



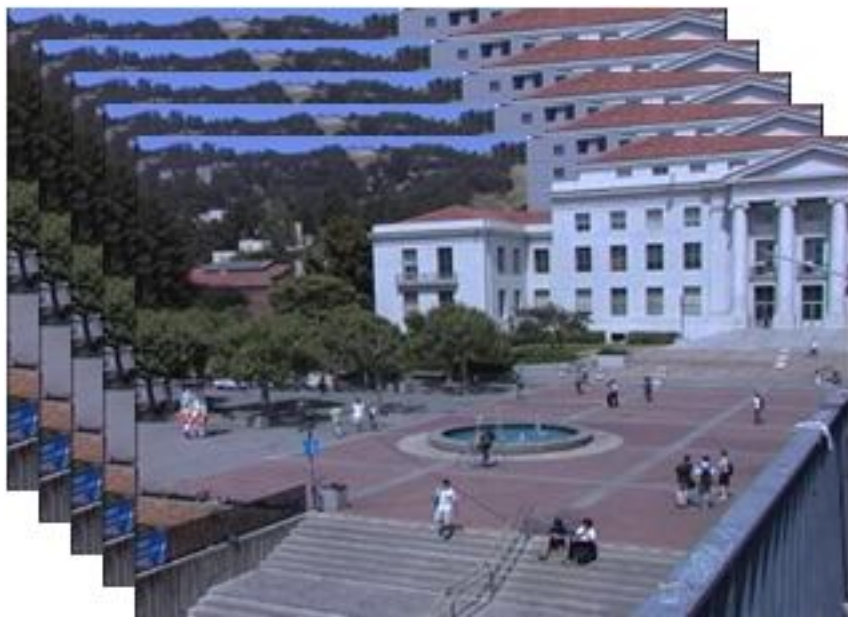
Видеокамеры



- Наиболее близкий аналог «человеческого глаза» из доступных компьютеру
- Широкое распространение:
 - Видеокамеры в смартфонах, планшетах, ноутбуках
 - ~200 000 камер видеонаблюдения в одном Лондоне!



Видеопоток



Видеопоток – упорядоченная последовательность изображений, полученных с одной камеры через небольшие промежутки времени

- Пользовательское видео – от 3-5 кадров/сек до 30-50 кадров/сек
- Разрешение – от 320x240 до 1920*1080 (HD)
- В градациях серого (одноканальное) или цветное (3х канальное)
- Поток данных – 2Мб (один канал HD) x 3 (RGB) x 50 кадр/с = 300 Мб/с
 - Больше, чем пропускная способность 1 гигабитной Ethernet сетки!



Состояние и перспективы

- Сейчас большинство видеокамер работают в режиме «регистрации»
 - Видеорегистраторы в автомобилях
 - Видеокамеры для любительской съёмки
 - Видеозапись с камер наблюдения
- Количество информации огромное; управлять ей без анализа очень сложно; хранится, в основном, на локальных носителях
- «Видеоаналитика» - все алгоритмы для извлечения информации из видео
 - Есть ряд практически работающих систем



Чего мы хотим достичь?

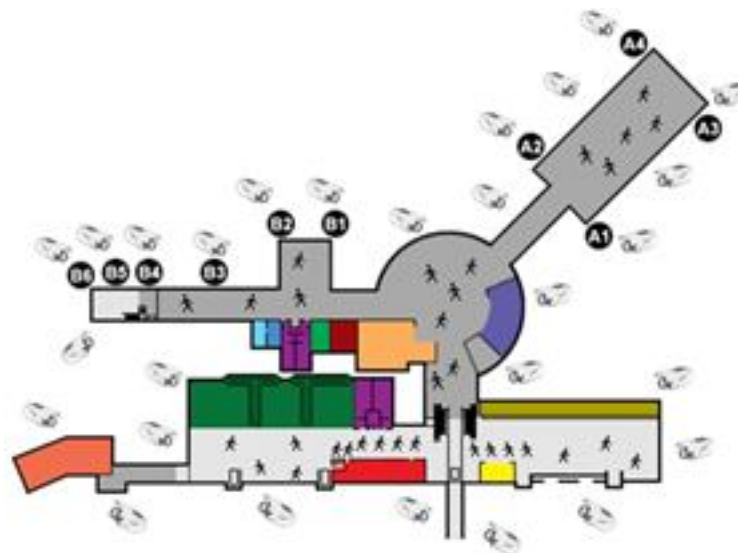


Извлечение данных из видеопотока:

- Разметить все объекты и людей
- Определить позу людей, распознать жесты
- Распознать происходящие события



Чего мы хотим достичь?



- «Situation awareness»
- Свойство систем и класс приложений, которые позволяют извлечь из сенсорных данных знание, ведущие к действиям



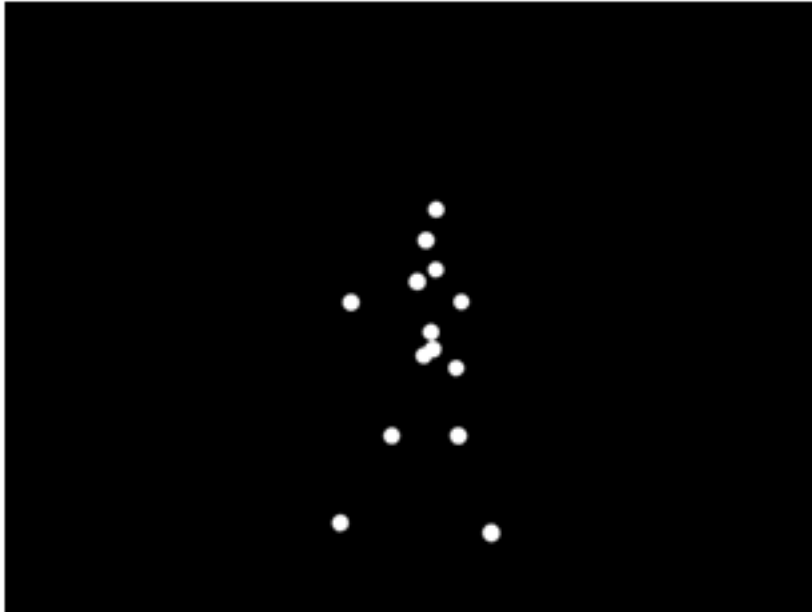


Видеопоток и изображения

Чем принципиально отличается видеопоток от обычных изображений?



Движение



Что показано в видео?

- Движение — главное отличие видео от изображений
- Движение само по себе является мощной визуальной подсказкой
- Суть многих действий именно в динамике
- Иногда достаточно отследить движение отдельных точек, чтобы распознать событие



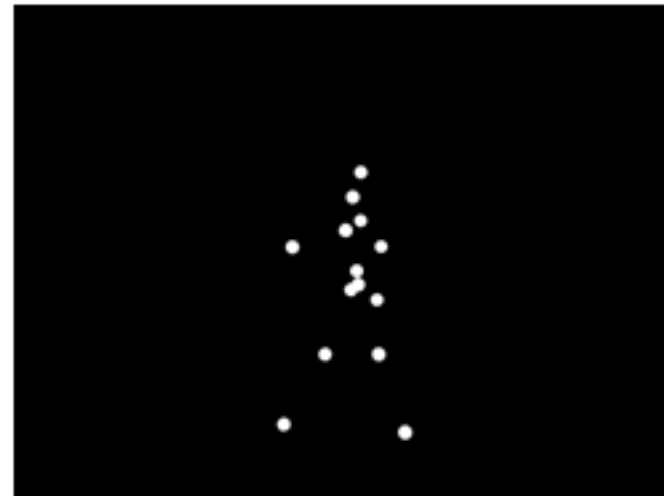
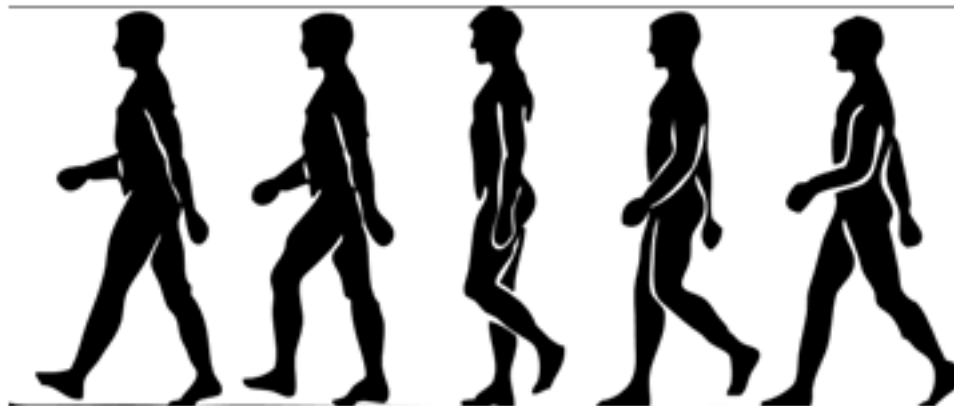
Распознавание по движению



<http://www.biomotionlab.ca/Demos/BMLwalker.html>



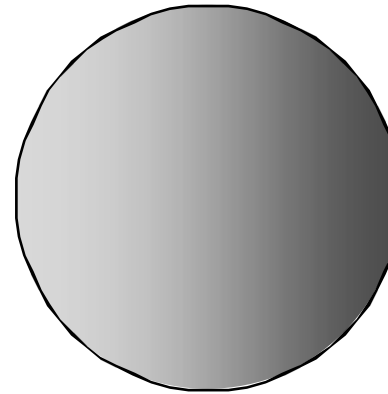
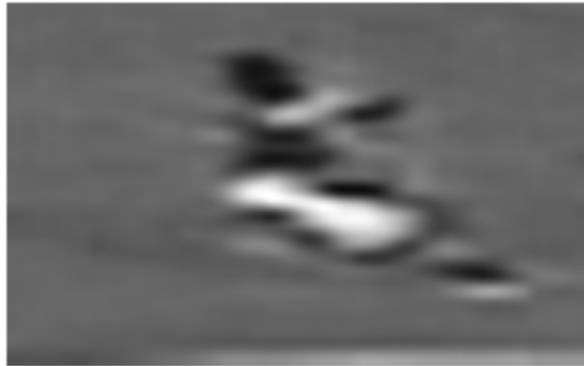
Описание движения



- Точки наблюдаемой сцены движутся относительно камеры / изображения
- Векторное поле движения 2D проекций на изображение 3D точек объектов сцены называется *полем движения* (*motion field*)
- Нужно это движение как-то формализовать, описывать и измерять



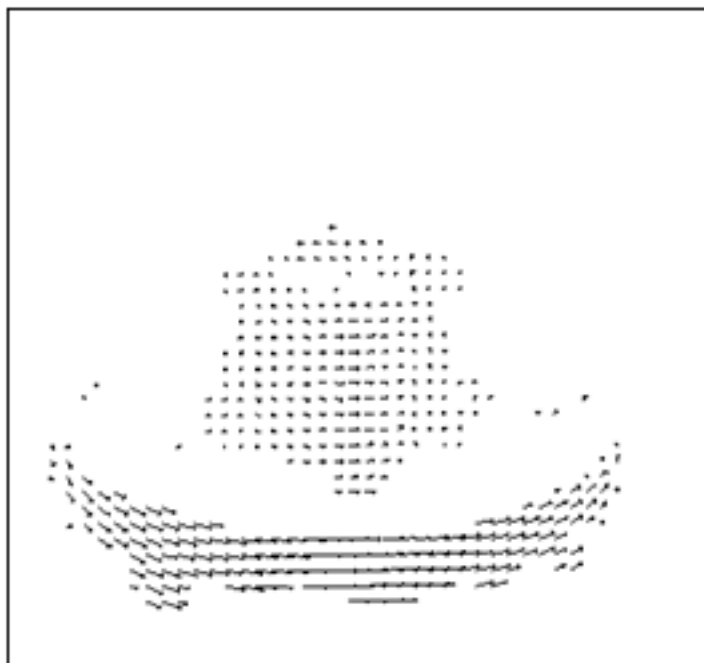
Оптический поток



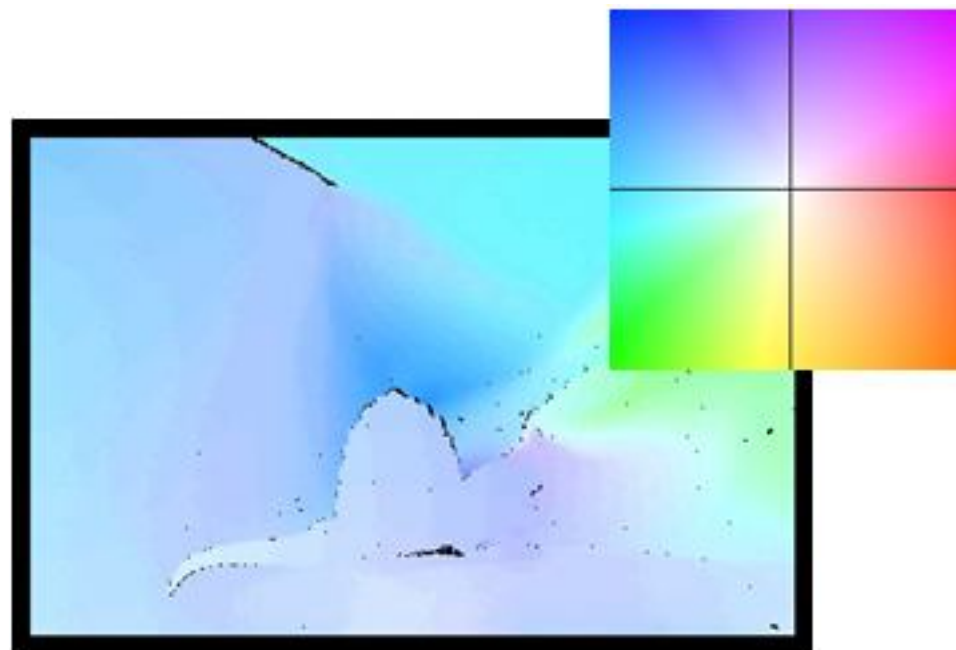
- Движение точек объектов по видео увидеть можно далеко не всегда
- Пример:
 - Серый матовый шар, освещается с одной стороны и вращается вокруг своей оси
 - Яркость всех пикселей изображения в этом случае будет постоянной
- **Оптический поток (optical flow)** – векторное поле *видимого (apparent)* движения пикселей между кадрами
 - Вычисление оптического потока – одна из базовых задач анализа видео
 - Задача похожа на задачу попиксельного сопоставления двух изображений одной и той же сцены



Визуализация



Вектора движения для
отдельных точек или всего
изображения



Цветовое кодирование вектора
движения. Каждому направлению и
амплитуде свой цвет и яркость

Оптический поток - векторное поле (u_{ij}, v_{ij}) видимого (наблюдаемого) движения пикселей между кадрами



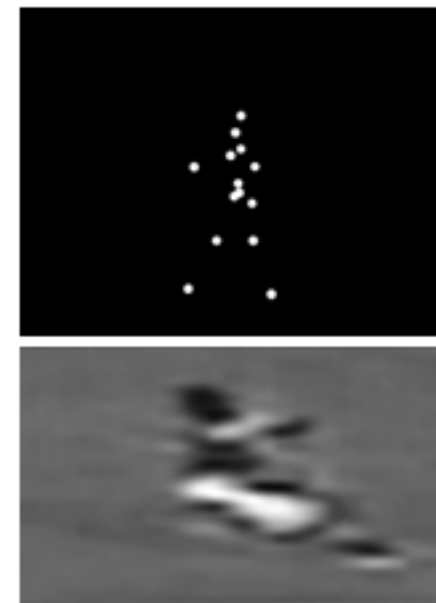
Ключевое предположение №1

- Как оценить движение пикселей от I_1 в изображение I_2 ?
- Нам нужно всем пикселям из I_1 найти соответствующие пиксели в I_2
- Предположение – пиксель при перемещении не меняет цвет!
- Запишем это условие для всего изображения:

$$E(\mathbf{u}, \mathbf{v}) = \sum_{i,j} \{\rho_D(I_1(i, j) - I_2(i + u_{i,j}, j + v_{i,j}))\}$$

$E(u, v)$ – целевая функция, ρ_D – функция вычисления ошибки

- Соответствует ли минимум функции нашей цели?





Минимизация функции

$$E(\mathbf{u}, \mathbf{v}) = \sum_{i,j} \{\rho_D(I_1(i, j) - I_2(i + u_{i,j}, j + v_{i,j}))\}$$

- Глобальный минимум может не соответствовать оптическому потоку, т.к. мы можем найти похожие по яркости пиксели где угодно по изображению.
- Нужны ещё предположения и ограничения!



Посмотрим пример



- Что можно сказать про смещение пикселей?
 - Пиксели смещаются на небольшое расстояние (предположение N2)
 - Можно наложить порог на длину вектора (u_{ij}, v_{ij})



Посмотрим пример



- Что можно ещё сказать про движение пикселей?
 - Соседние пиксели чаще всего двигаются похоже!
 - Т.е. вектора оптического потока близки у близких пикселей!



Ключевое предположение №3

Добавим в нашу целевую функцию штраф за изменение векторов оптического потока:

$$E(\mathbf{u}, \mathbf{v}) = \sum_{i,j} \{ \rho_D(I_1(i, j) - I_2(i + u_{i,j}, j + v_{i,j})) \quad (1) \\ + \lambda [\rho_S(u_{i,j} - u_{i+1,j}) + \rho_S(u_{i,j} - u_{i,j+1}) \\ + \rho_S(v_{i,j} - v_{i+1,j}) + \rho_S(v_{i,j} - v_{i,j+1})] \},$$

где $\rho_S(x)$ – штрафная функция, например, квадратичная

$$\rho(x) = x^2$$

Получили глобальную целевую функцию для вычисления оптического потока между двумя кадрами



Глобальный подход

$$E(u, v) = \sum_{i,j} \{ \rho_D(I_1(i, j) - I_2(i + u_{i,j}, j + v_{i,j})) \quad (1) \\ + \lambda [\rho_S(u_{i,j} - u_{i+1,j}) + \rho_S(u_{i,j} - u_{i,j+1}) \\ + \rho_S(v_{i,j} - v_{i+1,j}) + \rho_S(v_{i,j} - v_{i,j+1})] \},$$

- Это первая формулировка задачи оптического потока для всего изображения в целом (Horn, 1981)
- Будем оптимизировать её каким-нибудь методом оптимизации
- Такой подход для оценки оптического потока называют *глобальным*



Развитие глобальных методов

- Глобальные методы обеспечивают сейчас наилучшее качество оптического потока
- Отличаются друг от друга:
 - Функциями штрафов за изменение вида пикселей (яркость, цвет, градиенты и т.д.)
 - Функциями штрафов за разницу в векторах потока у соседних пикселей
 - Методами оптимизации (непрерывные, дискретные, стохастические)
 - Дополнительными эвристиками и ограничениями, например, регулярным применением медианной фильтрации



Посмотрим пример ещё раз



- Что можно ещё сказать про движение пикселей?
 - В некоторых случаях разница между векторами движения очень большая и оправданная (разные объекты)
 - В других областях вектора движения почти одинаковые



Робастные штрафы

- Штрафы должны быть «устойчивы» к разрывам в карте оптического потока на границах объектов
- Штраф должен ещё не «портить» функцию
 - Чтобы мы могли её хорошо оптимизировать



$$\rho(x) = \sqrt{x^2 + \epsilon^2}$$

Charbonnier
(дифференцируемый L1)

$$\rho(x) = \log\left(1 + \frac{x^2}{2\sigma^2}\right)$$

Lorenzian

Sun, D., Roth, S., and Black, M. J., Secrets of optical flow estimation and their principles, CVPR, June 2010



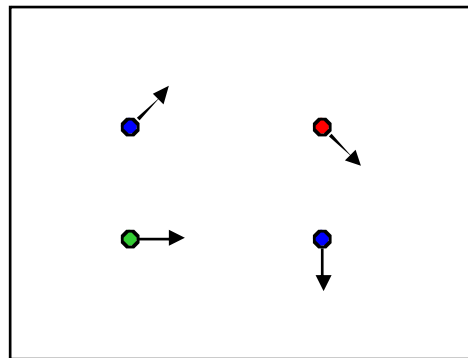
Резюме глобальных методов

$$E(\mathbf{u}, \mathbf{v}) = \sum_{i,j} \{ \rho_D(I_1(i, j) - I_2(i + u_{i,j}, j + v_{i,j})) \quad (1) \\ + \lambda [\rho_S(u_{i,j} - u_{i+1,j}) + \rho_S(u_{i,j} - u_{i,j+1}) \\ + \rho_S(v_{i,j} - v_{i+1,j}) + \rho_S(v_{i,j} - v_{i,j+1})] \},$$

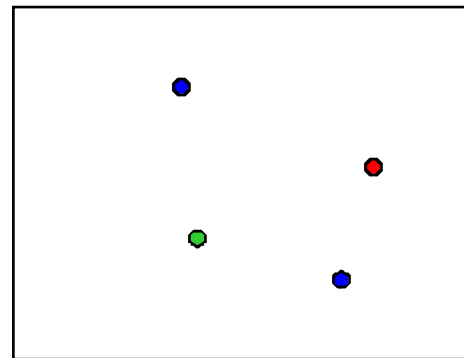
- Оптимизируем одну целевую функцию для всего векторного поля
- Обеспечивают наибольшую точность
- Многие методы крайне медленные (до нескольких часов на пару кадров!)



Локальный подход



$H(x, y)$

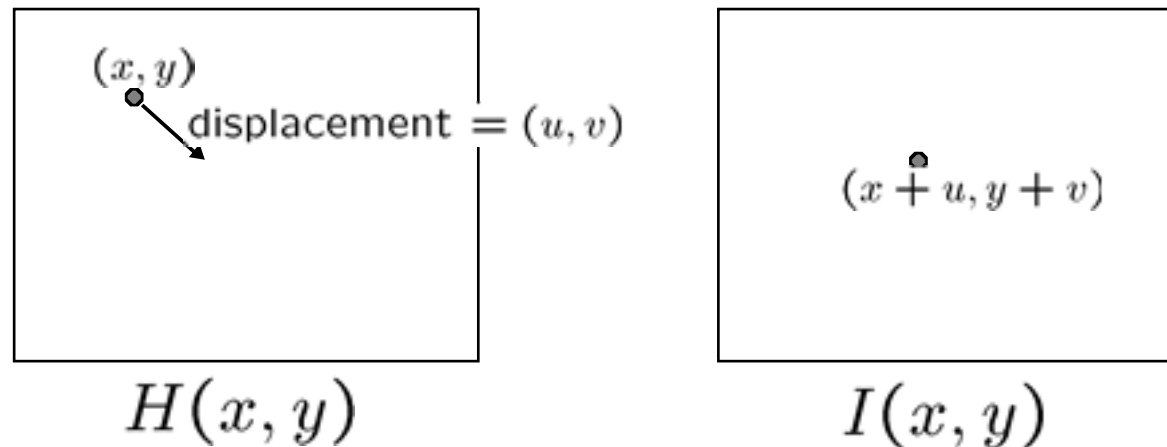


$I(x, y)$

- Попробуем вычислить вектор движения для одного пикселя (x, y) из изображения I в изображение H
- Используем ограничения для формализации задачи
 - Постоянная яркость
 - Малое смещение



Локальный подход



С учётом малого смещения, разложим $I(x+u, y+v)$ в ряд Тейлора:

$$\begin{aligned} I(x+u, y+v) &= I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms} \\ &\approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v \end{aligned}$$



Уравнение оптического потока

- Объединим с условием постоянной яркости:

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

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

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



Уравнение оптического потока

- Элементарное уравнение оптического потока:

$$0 = I_t + \nabla I \cdot [u \ v]$$

- Вопрос: сколько неизвестных и уравнений для каждого пикселя?
- 1 уравнение, 2 неизвестных (u, v)



Дополнительные уравнения

- Как можно получить больше уравнений?
- Воспользуемся предположением о близости векторов движения для соседних пикселей
- Пусть для всех пикселей p из окрестности (x, y) смещение (u, v) одинаковое!!
 - Для окна 5×5 получаем 25 уравнений для каждого пикселя!

$$0 = I_t(p_i) + \nabla I(p_i) \cdot [u \ v]$$

$$\underbrace{\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix}}_{\substack{A \\ 25 \times 2}} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\substack{d \\ 2 \times 1}} = - \underbrace{\begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix}}_{\substack{b \\ 25 \times 1}}$$



Алгоритм Лукаса-Канаде

- Проблема: больше уравнений, чем неизвестных!

$$\begin{matrix} A & d & = & b \\ 25 \times 2 & 2 \times 1 & & 25 \times 1 \end{matrix} \longrightarrow \text{minimize } \|Ad - b\|^2$$

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

$$\begin{matrix} (A^T A) & d & = & A^T b \\ 2 \times 2 & 2 \times 1 & & 2 \times 1 \end{matrix} \Rightarrow d = (A^T A)^{-1} A^T b$$

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

- Суммируем по всем пикселям в окне $K \times K$

B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, April 1981.



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 (вырожденной)
 - $A^T A$ должна быть хорошо определена



Что это напоминает?

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}$$

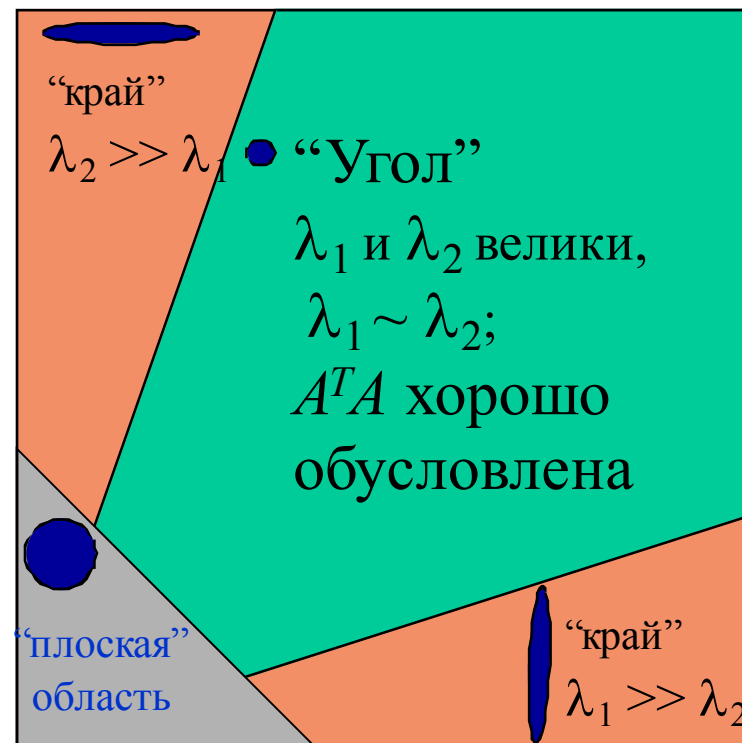
Матрица для
нахождения локальных
особенностей
детектором Харриса!

$$A^T A = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

$A^T A$ хорошо определена в
локальных особенностях
изображения!

λ_1 и λ_2 малы;
 $A^T A$ плохо
обусловлена

λ_2

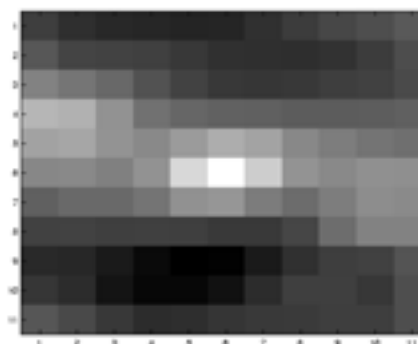
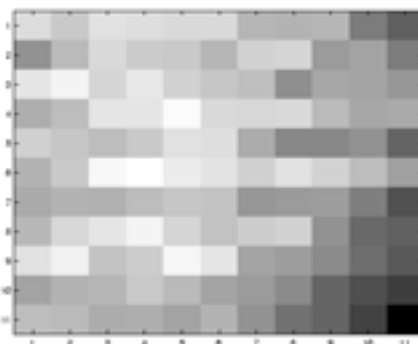
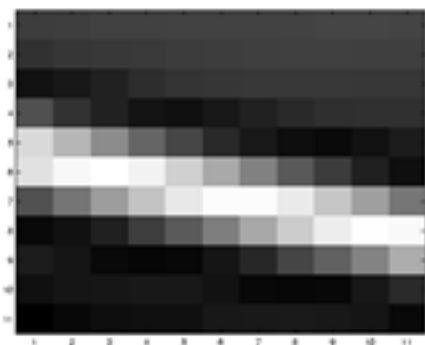


λ_1



Локальный оптический поток

- Локально оценить оптический поток может только в некоторых точках, соответствующих локальным особенностям
- Как следствие, хотя вычисляется поток по 2м изображениям, но определить, где он будет считаться хорошо мы можем по 1 изображению



Jianbo Shi and Carlo Tomasi, "Good Features to Track," *CVPR* 1994



Итеративное уточнение

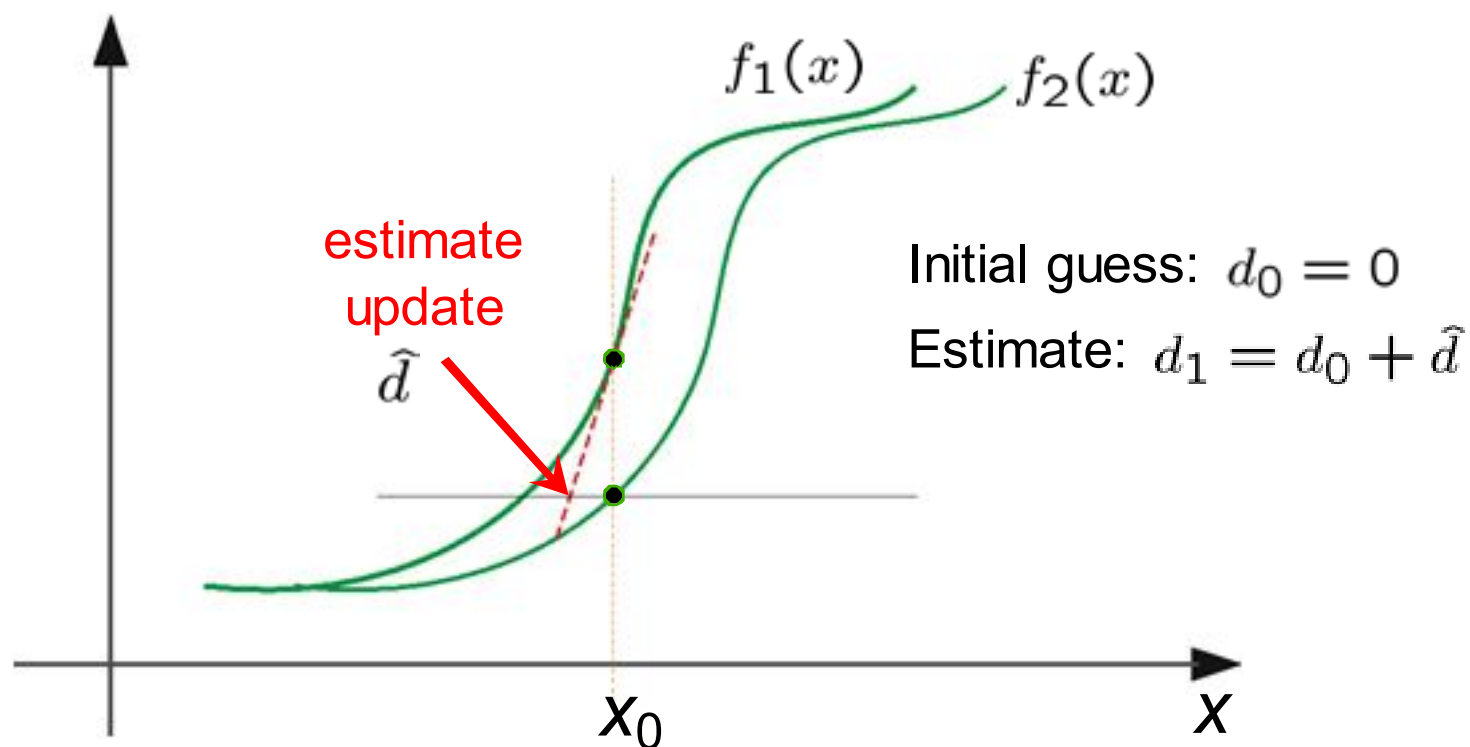
- Вспомним первый шаг

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$
$$\approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

- Поскольку мы с самого начала использовали приближение, то получаем только приближенное решение
- Можем уточнить за счёт итеративной процедуры:
 1. Оценить движение в каждом пикселе, решив уравнения Лукаса-Канаде
 2. Преобразовать изображение I используя вычисленное движение
 3. Повторить 1-2 до сходимости



Пример

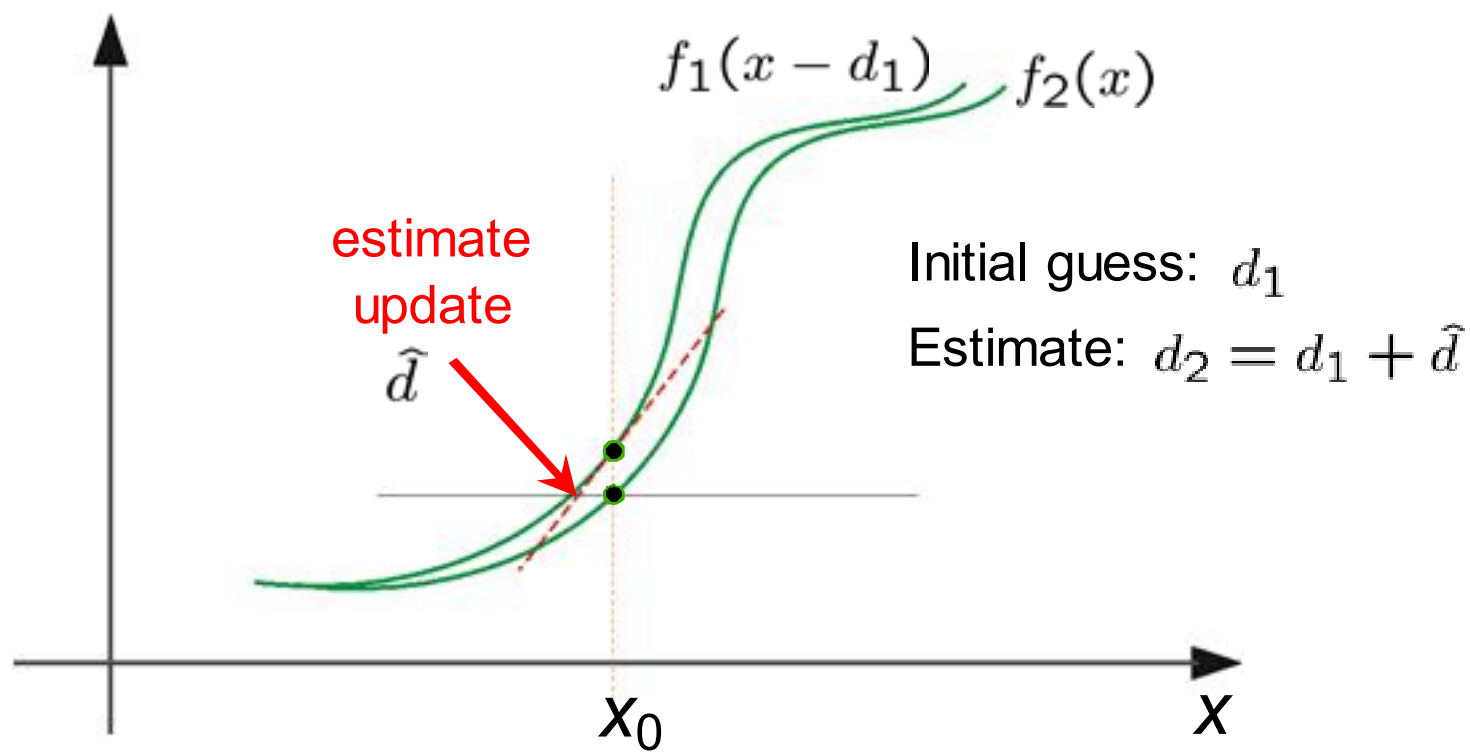


- Рассмотрим одномерный случай (d-смещение)
- Сдвиг по градиенту должен быть равен изменению значения (яркости).

$$0 = I_t + I_x d$$

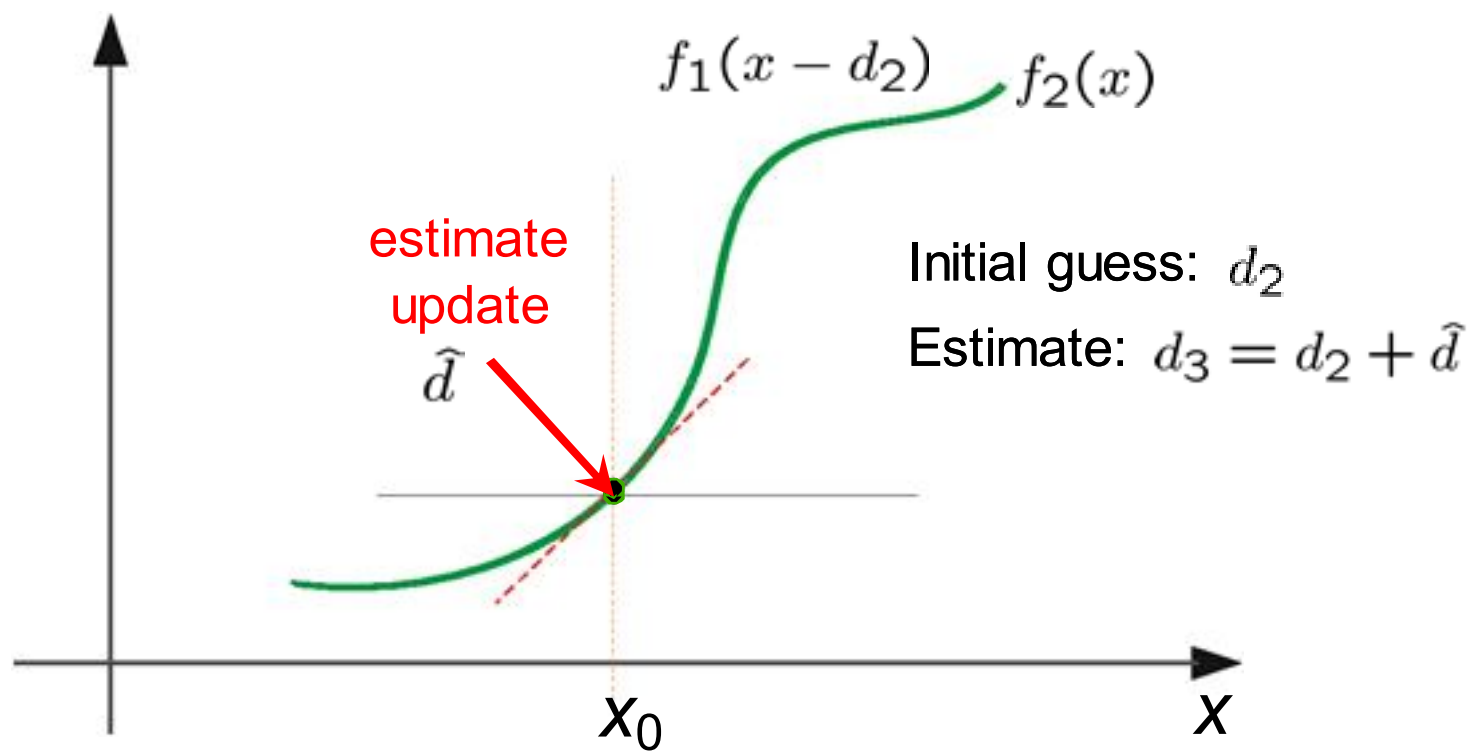


Пример

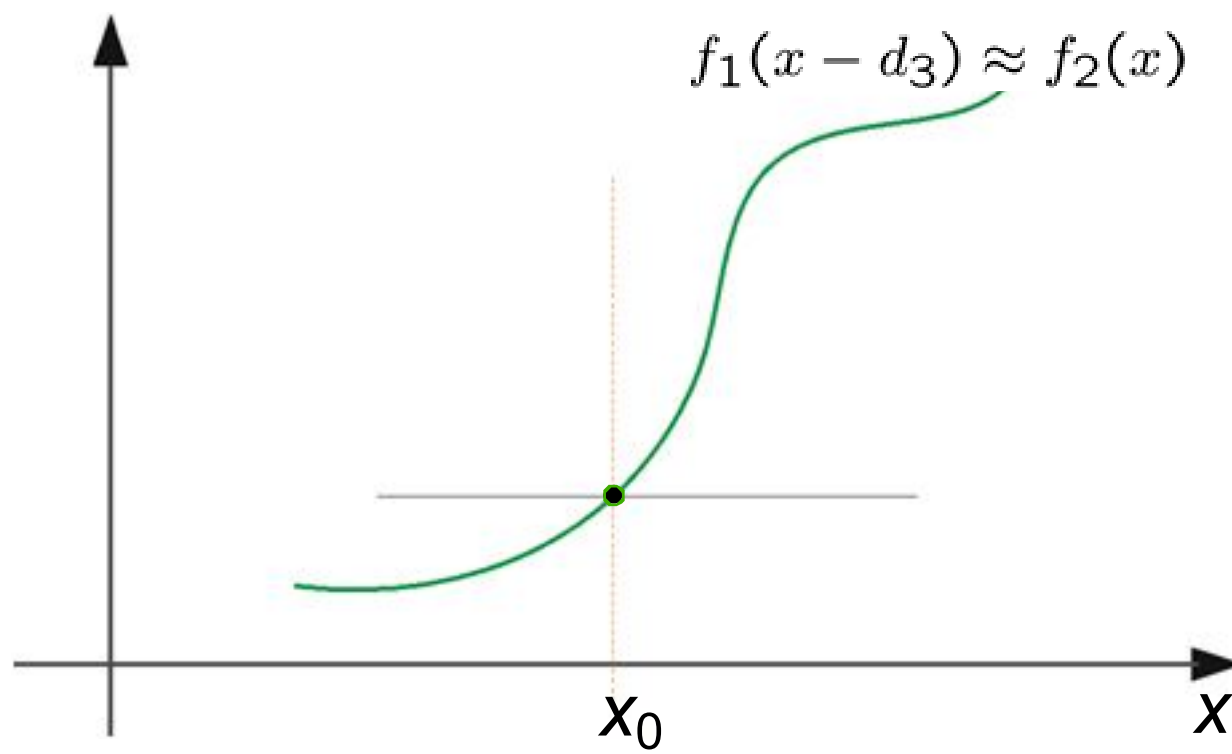




Пример



Пример





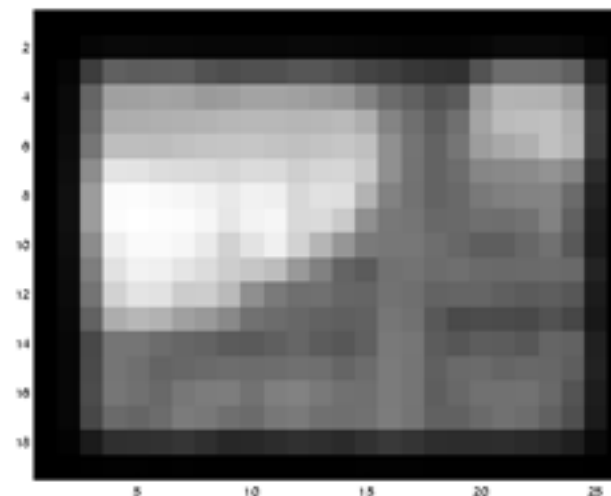
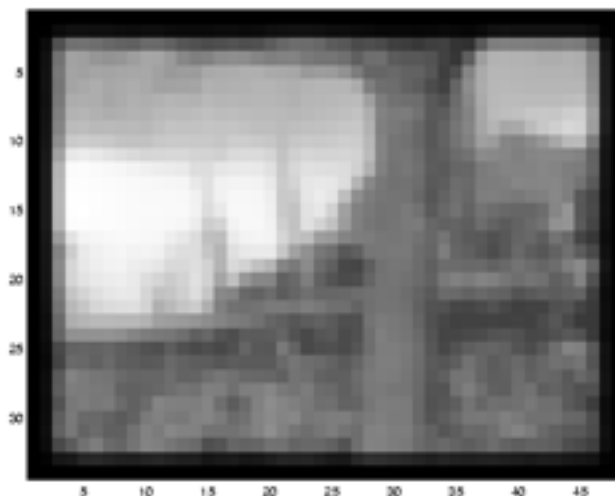
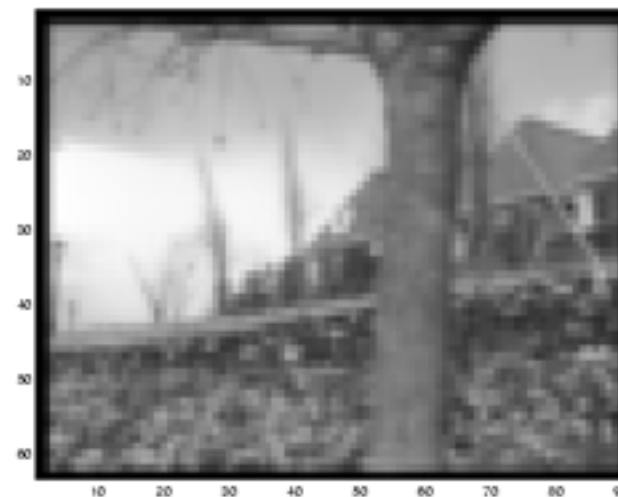
Проблема большого смещения



- Насколько мало движение в изображении?
 - Существенно больше 1-го пикселя
 - Как можно решить эту проблему?



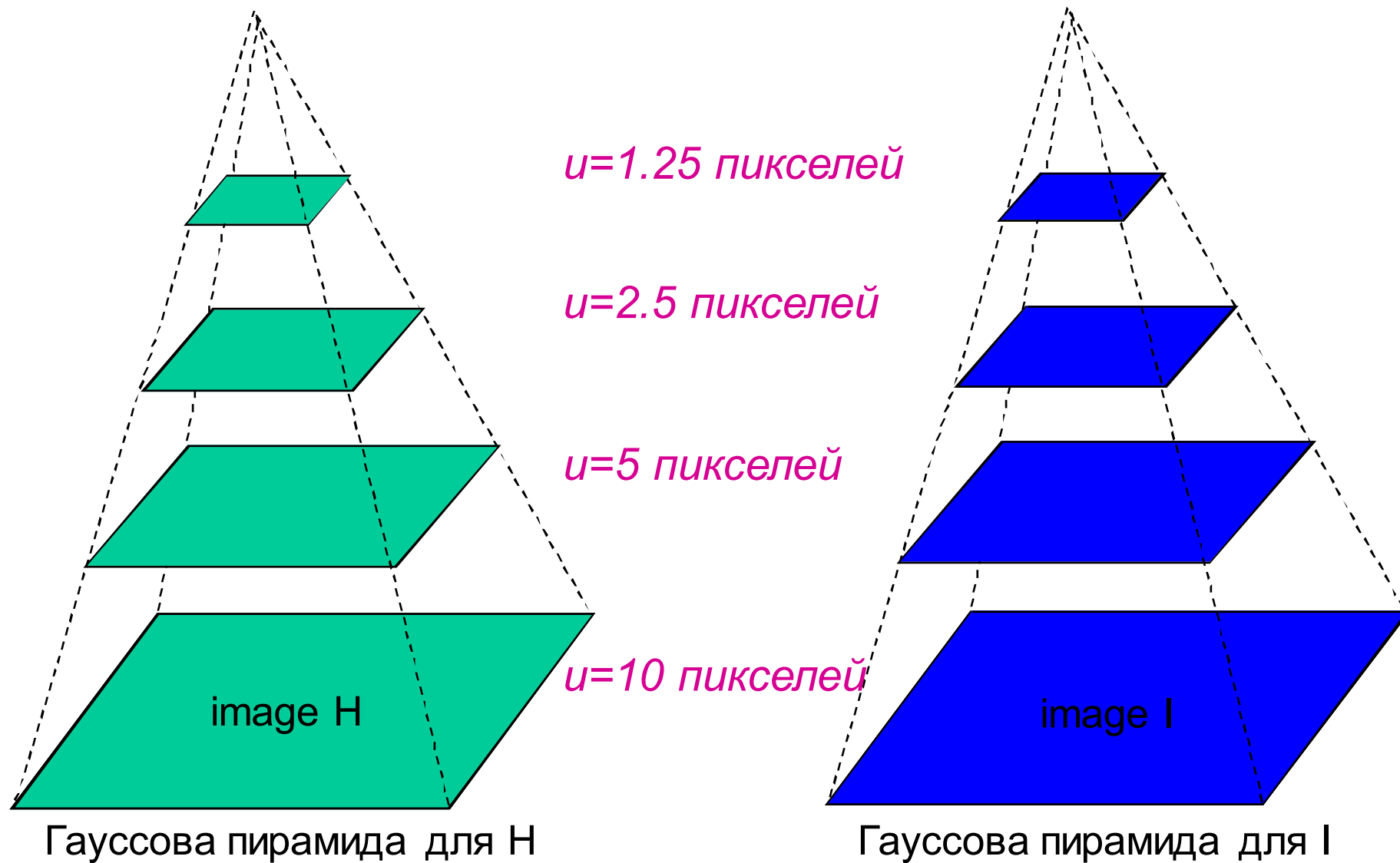
Пирамида разрешений



Насколько мало движение в изображении?

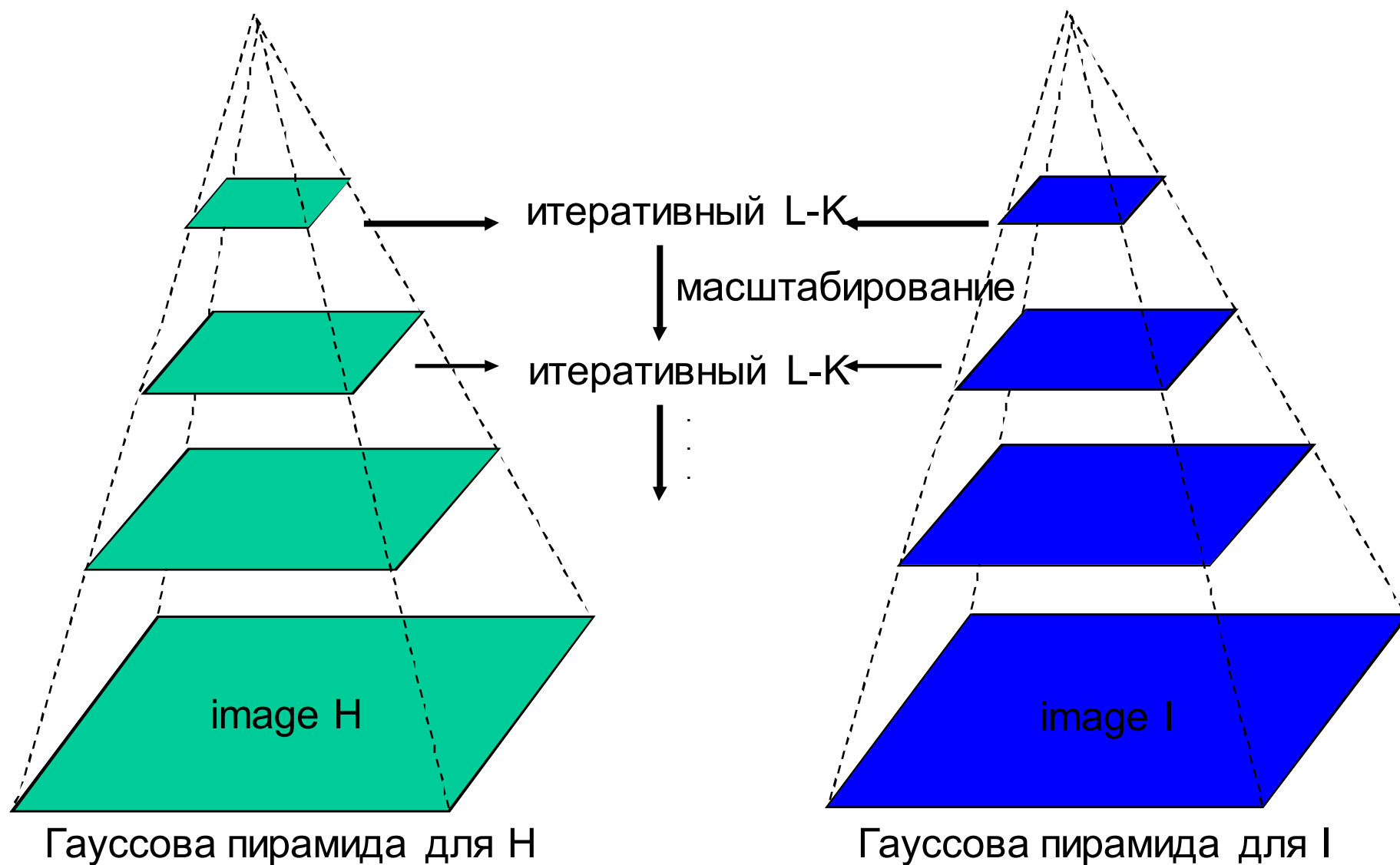


Иерархический метод





Иерархический метод





Резюме локального метода

- Локальный метод Лукаса-Канаде позволяет надежно оценивать поток для отдельных пикселей, но только в окрестности особых точек
- Для повышения качества для больших смещений используется многомасштабная версия алгоритма



Рейтинг алгоритмов

Average angle error	avg.	Dimetrodon (hidden texture)			Seashell (hidden texture)			Rock (Synthetic)			Grove (Synthetic)			Yosemite (Synthetic)			Venus (Stereo)			Moebius (Stereo)		
	rank	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Bruhn et al.	1.6	<u>10.99</u> ₃	9.41 ₁	14.22 ₃	<u>11.09</u> ₂	19.48 ₃	16.21 ₂	6.14 ₁	17.41 ₁	12.86 ₂	6.32 ₁	12.41 ₁	10.98 ₁	1.69 ₁	2.86 ₁	1.05 ₁	<u>8.73</u> ₂	31.46 ₃	8.15 ₂	5.85 ₁	10.12 ₂	8.80 ₂
Black and Anandan	2.1	9.26 ₁	10.11 ₃	12.08 ₁	<u>11.20</u> ₃	19.83 ₃	17.01 ₃	<u>7.67</u> ₃	18.44 ₃	16.80 ₄	<u>7.89</u> ₂	13.55 ₂	13.96 ₄	<u>2.65</u> ₂	4.18 ₂	1.88 ₂	7.64 ₁	30.13 ₁	7.31 ₁	<u>7.05</u> ₂	10.02 ₁	8.41 ₁
Pyramid LK	2.8	<u>10.27</u> ₂	9.71 ₂	13.63 ₂	9.46 ₁	18.62 ₁	12.07 ₁	<u>6.53</u> ₂	18.43 ₃	10.95 ₁	<u>8.14</u> ₃	15.08 ₃	12.78 ₂	<u>5.22</u> ₃	6.64 ₃	4.29 ₃	<u>14.61</u> ₄	36.18 ₄	24.67 ₅	<u>12.98</u> ₅	13.85 ₄	20.61 ₅
MediaPlayer™	4.1	<u>15.82</u> ₄	26.42 ₄	16.96 ₄	<u>23.18</u> ₄	27.71 ₅	21.78 ₄	<u>9.44</u> ₄	22.25 ₄	15.03 ₃	<u>10.99</u> ₄	18.15 ₅	13.64 ₃	<u>11.09</u> ₄	17.16 ₄	10.66 ₅	<u>15.48</u> ₅	43.56 ₅	15.09 ₄	<u>9.98</u> ₄	15.04 ₅	9.47 ₃
Zitnick et al.	4.2	<u>30.10</u> ₅	34.27 ₅	31.58 ₅	<u>29.07</u> ₅	27.55 ₄	21.78 ₄	<u>12.38</u> ₅	23.93 ₅	17.59 ₅	<u>12.55</u> ₅	15.56 ₄	17.35 ₅	<u>18.50</u> ₅	28.00 ₅	9.41 ₄	<u>11.42</u> ₃	31.46 ₃	11.12 ₃	<u>9.88</u> ₄	12.83 ₃	11.28 ₄

- Общий рейтинг алгоритмов вычисления оптического потока и набор сложных тестовых данных
- Как всегда, такой рейтинг и сложные данные стимулируют развитие алгоритмов
- <http://vision.middlebury.edu/flow/>

S.Baker et. Al. [A Database and Evaluation Methodology for Optical Flow](#), IJCV, 92(1):1-31, March 2011



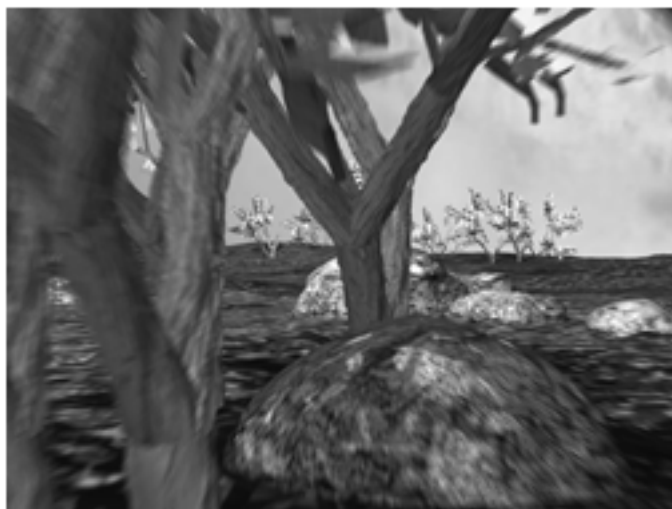
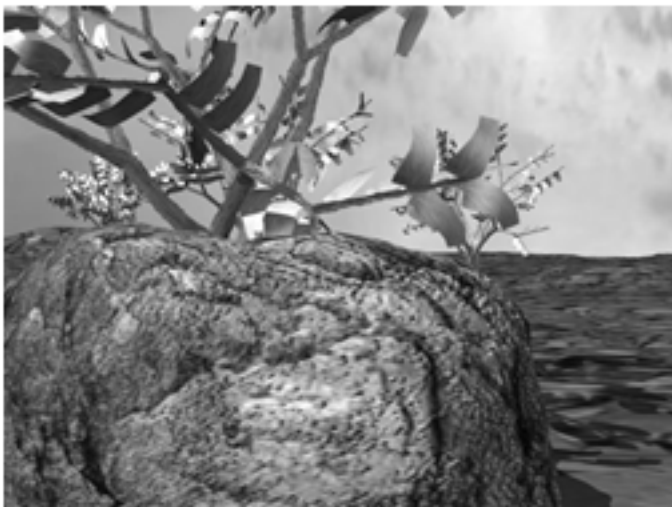
Создание сложных данных



- Для численной оценки алгоритмов нужны эталонные данные.
- Вопрос, как их можно получить для реальных данных?
- Подход:
 - Съёмка в двух диапазонах
 - Флюоресцентная краска создает высокочастотную текстуру, позволяющую оценить поле движения
- Плюс сложные синтетические данные

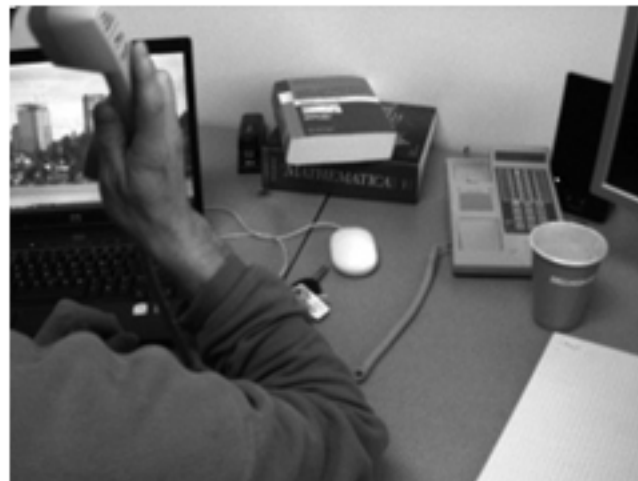
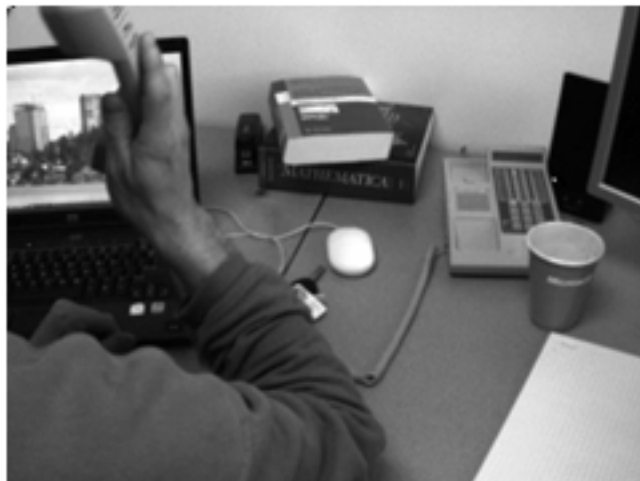


Синтетические данные





Интерполяция



Снимаем данные камерой 100Hz, оставляем каждый 4ый кадр.
Задача – по оптическому потоку проинтерполировать
промежуточные



Реализации

- OpenCV 2.0
 - GoodFeatureToTrack (Выбор особых точек (фактически, Harris))
 - cvCalcOpticalFlowPyrLK (Иерархическое расширение метода Lucas-Kanade для оптического потока)
- <http://gpu4vision.icg.tugraz.at>
 - Несколько алгоритмов оптического потока с ускорением на GPU
- <http://www.cs.brown.edu/~black/code.html>
 - Несколько C++ и Матлаб реализаций
- <http://people.csail.mit.edu/celiu/OpticalFlow/>
 - C++ и MEX

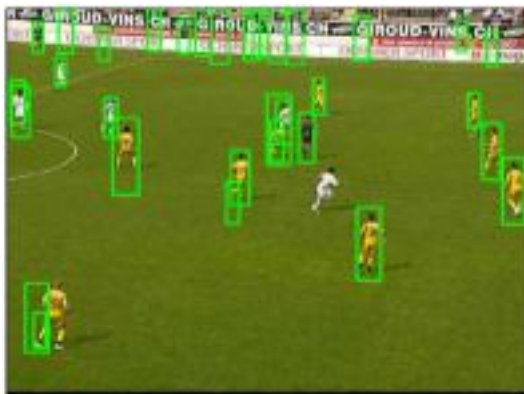


Резюме

- «Оптический поток» формализует понятие движения в видео
- Алгоритмы его вычисления бывают локальные и глобальные
- Методы можно разделить на качественные (и медленные) и быстрые (но менее точные)
- Для сравнения методов используется база Middlebury
- Есть целый ряд доступных реализаций



Система видеонаблюдения



- Камера, наблюдающая некоторую сцену (в закрытом или открытом пространстве)
- Необходимо выделить «объекты интереса» на каждом кадре
- Результат:
 - Ограничивающий прямоугольник
 - Попиксельная маска объекта интереса



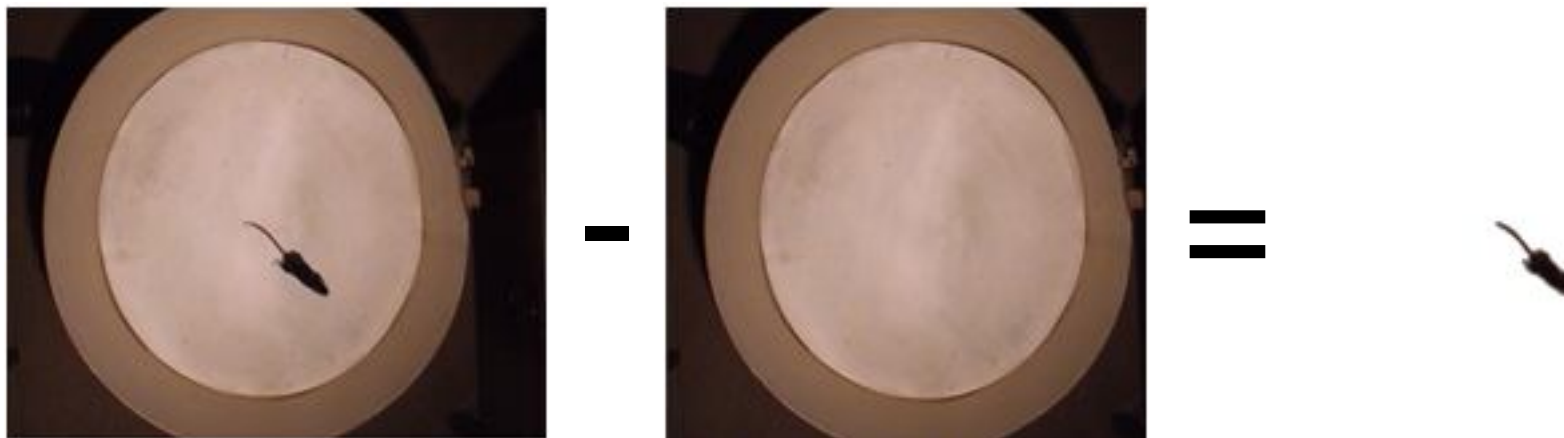
Свойства задачи



- Камеру считаем стационарной, не меняющей ракурса
- Фон считаем «стабильным», малоподвижным, незначительно меняющимся между кадрами
- Вывод – объекты должны отличаться от фона
- Ещё системы называют «детектор движения»



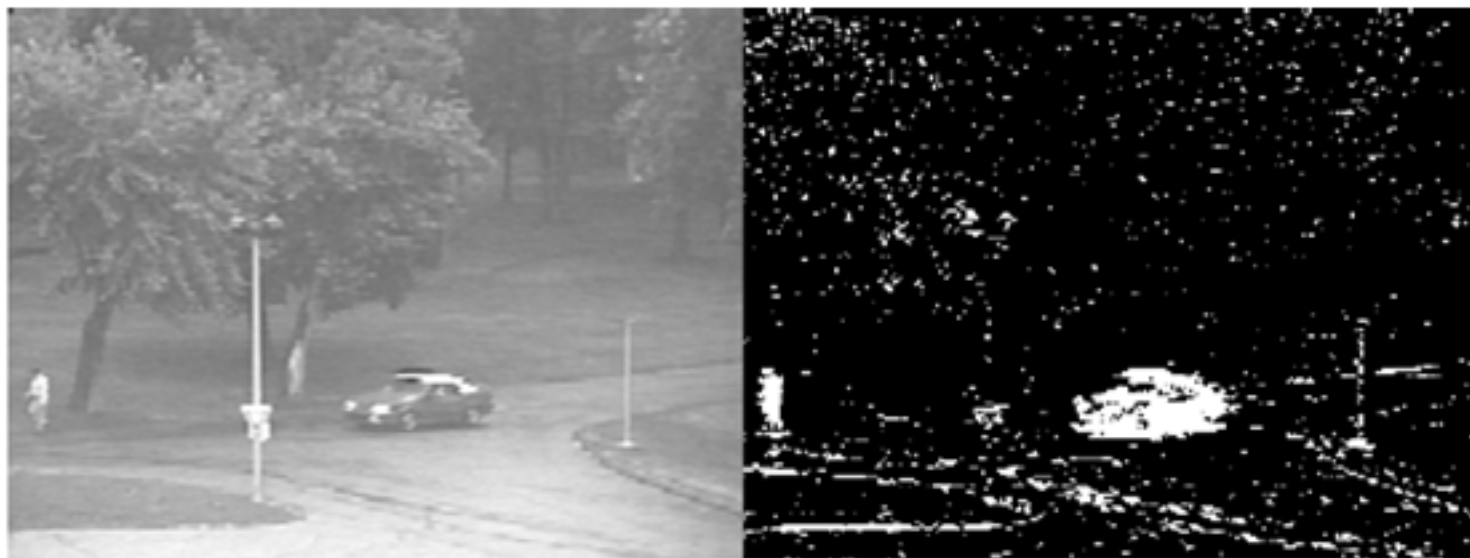
Вычитание фона



- Метод «Background subtraction»:
 - Возьмем «чистое» изображение без объектов – «фон» (*background*)
 - Вычтем фон из новых изображений с объектами
 - Сравним разницу для каждого пикселя с порогом
 - Порог – параметр алгоритма
 - Если разница больше порога - то считаем пиксель принадлежащим «переднему плану» (*foreground*)
 - Получаем бинарную маску «переднего плана»
 - Фон (0), передний план (1)



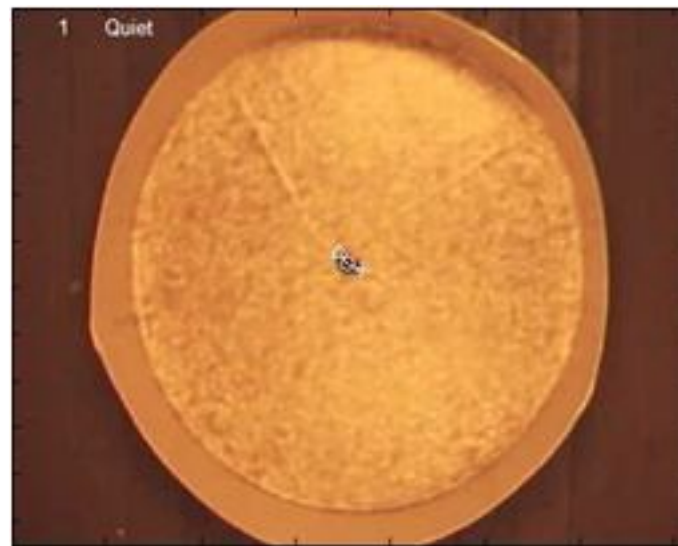
Обработка переднего плана



- Бинарная маска переднего плана обычно шумная и пиксели не связаны между собой
- Нам нужно:
 - Уменьшить шум в изображении (фильтрация)
 - Выделить области, потенциально соответствующие объектам (связанные компоненты)
- «Блоб» («капля», *blob*) - связанная компоненты маски переднего плана, потенциальный объект



Наблюдение за мышами



- Если объект в сцене может быть только один, то самый крупную связанную компоненту (блób) будем считать объектом
- Получили «базовый» алгоритм вычитания фона, который работает в ряде лабораторных задач (отслеживание мышей)
- Схема:
 - Попиксельное вычитание изображения фона из текущего кадра
 - Сравнение попиксельной разницы с порогом
 - Фильтрация маски
 - Выделение связанных компонент и выбор самой большой

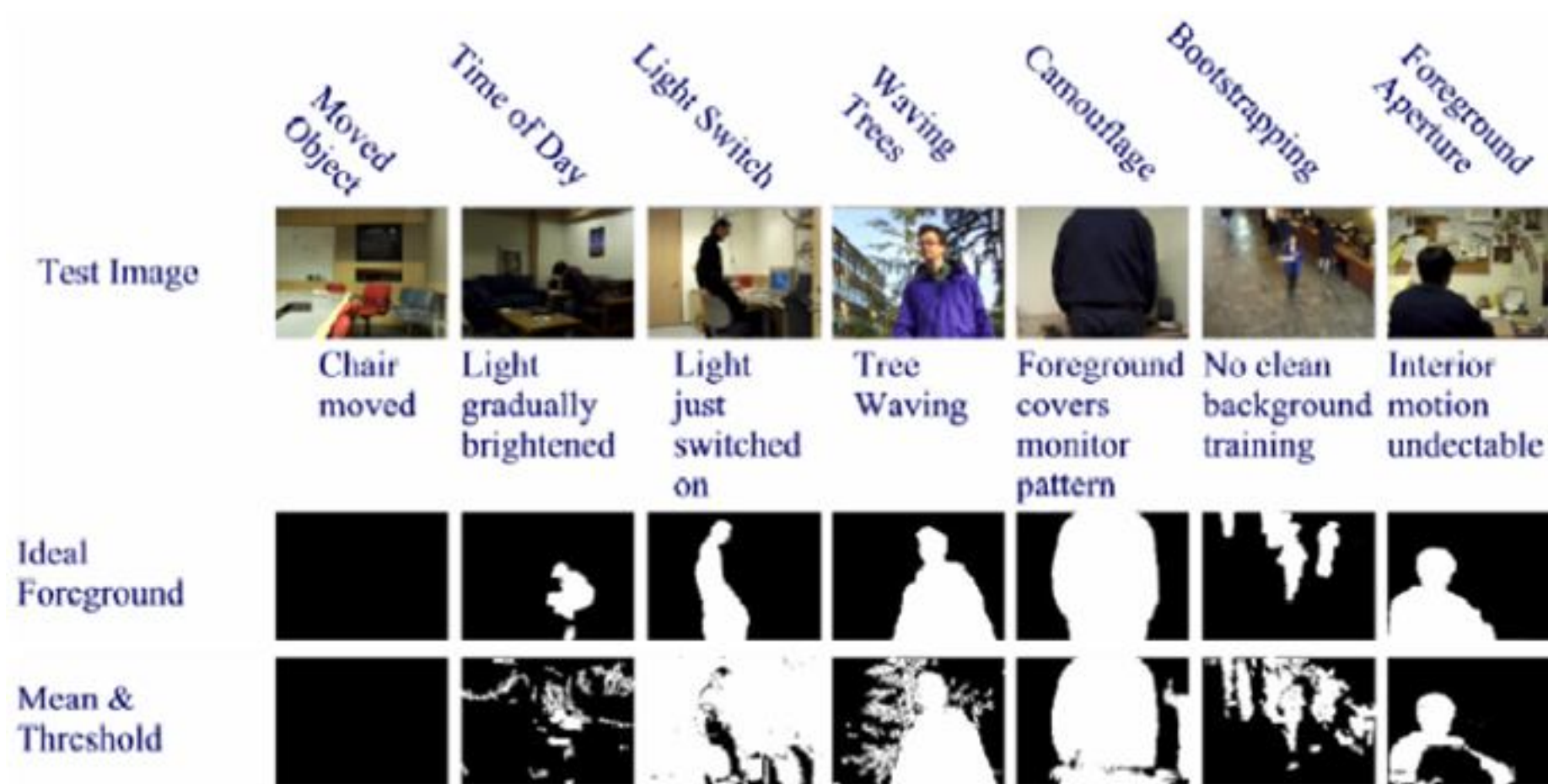


Если объектов много?

- Случай множества связанных компонент – это уже задача отслеживания объекта
- Некоторые подходы к ней рассмотрим на следующей лекции



Проблемы вычитания фона



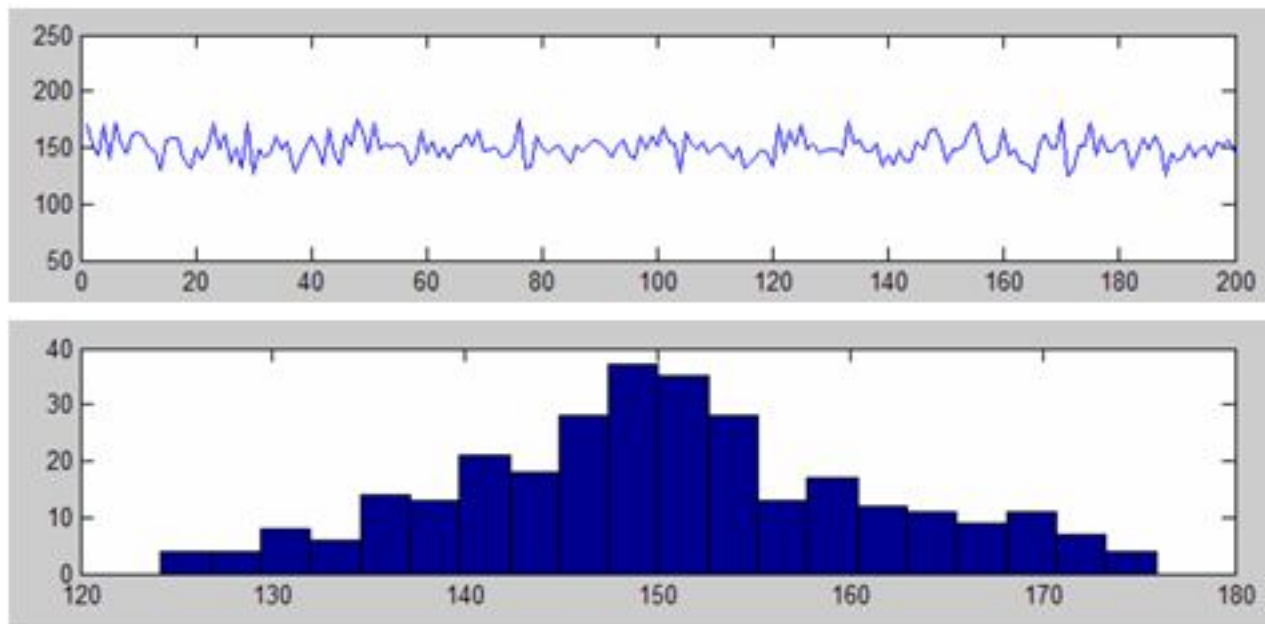
Плюс камера обычно «шумит»

from K. Toyama et al.



Моделирование фона

- Реальная статичная сцена:
 - «Шум камеры»
 - Интенсивность меняется в небольших пределах вокруг какого-то значения
 - Нормальное (Гауссово) распределение



Wren, Christopher R., Ali Azarbayejani, Trevor Darrell, and Alex Pentland. "Pfinder: Real-Time Tracking of the Human Body," IEEE PAMI, 1997



Оценка параметров модели

- По чистому видеоролику для каждого пикселя вычислим параметры фона для модели – нормального распределения (случай одного канала):

- Вычисляем среднее

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

- Вычисляем дисперсию

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2$$



Вычитание фона

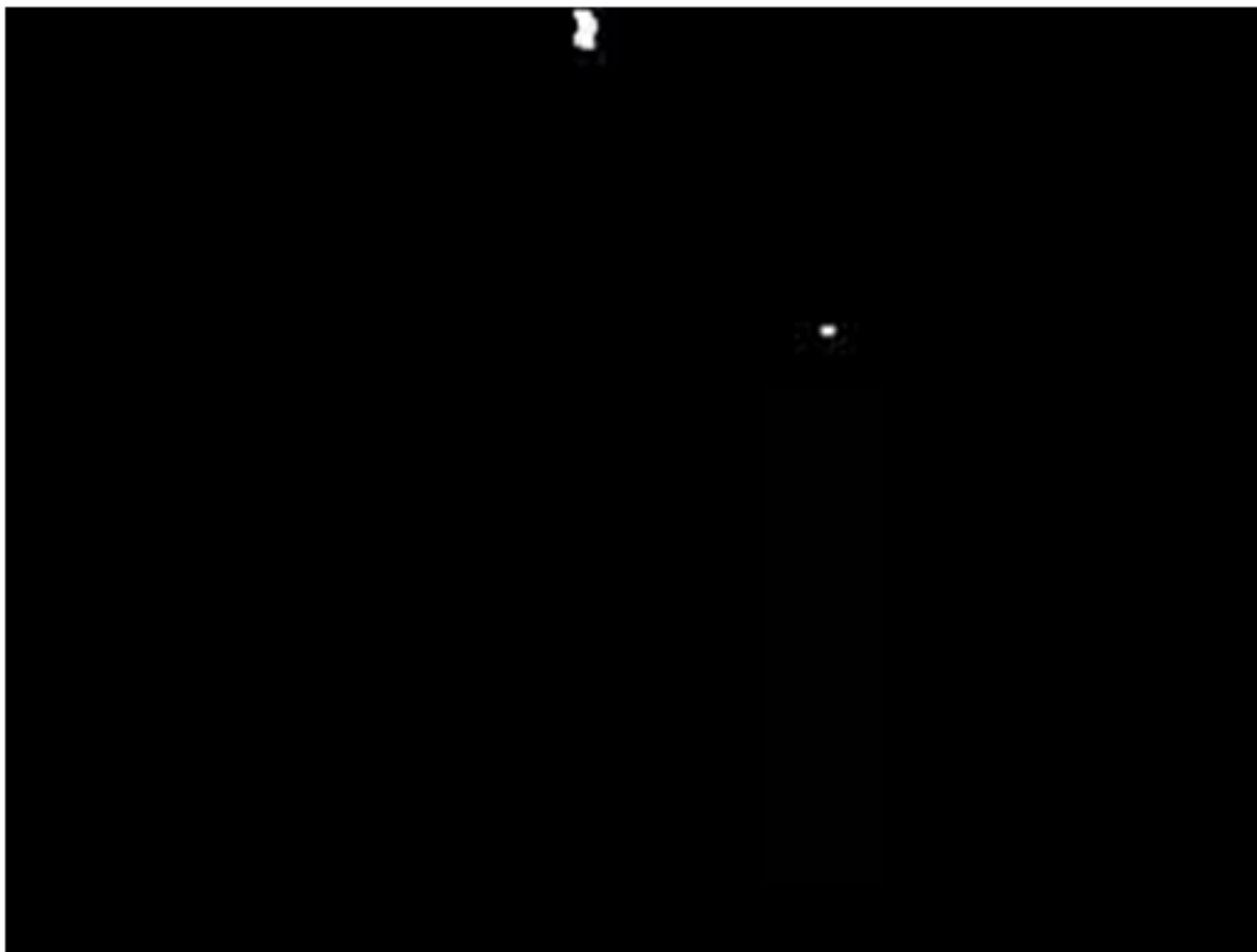
- Вероятность x при вычисленных параметрах модели фона:

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

- Проще использовать сравнение с порогом
- Считаем, что все пиксели фона должны попасть в интервал $(\mu-3\sigma, \mu+3\sigma)$
- Если $|x-\mu| > 3\sigma$, тогда x – пиксель переднего плана



Пример работы





Длительное время работы

- Как меняется фон на длинных периодах времени?
- Может плавно меняться освещенность сцены
- Что это скажется на работе алгоритма?
- Что мы можем сделать для того, чтобы алгоритм мог работать?
- Нам нужно дополнить алгоритм этапом обновления модели фона





Общая схема вычитания фона

- *Initialize_background_model()*
- *For t = 1:N*
 - *Compute_frame_difference()*
 - *Threshold_frame_difference()*
 - *Noise_removal()*
 - *Update_background_model()*
- *end*

Как обновить модель фона для случая одной гауссианы?



Обновление параметров модели

Одноканальное изображение (серое)

- 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)



Многоканальное изображение

- Можно рассчитать полную матрицу ковариации K
- Обновлять полную матрицу сложно
- Обновление только среднего, с сохранением ковариации

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



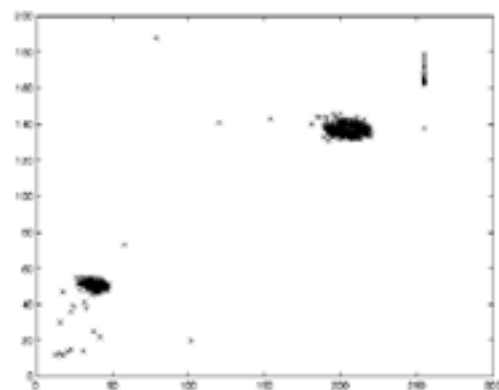
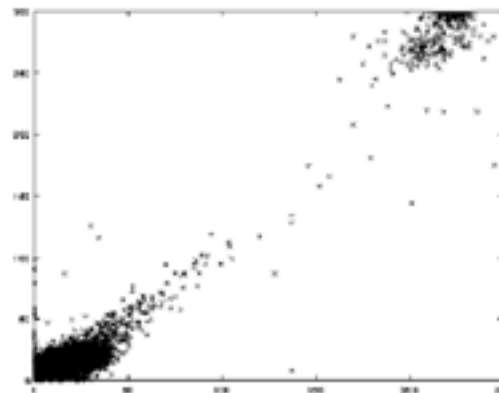
Как настраивать модель?

- Если есть размеченные данные:
 - Оценка точности и полноты (precision/recall) на наборе видеороликов
 - Выбор параметров, оптимизирующих желаемую точность
- Если нет размеченных данных
 - Можно взять видео без объектов
 - Настроить параметры таким образом, чтобы получить заданное количество ложных обнаружений
 - Минимальные пороги для достижения заданной «чувствительности»





Более сложный пример

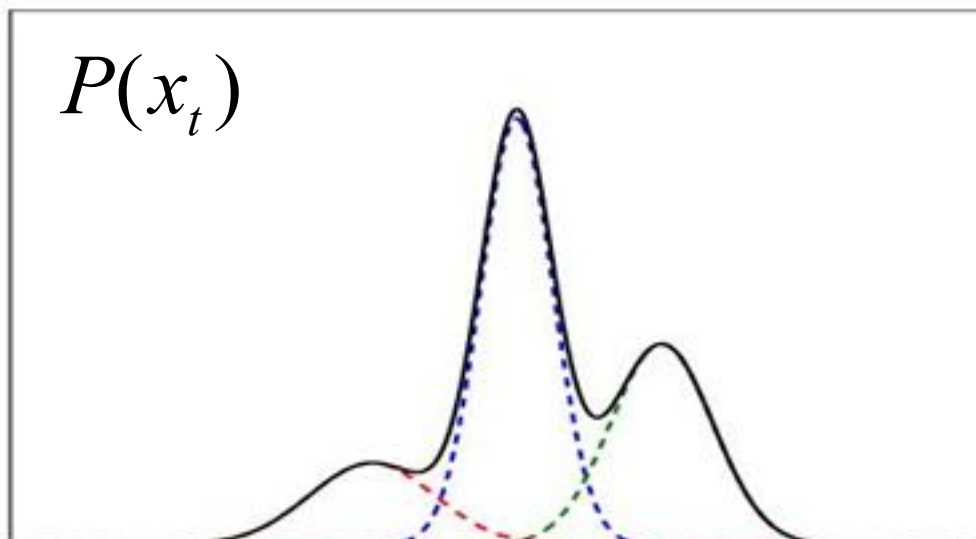
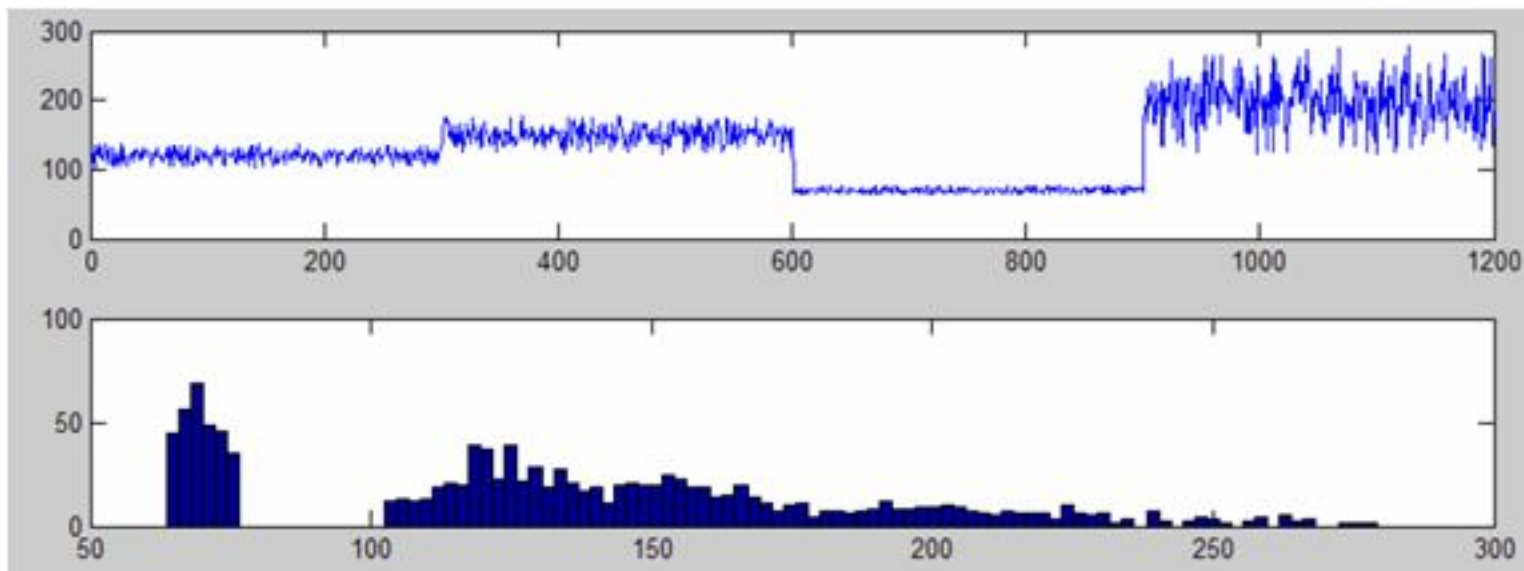


Распределение интенсивности красного
и зеленого каналов

Как моделировать фон в таких, более сложных,
случаях?



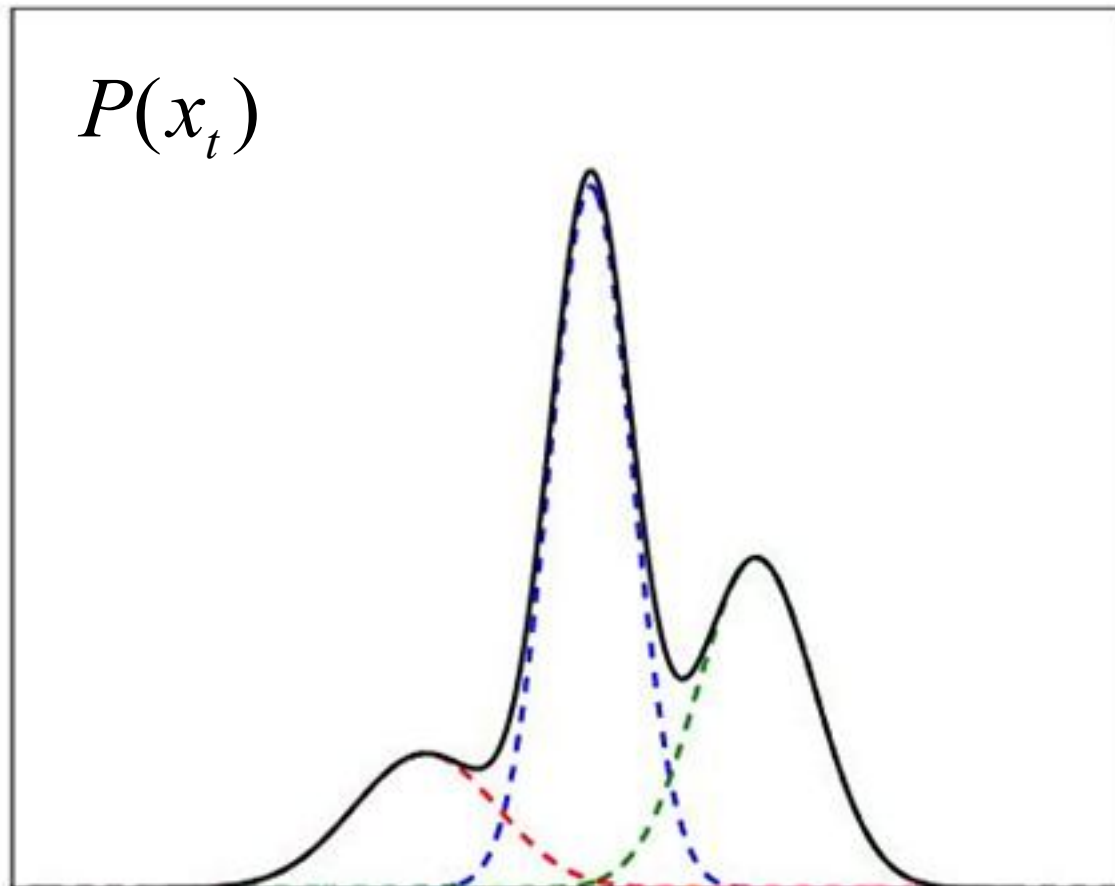
Более сложные модели



«Многомодальное распределение» - значения могут группироваться в несколько кластеров



Смесь Гауссиан



$$P(x_t) = \sum_{i=1}^K \omega_{i,t} N(x_t, \mu_{i,t}, \sigma_{i,t})$$

Мы полагаем, что
сигнал порождён
несколькими
независимыми
моделями

Пример №1:

- Вода
- Блик на воде

Пример №2:

- Небо
- Листья на дереве



Смесь Гауссиан

- Плотность вероятности яркости пикселя x при модели смеси из K нормальных распределений

$$P(x_t) = \sum_{i=1}^K \omega_{i,t} N(x_t, \mu_{i,t}, \sigma_{i,t})$$

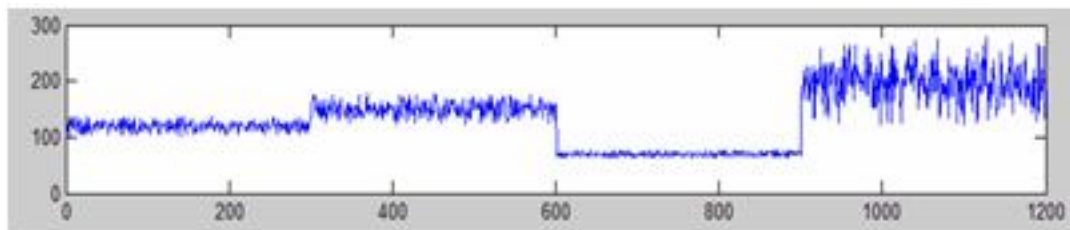
$\omega_{i,t}$ - вес компоненты i в момент t

- Вычисление смеси требует сложного EM (Expectation-Maximization) алгоритма (не в реальном времени), поэтому был предложен приближенный алгоритм



Обучение смеси на лету

- Пусть N – количество компонент в смеси
 - Инициализируем 1ую компоненту по первому изображению, вес = 1, вес остальных – 0
 - Сравниваем пиксель с каждой компонентой, пока не найдем совпадения
 - Обновляем матожидание и дисперсию совпавшей компоненты
 - Если совпадения не найдено, то заменяем компоненту с наименьшим весом
 - Обновляем веса



$$\sum_{i=1}^K \omega_{i,t} N(x_t, \mu_{i,t}, \sigma_{i,t})$$



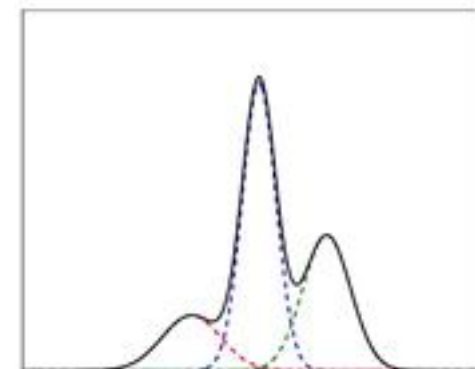
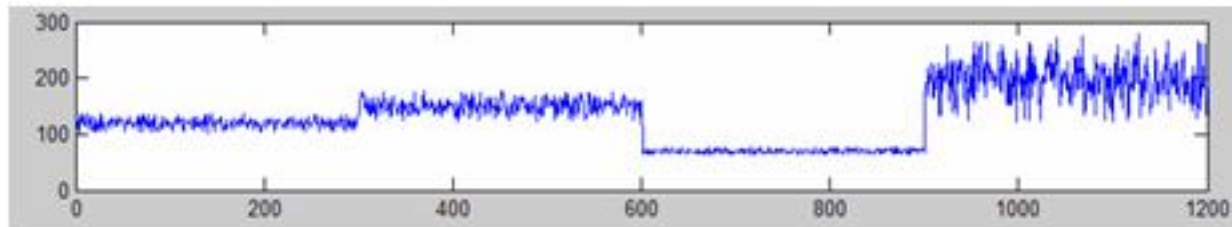
Обновление весов

Обновление весов:
$$P(x_t) = \sum_{i=1}^K \omega_{i,t} N(x_t, \mu_{i,t}, \sigma_{i,t})$$

$$\omega_{i,t} = (1 - a)\omega_{i,t-1} + aM_{i,t}$$

$M_{i,t} = 1$, если интенсивность пикселя
удовлетворяет i -ой компоненте

После обновления всех весов, они нормализуются





Моделирование фона

- Упорядочим все компоненты по критерию ω_i / σ
 - Чем «стабильнее» компонента, тем выше
 - Чем больше вес (чаще встречается, тем тоже выше)
- Определим порог T - какая доля выборки для каждого пикселя должна соответствовать фону
- Тогда фон для каждого пикселя:

$$B = \arg \min_b \left(\sum_{i=1}^b \omega_i > T \right)$$

- Все остальные компоненты - объекты



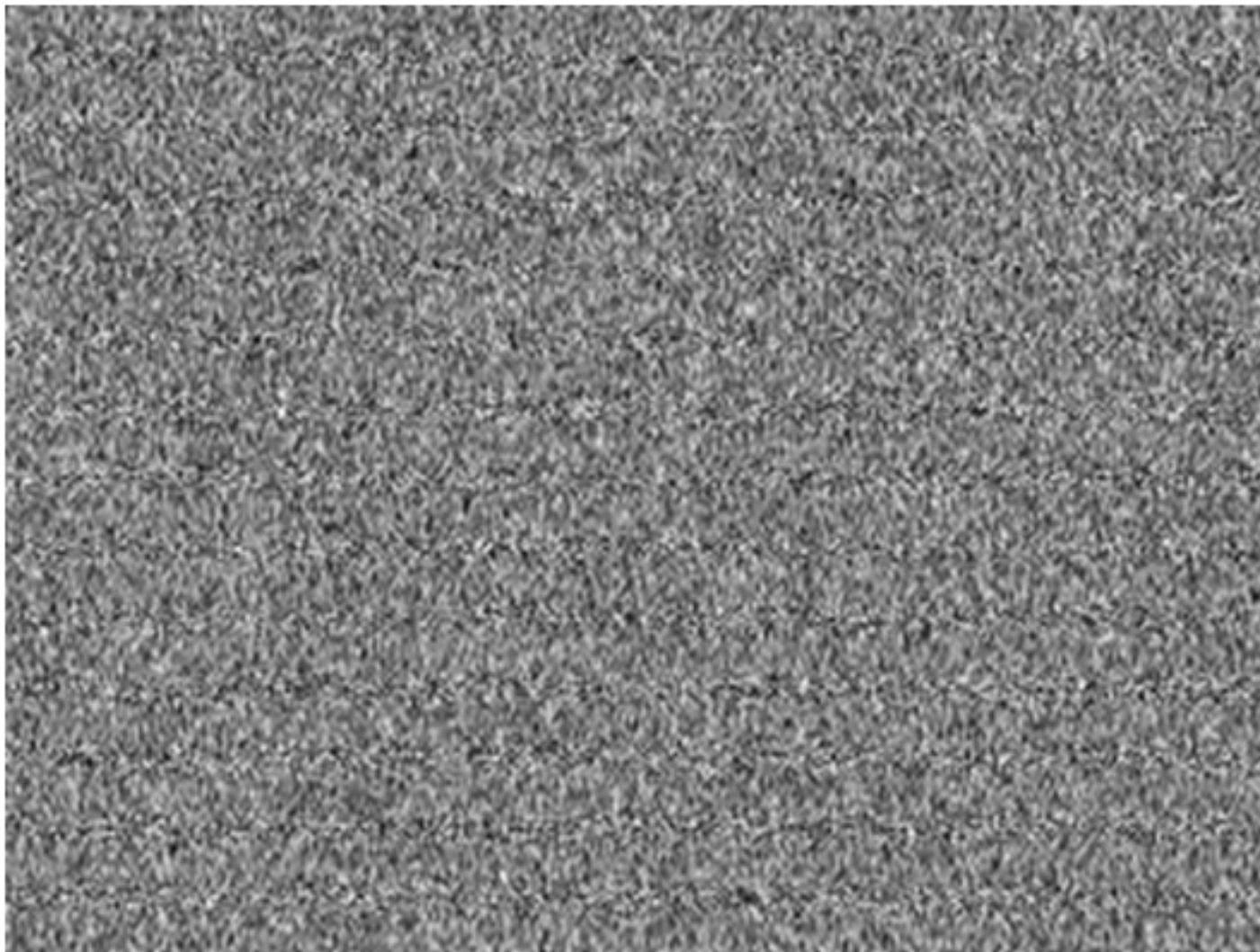
Иллюстрация работы



1. Текущий кадр видео (момент t)
2. Старшая гауссиана для каждого пиксела (модель фона)
3. Вторая гауссиана для каждого пиксела (модель движущегося объекта)
4. Маска переднего плана для текущего кадра



Пример визуализации фона



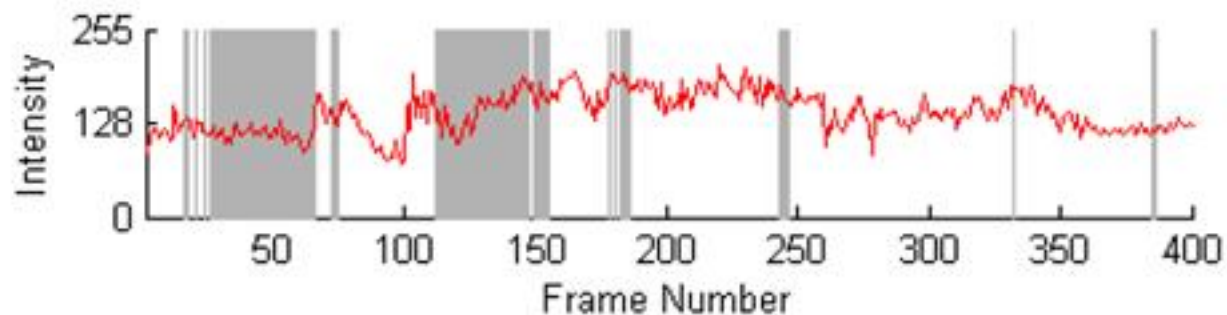


Ещё посложнее пример

Камера наблюдения в лесу у кормушки для птиц



График яркости одного из пикселей



Как работают алгоритмы и почему?



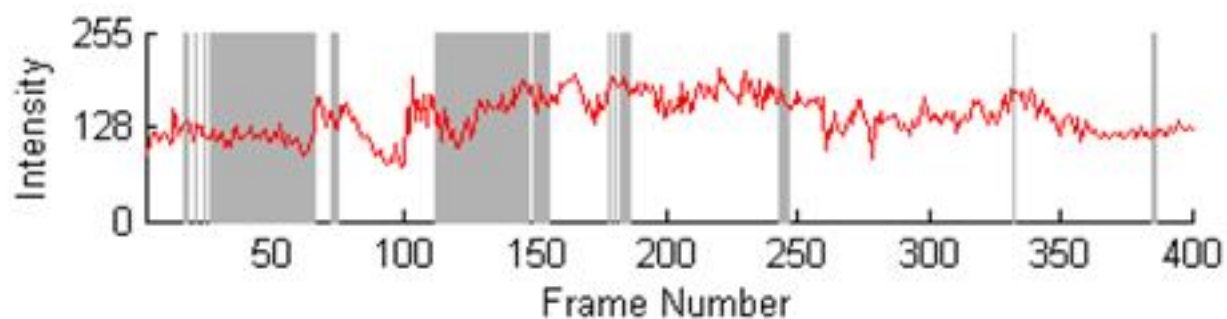
Как быть?



Будем анализировать окрестность каждого пиксела



Статистики по окрестностям



[Ko, T.\[Teresa\]](#), [Soatto, S.\[Stefano\]](#), [Estrin, D.\[Deborah\]](#), Background Subtraction on Distributions, *ECCV 2008*



Схема алгоритма

- Посчитаем статистику (гистограмму) по 3D окрестности пикселя (фон):

$$p_{ij}(x) = \frac{1}{|S|} \sum_{s \in S} \delta(s - x)$$

$$S = \{x_t(a, b) \mid |a - i| < c, |b - j| < c, 0 \leq t < T\}$$

- Посчитаем статистику по 2д окрестности пикселя на текущем кадре:

$$q_{ij,\tau}(x) = \frac{1}{|S_\tau|} \sum_{s \in S_\tau} \delta(s - x)$$

$$S_\tau = \{x_\tau(a, b) \mid |a - i| < c, |b - j| < c\}$$

- Вычислим разницу:

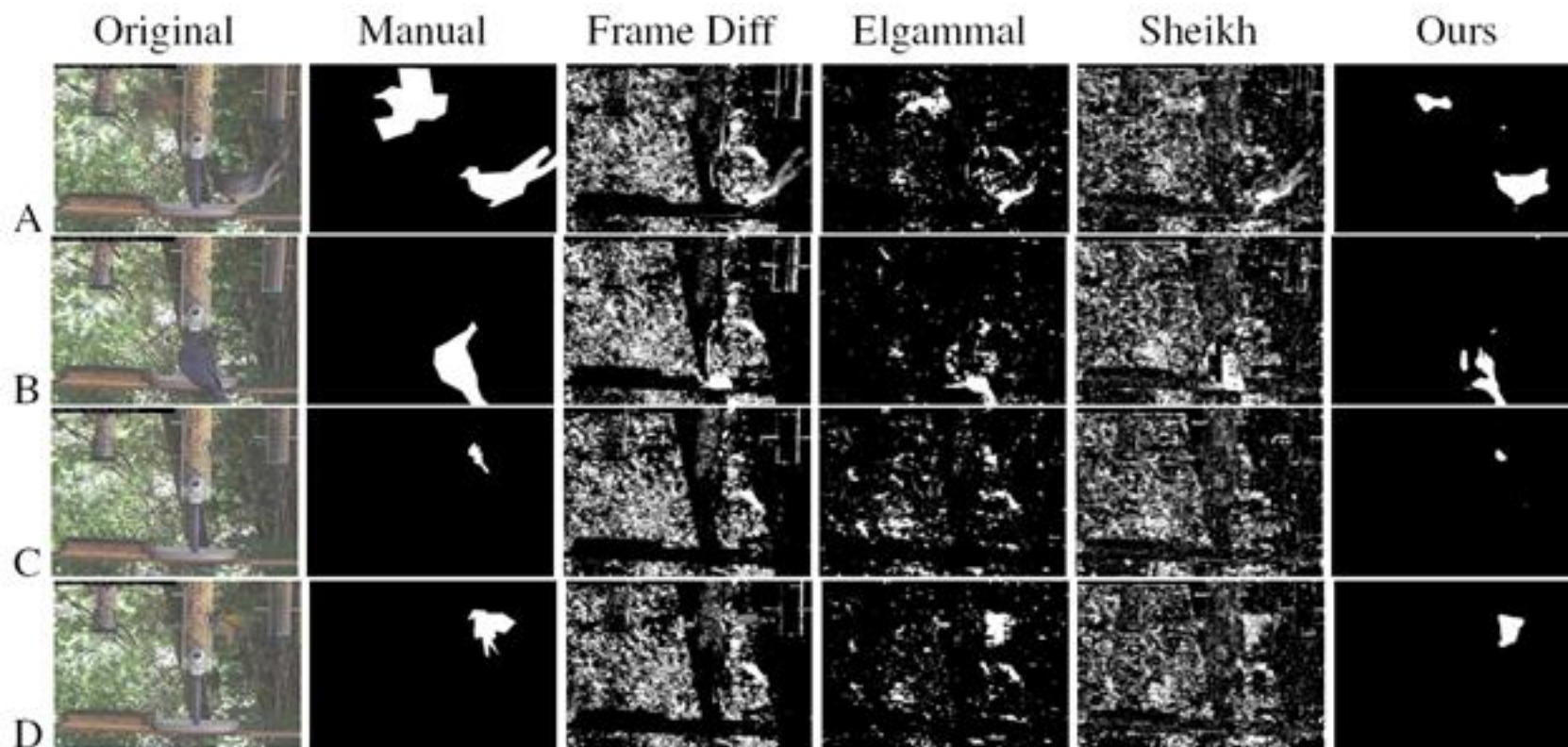
$$d = \int_X \sqrt{p_{ij,\tau-1}(x) q_{ij,\tau}(x)} dx$$

- Обновление модели:

$$p_{ij,t}(x) = (1 - \alpha) p_{ij,\tau-1}(x) + \alpha q_{ij,\tau}(x)$$



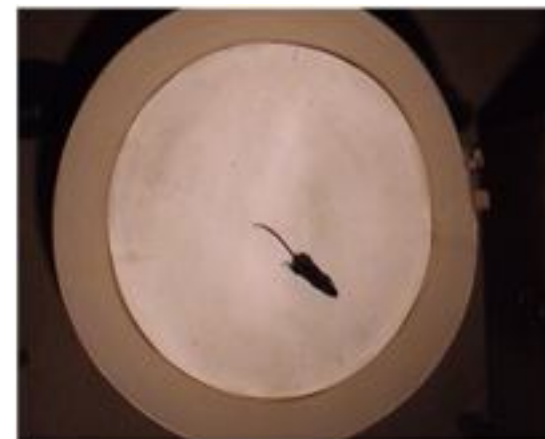
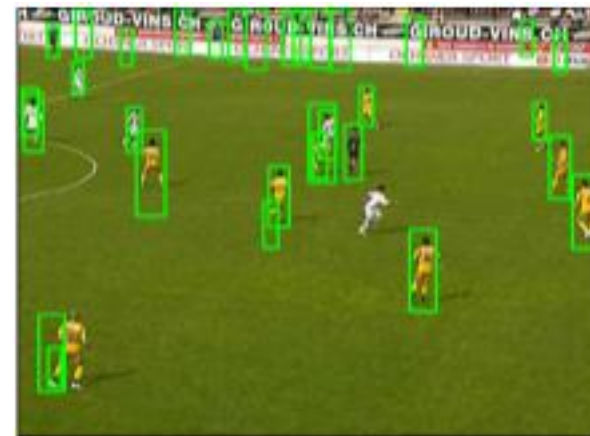
Результаты работы





Глобальные модели цветов

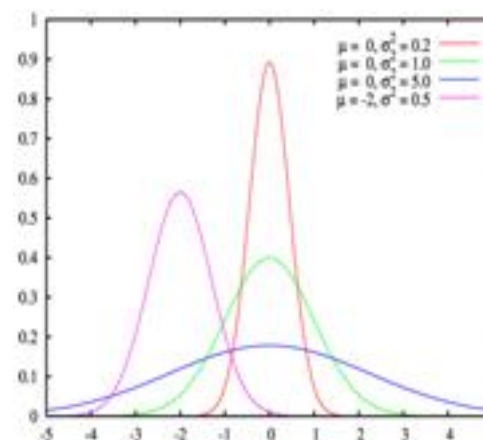
- Если у объекта и фона характерные цвета, то почему бы это не использовать?
- Построим цветовые модель объекта и модель фона
- Для каждого пикселя будем сравнивать вероятности принадлежности пикселя обоим моделям и выбирать ту, вероятность для которой будет выше





Модель фона

- Построим и глобальную, и локальную модели фона
- Глобальная: смесь гауссиан (GMM)



($K_b = 10 - 15$)

$$p(I_r | x_r = B) = \sum_{k=1}^{K_b} \omega_k^b N(I_r | \mu_k^b, \Sigma_k^b)$$



Пример медианного фона



-



=





Данные



- На пути создания общедоступных тестовых баз два препятствия:
 - Приватность
 - Сложность разметки данных
- PETS Performance Evaluation of Tracking and Surveillance
- <http://www.cvg.rdg.ac.uk/PETS2009/a.html>

Пример



Пример слежения с реальной системы видеонаблюдения



Развитие методов

- Другие признаки для моделирования фона
- Более сложные модели и классификаторы фон / объект
- Подавление теней и отражений
- Обработка нестандартных ситуаций (резкое изменение освещения)
- Иерархические методы
- Ускорение вычислений (параллельные реализации, GPU и т.д.)



Резюме вычитания фона

- Вычитание фона – очень эффективный инструмент выделения объектов при условии стационарности камеры
- Почти вся видеоаналитика в современных системах основана на вычитании фона
- Есть множество методов вычитания фона, четыре из которых мы рассмотрели
- Точное вычитание фона в реальном времени для HD разрешений сделать крайне сложно