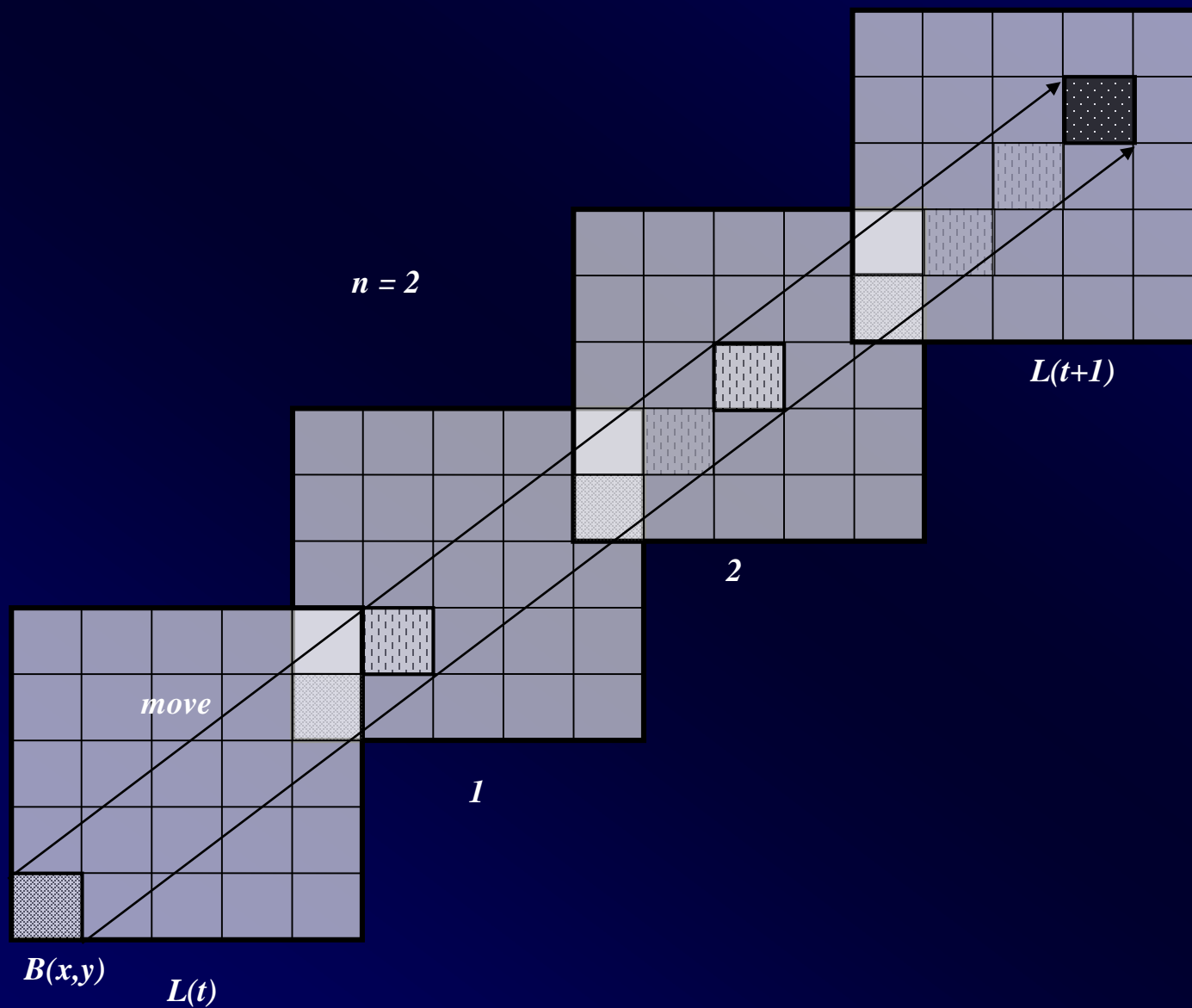
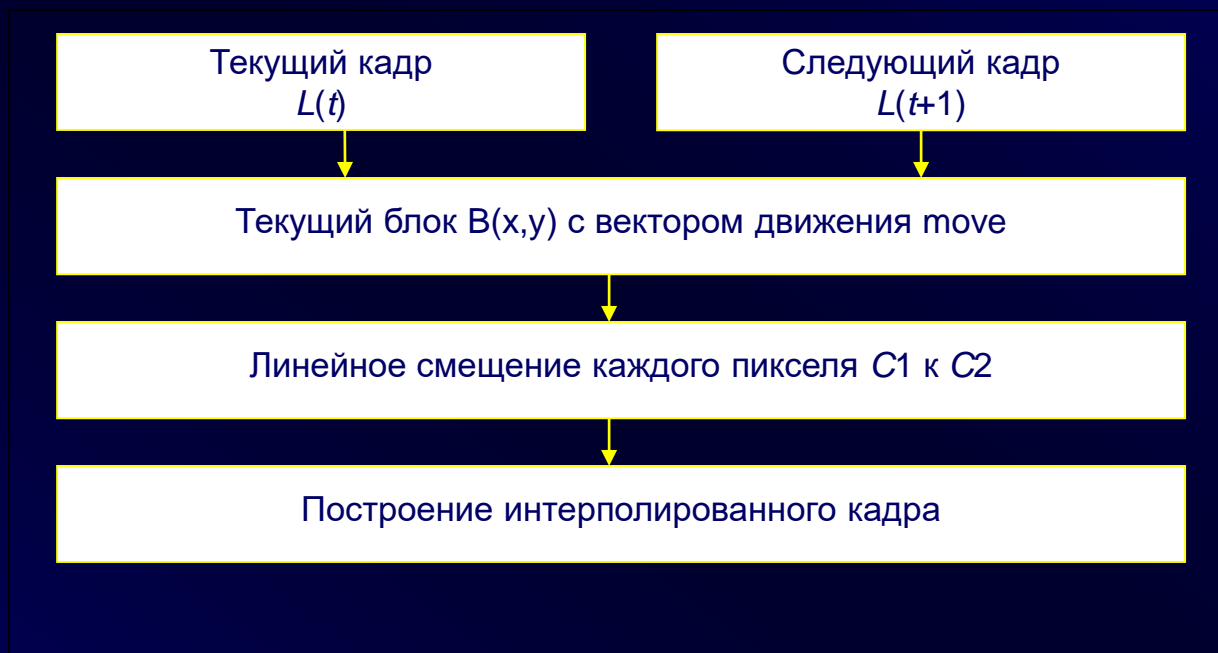


Схема построения интерполированных кадров

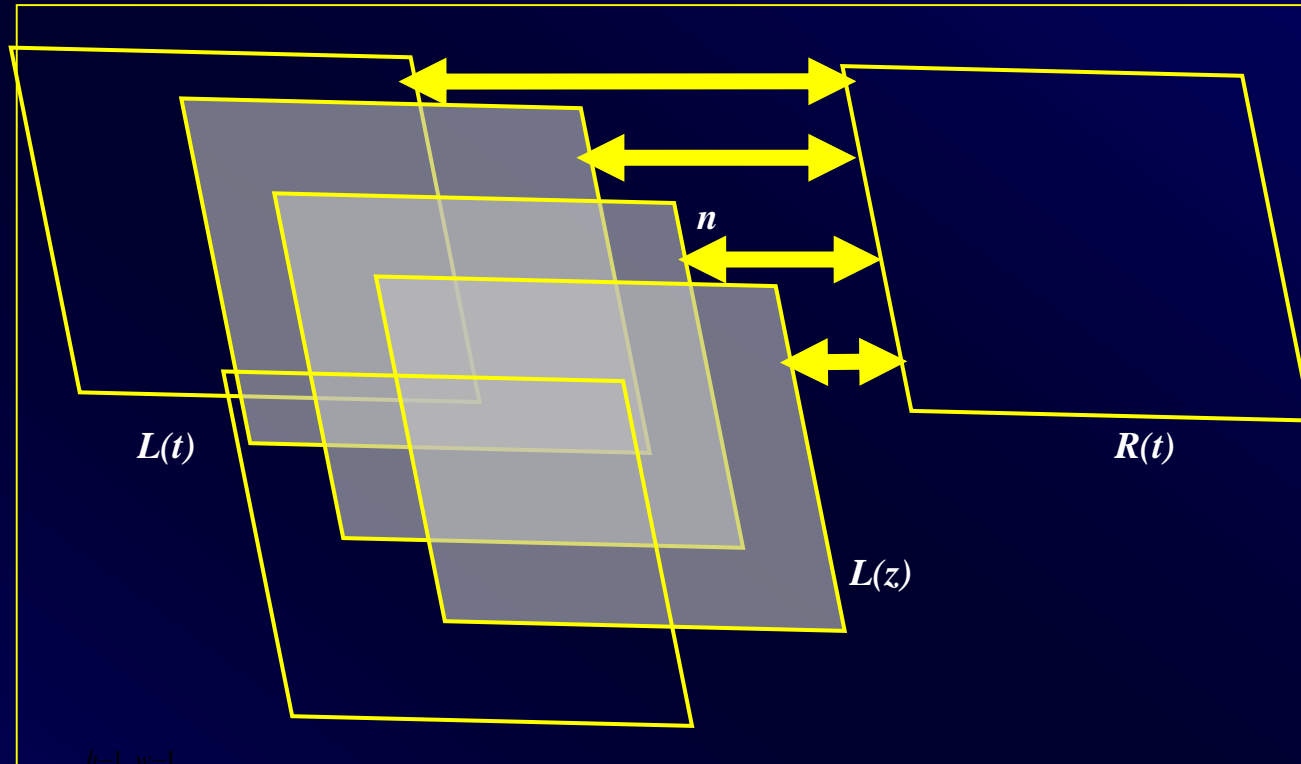


$$C_n = C_1 + (C_2 - C_1) \cdot t, \quad t \in [0;1]$$

где C_n - значение цветовой компоненты Y пикселя (x,y) блока $B(x,y)$ на интерполированном кадре n ;

t – параметр, задающий линейное смещение пикселей на каждом интерполированном кадре. $t = \frac{m}{n-1}, \quad m \in \{1,2,3,\dots,n\}$

Совмещение кадров



$$MSE = \frac{\sum_{x=0}^{h-1} \sum_{y=0}^{w-1} ((Y_2(x, y) - Y_1(x, y))^2 + (U_2(x, y) - U_1(x, y))^2 + (V_2(x, y) - V_1(x, y))^2)}{3hw}$$

$$psnr(z) = 20 \log_{10} \frac{MAX}{\sqrt{MSE}}$$

$$like = \max(psnr(z), psnr(L(t))), \quad z \in \{2, 3, \dots, n\}$$

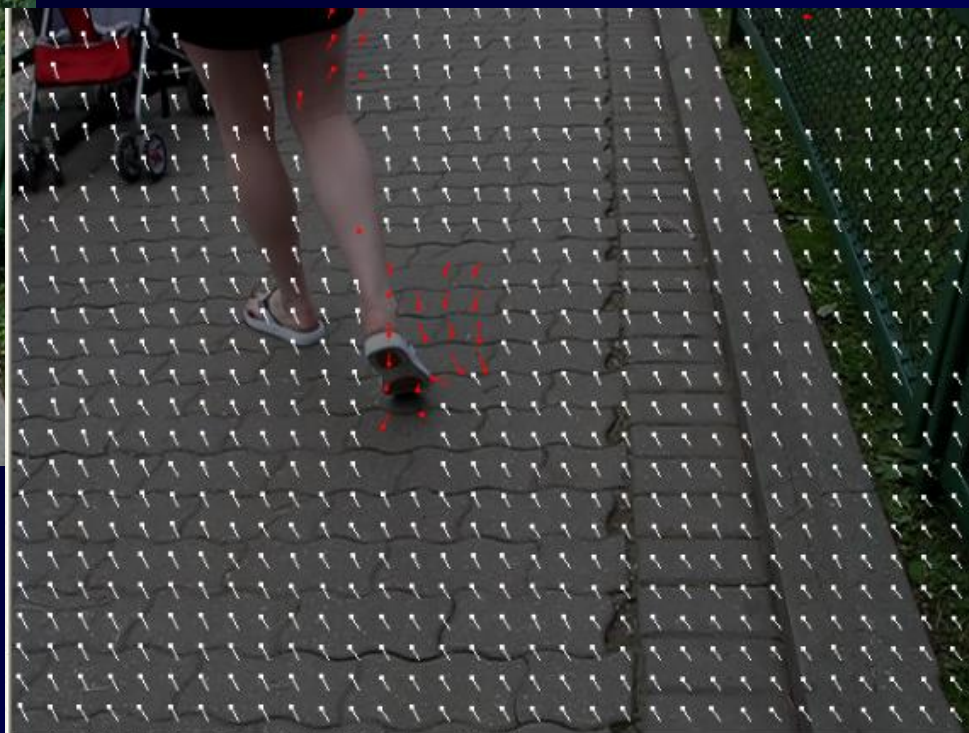
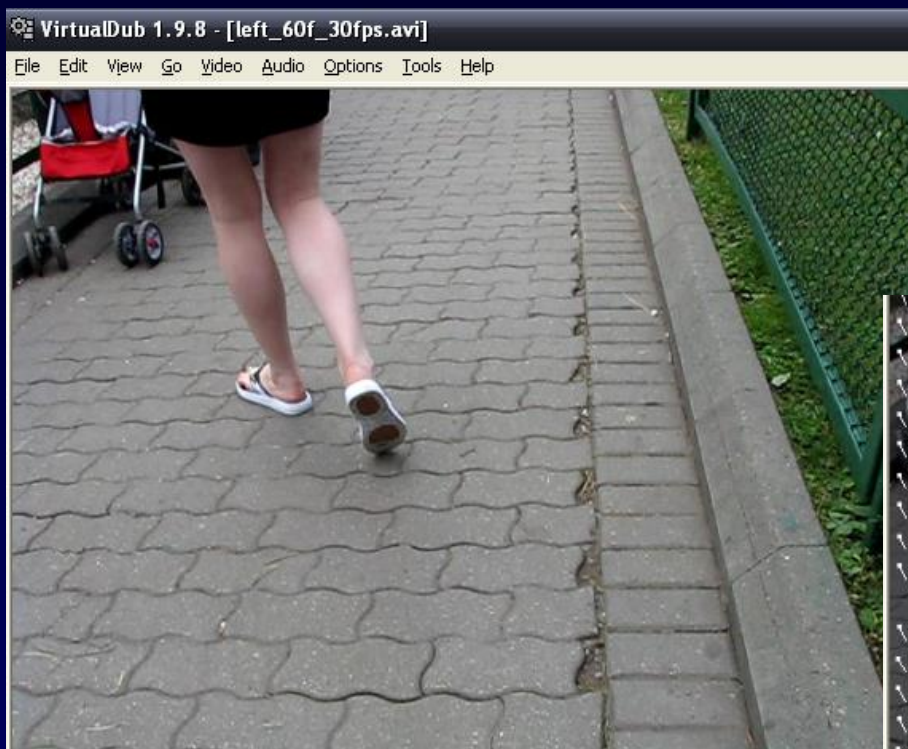
Схема совмещения кадров



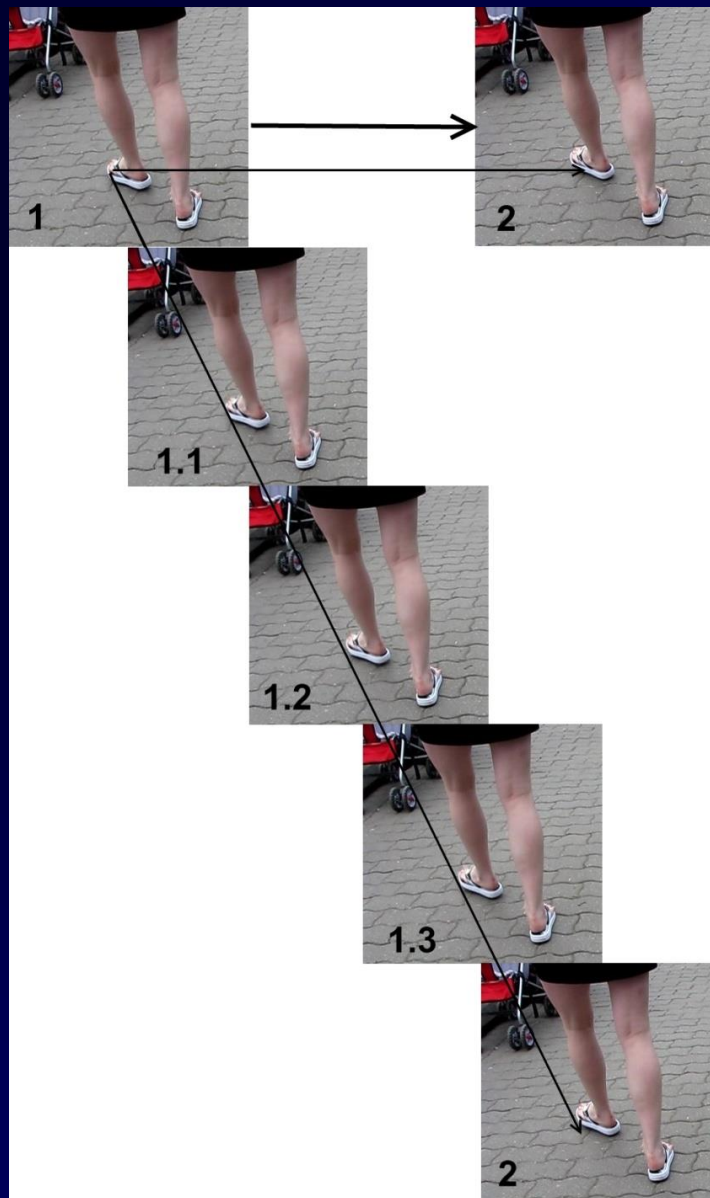
Реализация программного продукта



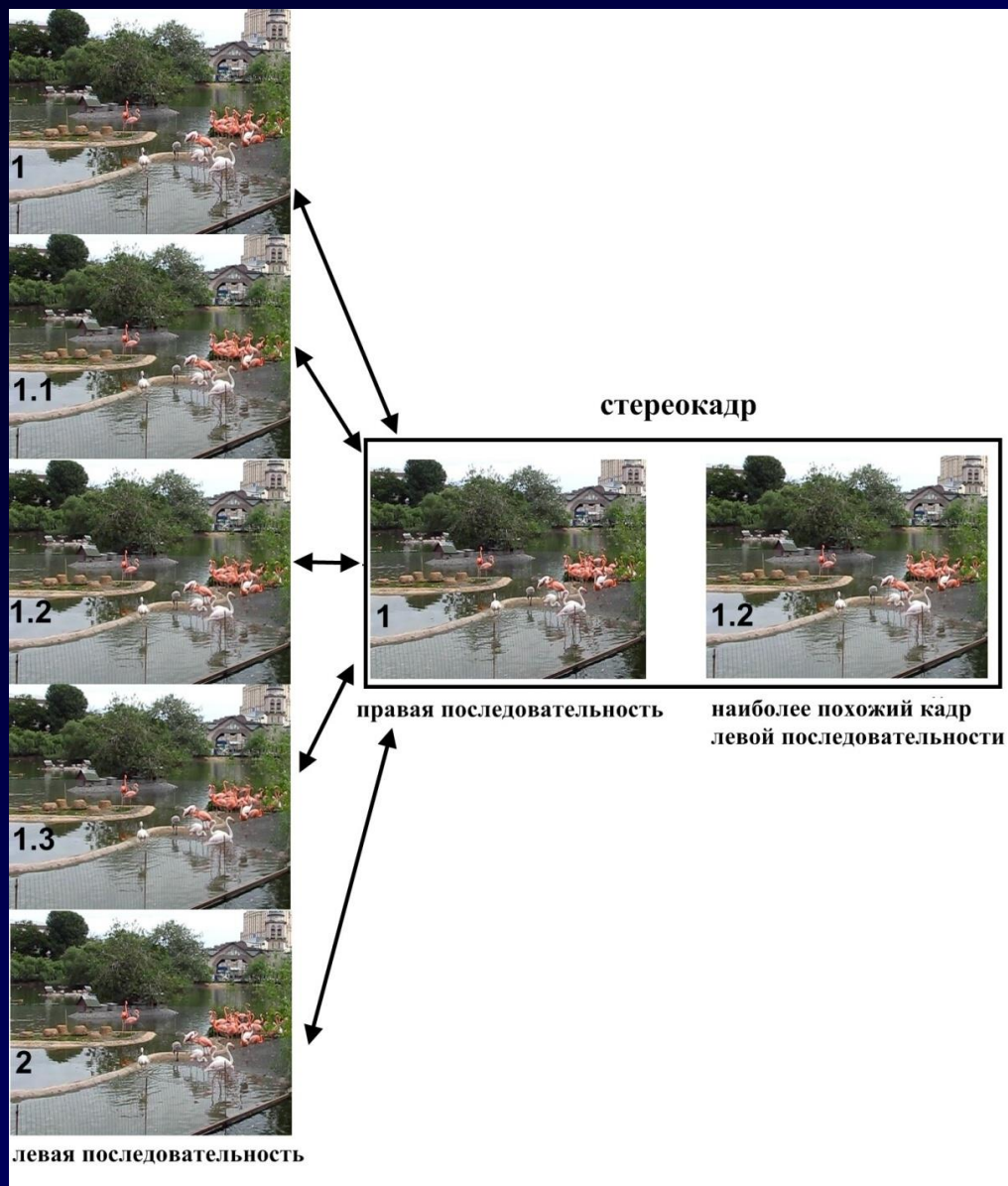
Реализация метода. Построение векторов движения



Реализация метода. Преобразование частоты кадров



Реализация метода. Совмещение кадров



Классический KLT алгоритм

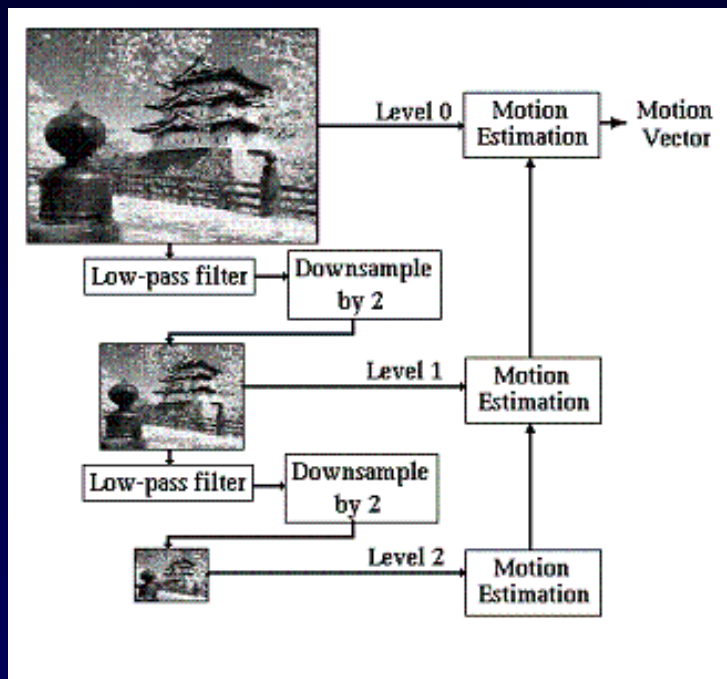
- Цель – для точки u поиск вектора d , минимизирующего следующую сумму

$$\sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} (I(x, y) - J(x + d_x, y + d_y))^2$$

где $I(x, y), J(x, y)$ – яркости первого и второго изображений в точке (x, y) .

- Происходит итеративное уточнение текущего вектора потока (метод Ньютона). В качестве начального приближения берется нулевой вектор.

Пирамидальный KLT алгоритм



$$g_L(p, q) = \left[\frac{1}{4} \sum_{u=0}^1 \sum_{v=0}^1 g_{L-1}(2p+u, 2q+v) \right]$$

1. Строится пирамида изображений =>
2. Производится поиск сдвига на самом верхнем уровне=>
3. Найденный вектор посылается на уровень выше =>
4. Производится уточнение(используя классический KLT) =>
5. Продолжается,пока не дойдем до самого нижнего уровня =>
6. Получаем вектор смещения d

Фильтрация в KLT

- Если на данном уровне пирамиды определитель обратной матрицы пространственного градиента меньше заранее определенного порога, вектор потока считается неизвестным.
- Пороги для каждого уровня определяются опытным путем.
- Для 12ти битных изображений сравнительно неплохие результаты получаются при величине порога $10000 * 2^{24}$

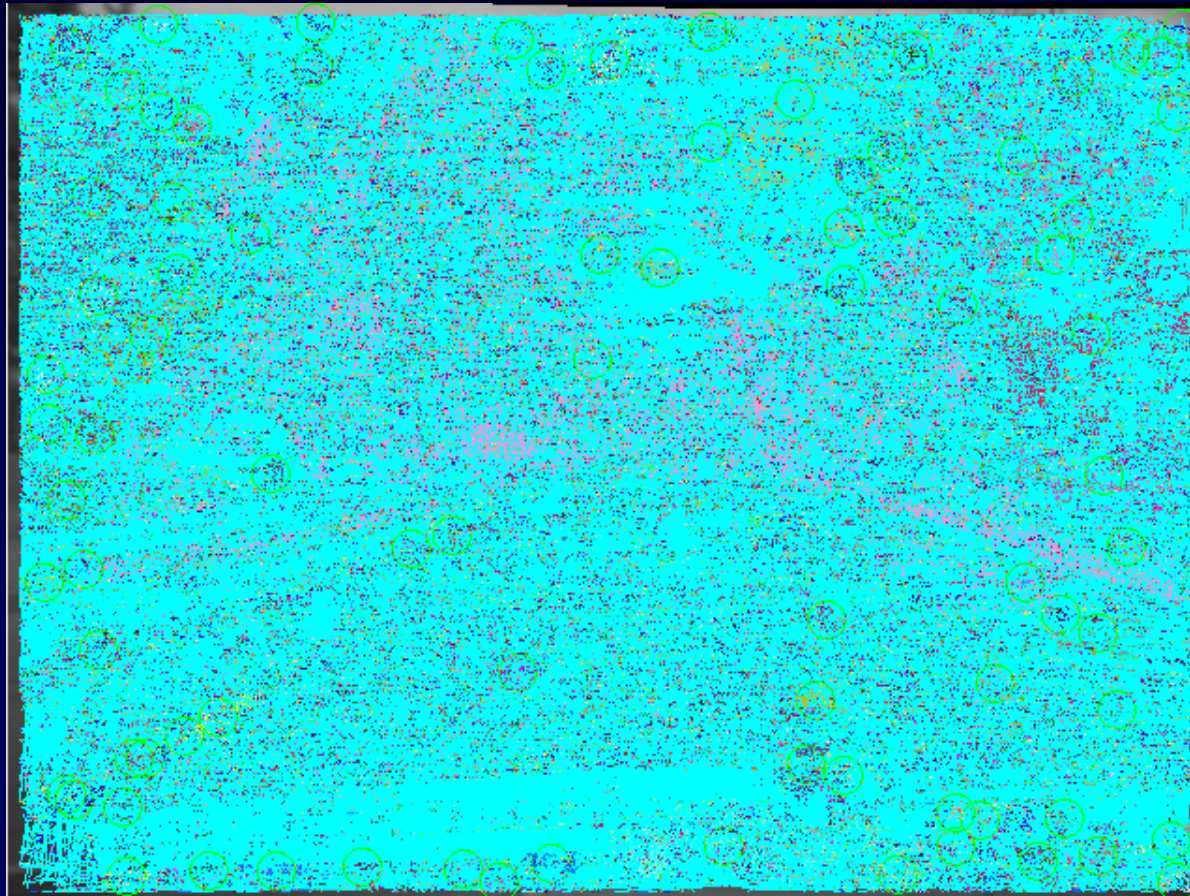
ViFlow

- Для каждой точки обоих изображений вычисляется hash – функция. Для каждого изображения получается массив значений этой функции.
- Каждый из полученных массивов сортируется.
- Считается, что точка u первого изображения переходит в точку v второго изображения, если соответствующие им значения хэша равны.

Уточнение ViFlow

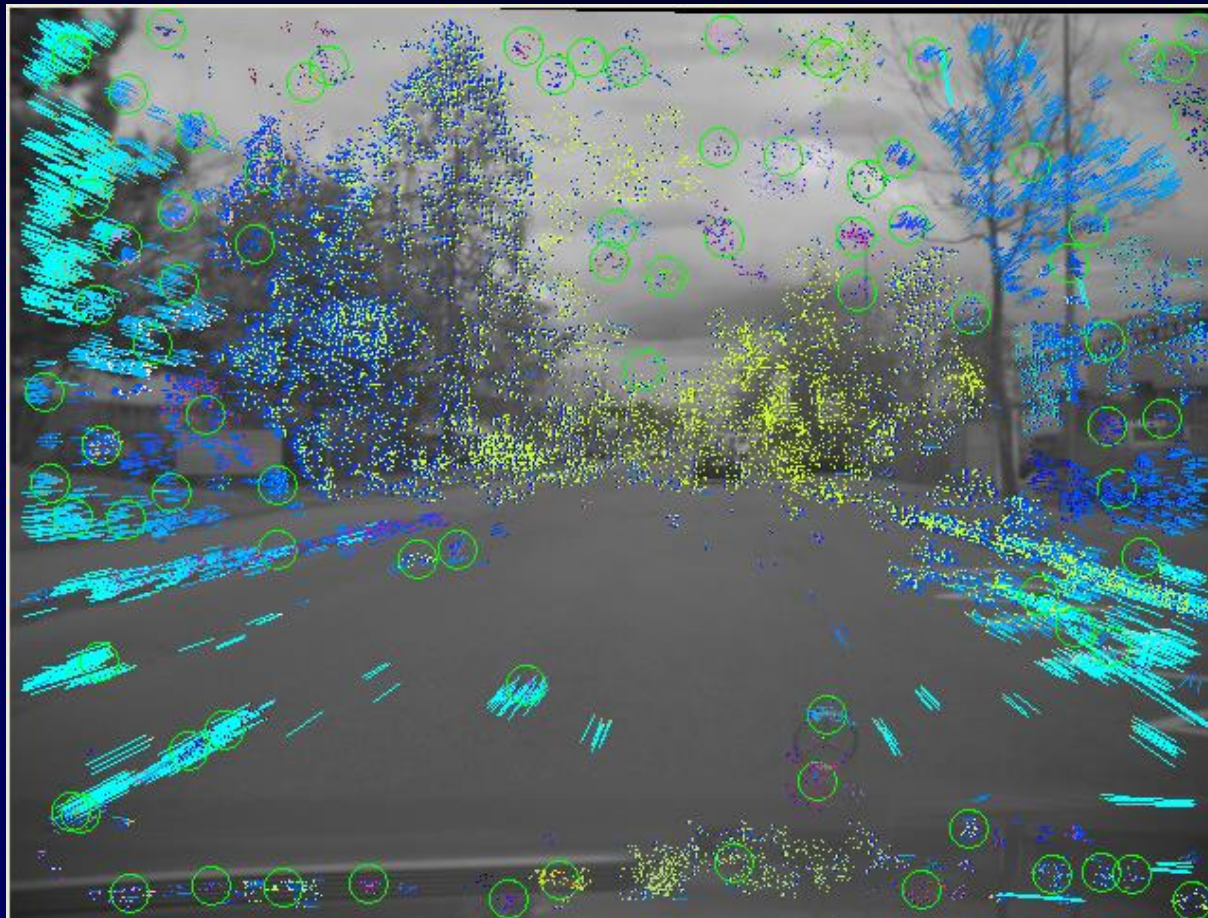
- К двум изображениям применяется ViFlow.
- Если для точки определен вектор потока, он принимается за начальное приближение в методе Ньютона, применяющимся в классическом KLT.

ViFlow



- Density = 21.86 %
- Time = 280

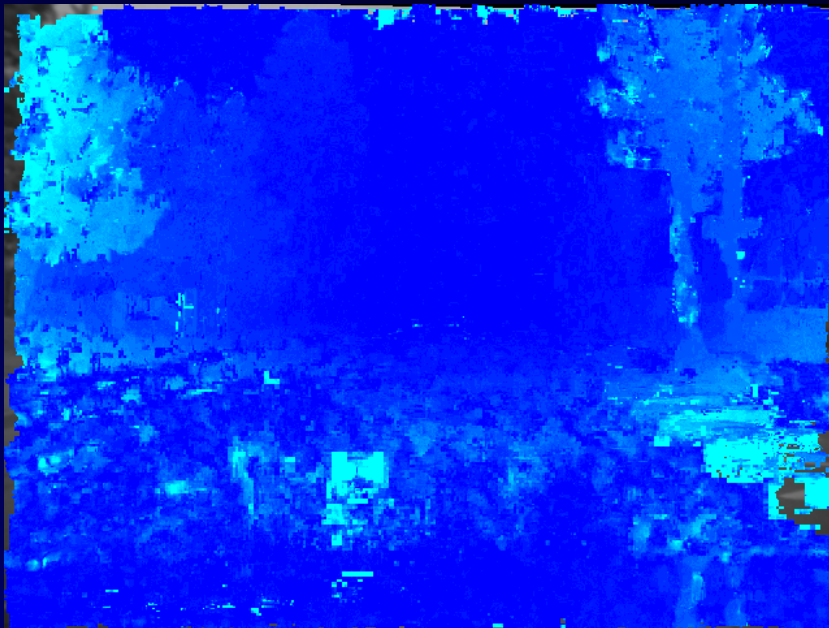
ViFlow



- Density = 5.07 %
- Time = 470

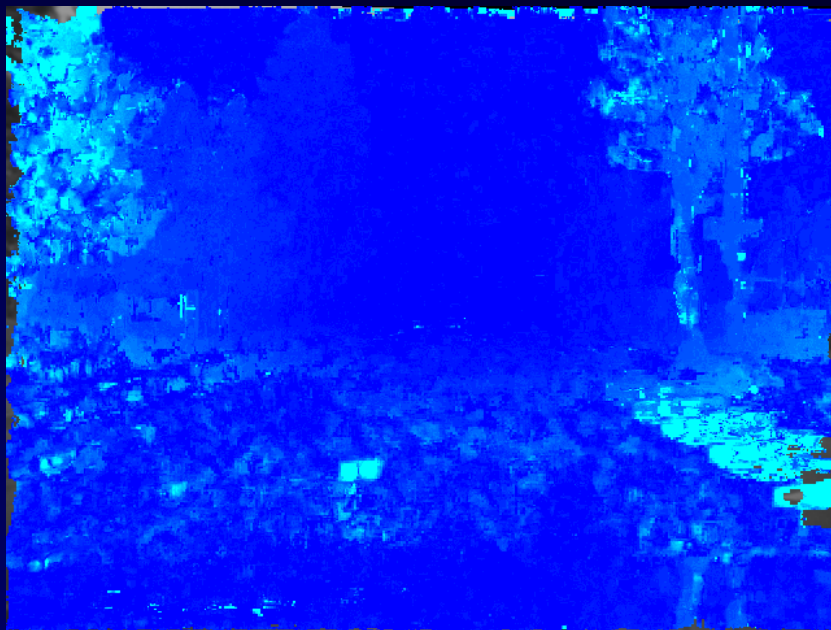
KLT

- Высота 3
- 4 итерации метода Ньютона



KLT

- Высота 2
- 4 итерации метода Ньютона



KLT vs. ViFlow

	ViFlow	ViFlow + filter	ViFlow + фильтр + уточнение	KLT (3,4)	KLT(2,4)	KLT(3,4) + фильтр
Время, clocks	280	470	760	11540	10350	2374
Плотность, %	21.86	5.07	4.68	97.17	98.33	7.9

KLT vs. ViFlow

Результаты сравнения:

- Скорость работы уточненного фильтрованного ViFlow выше скорости работы фильтрованного пирамидального KLT примерно в 4-7 раз.
- Плотность потока, полученного фильтрованным пирамидальным KLT, более чем в 2-3 раза превосходит плотность потока, полученного уточненным фильтрованным ViFlow.