



Сегментация видео на основе Mean-shift

Сергей Матюнин

Video Group

CS MSU Graphics & Media Lab



Содержание

- **Введение**
- Анизотропный подход
- Обновление Mean-shift
- Ускорение Mean-shift
- Заключение

Введение

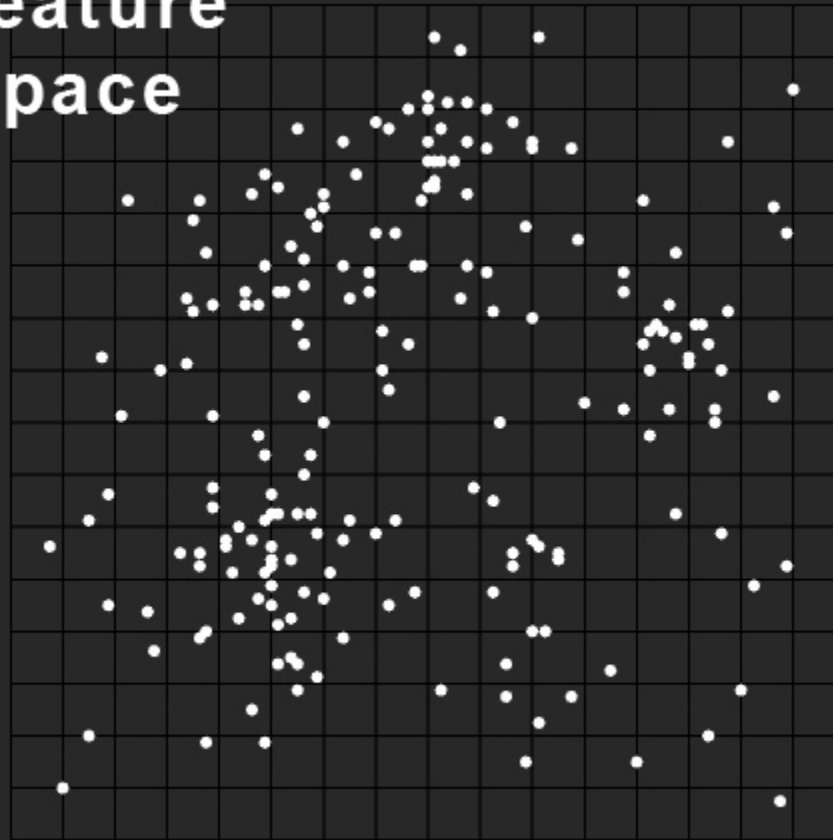


- Почему Mean-shift?
 - Распространенный
 - Простая идея
 - Простая реализация
 - Относительно невысокая вычислительная сложность
 - Возможность комбинировать с другими методами
- Главный недостаток – необходимость подбирать параметры

Введение

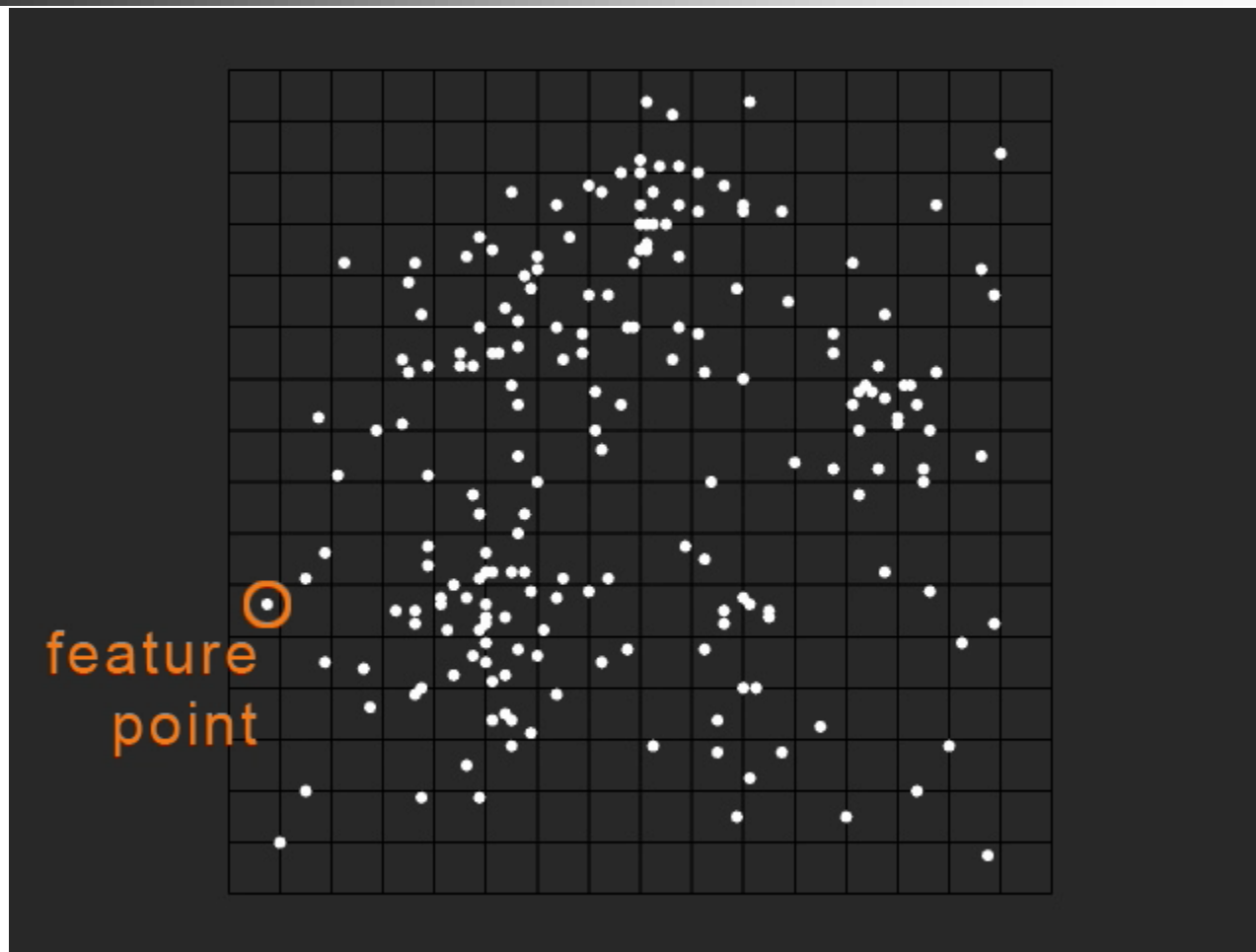
Mean-shift Segmentation

feature
space



Введение

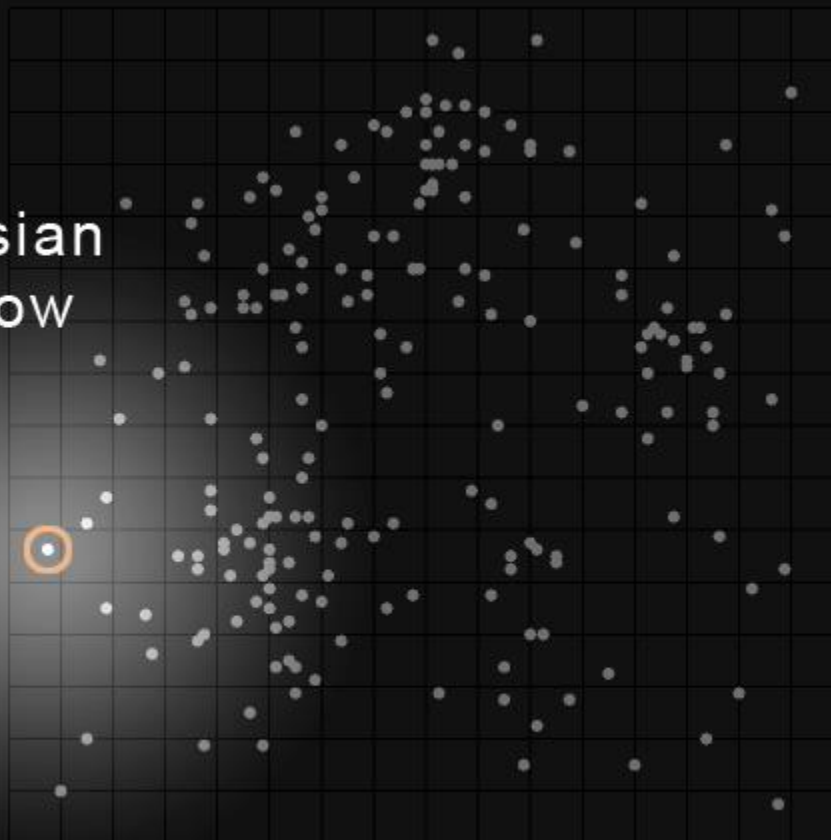
Mean-shift Segmentation



Введение

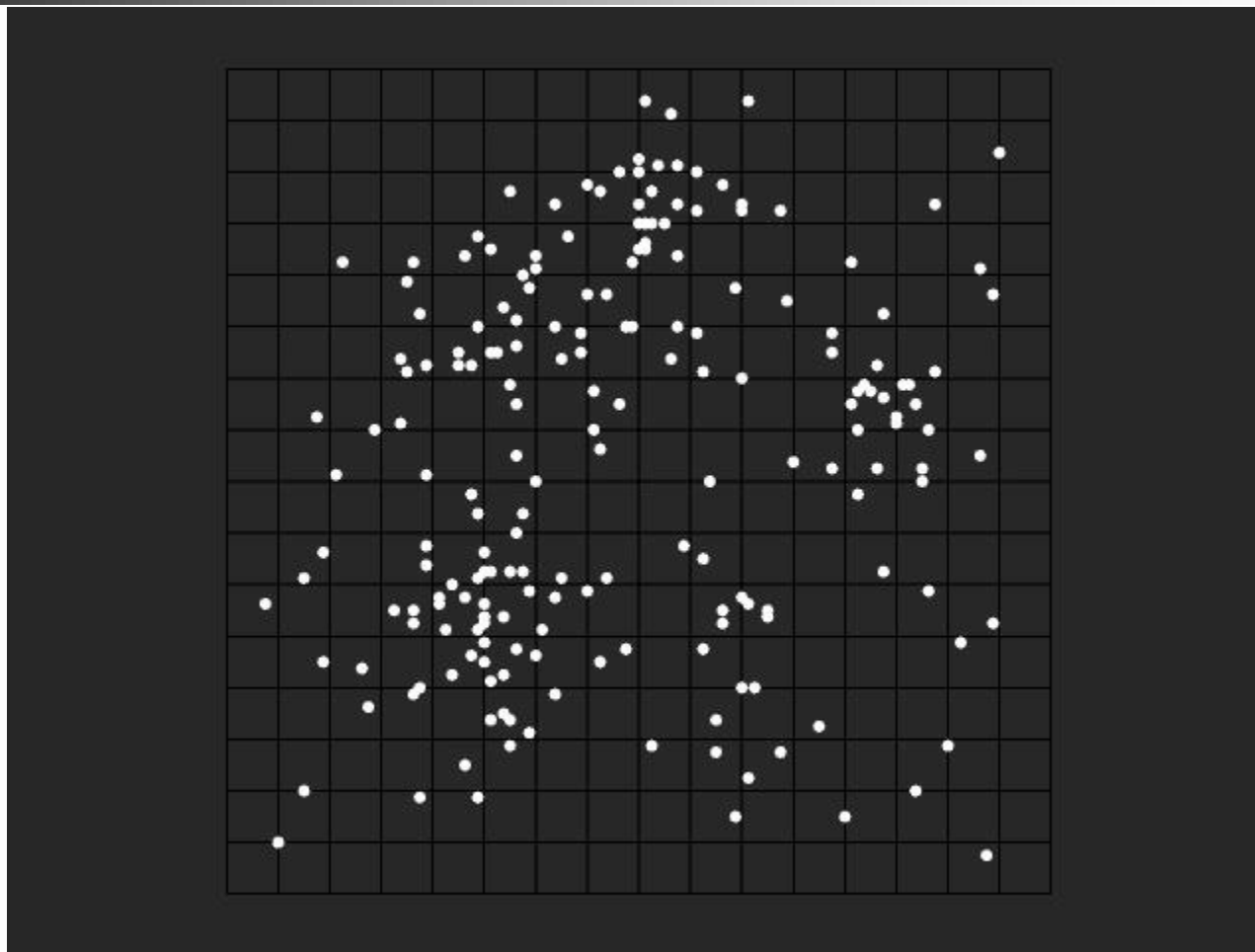
Mean-shift Segmentation

Gaussian
window



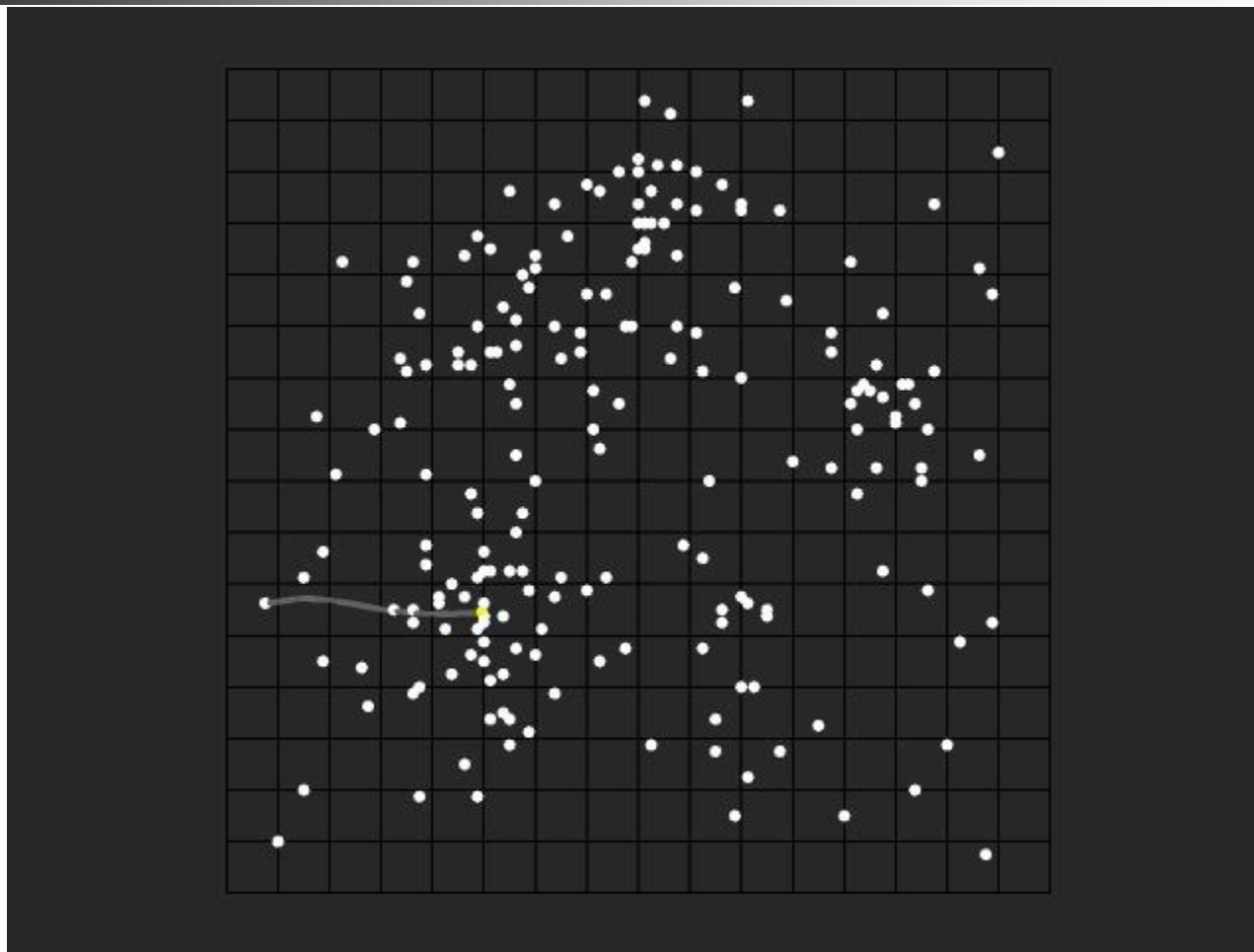
Введение

Mean-shift Segmentation



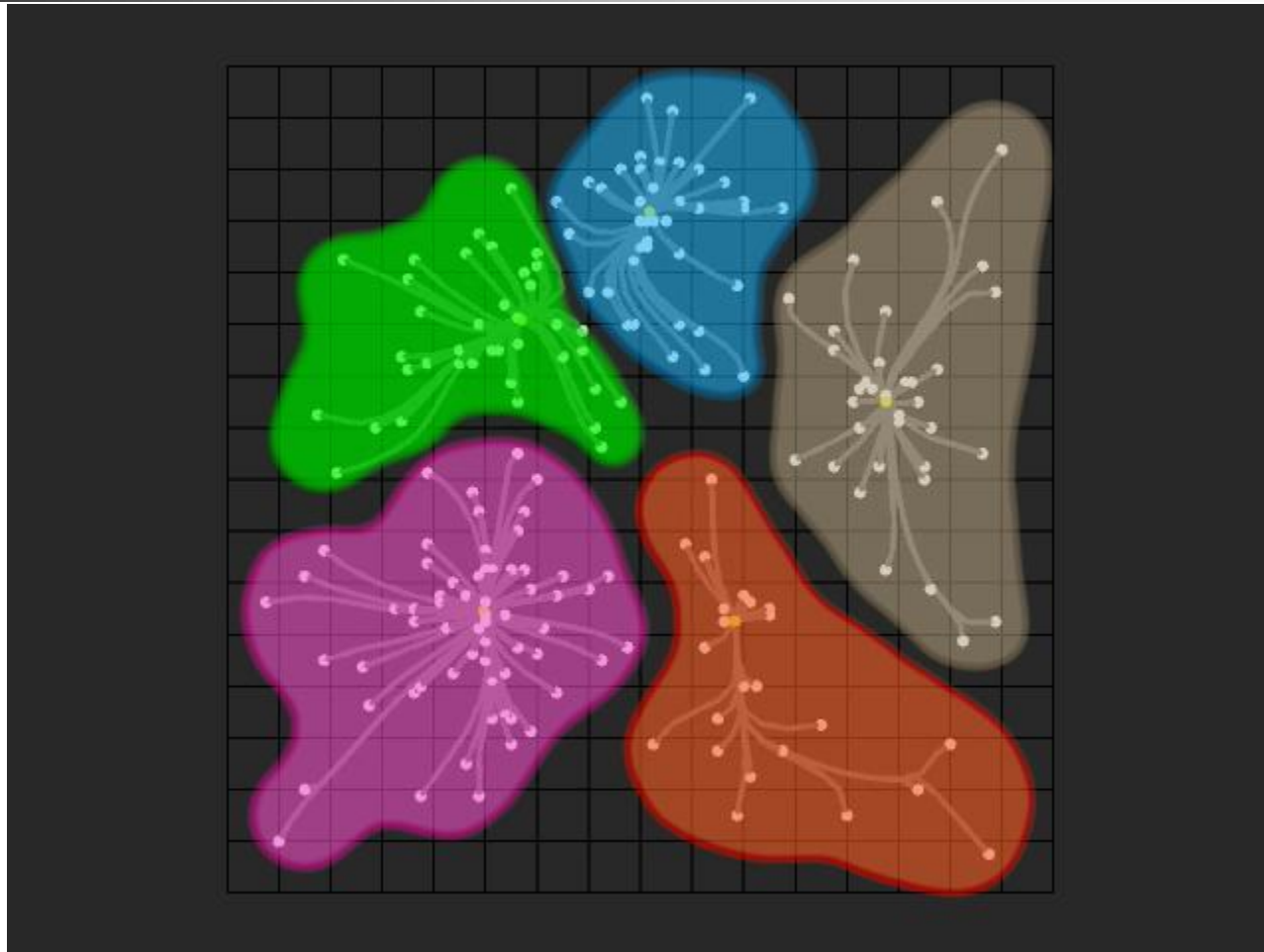
Введение

Mean-shift Segmentation



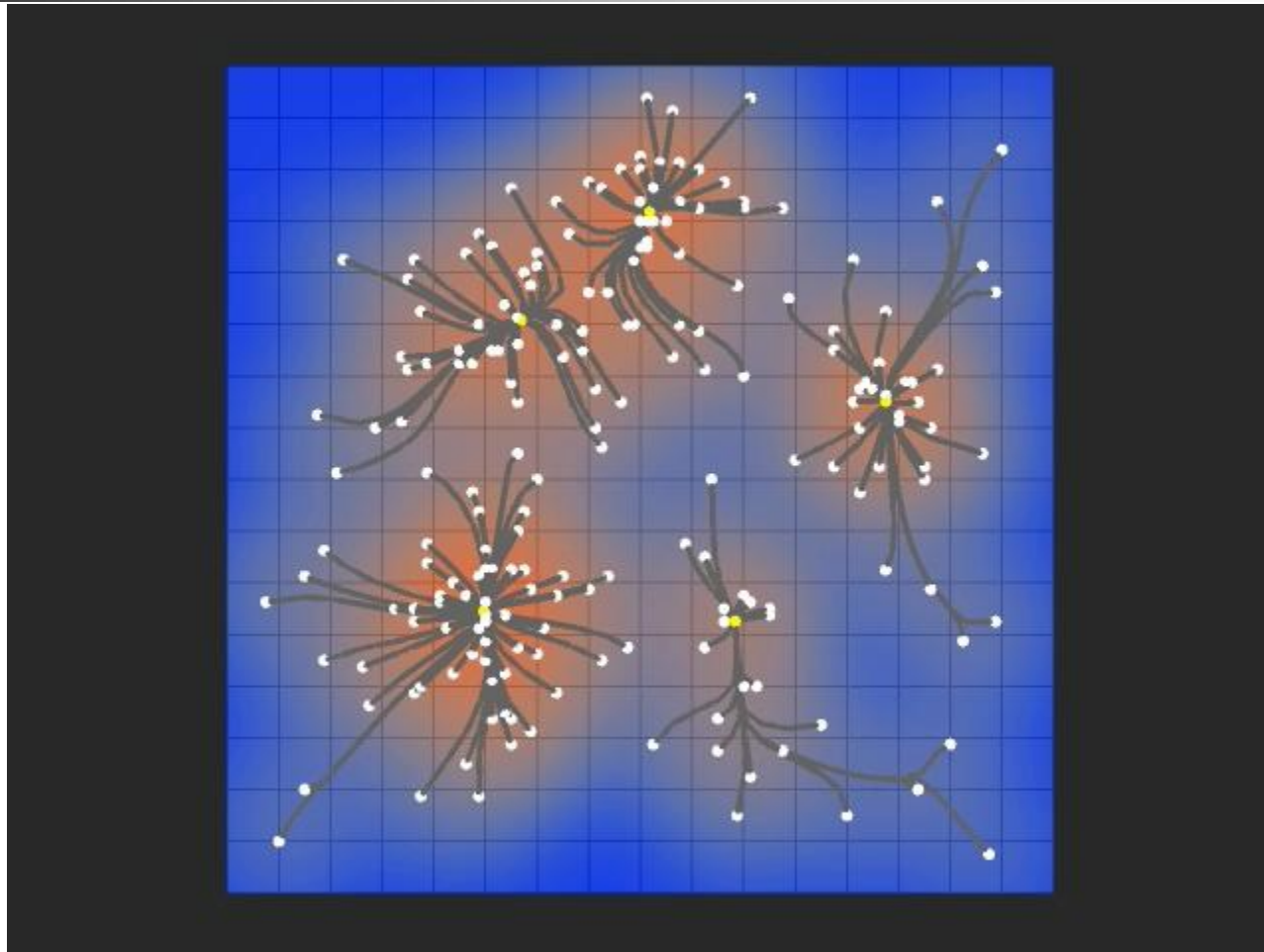
Введение

Mean-shift Segmentation



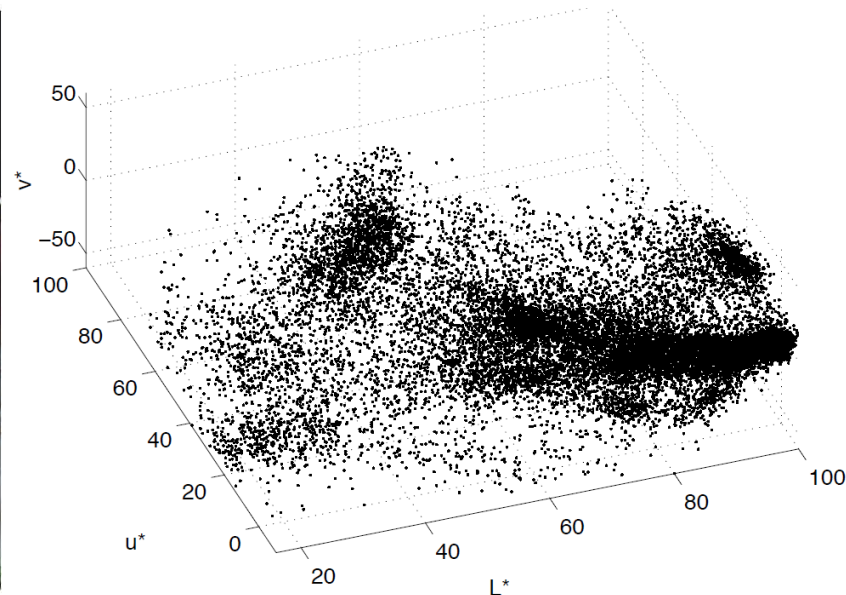
Введение

Mean-shift Segmentation



Mean-shift

Уменьшение сложности



$$f(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^n K \left(\frac{\mathbf{x} - \mathbf{x}_i}{h} \right)$$

Можно хранить в памяти, какие точки соседние

Mean-shift

Выбор функции ядра

- Выбор функции ядра (форма, размер) – критичная задача
- Ядро задает сходство точек, которые нужно поместить в один сегмент

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n K_H(x - x_i)$$

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{c}{(h_i^s)^p (h_i^r)^q} k^s \left(\left\| \frac{x^s - x_i^s}{h_i^s} \right\|^2 \right) k^r \left(\left\| \frac{x^r - x_i^r}{h_i^r} \right\|^2 \right)$$



Содержание

- Введение
- **Анизотропный подход**
- Обновление Mean-shift
- Ускорение Mean-shift
- Заключение

Anisotropic Mean-shift

Выбор ядра

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{c}{(h_i^s)^p (h_i^r)^q} k^s \left(\left\| \frac{x^s - x_i^s}{h_i^s} \right\|^2 \right) k^r \left(\left\| \frac{x^r - x_i^r}{h_i^r} \right\|^2 \right)$$



Ядро нужно выбирать несимметричным

Anisotropic Mean-shift

Выбор ядра

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{c}{(h_i^s)^p (h_i^r)^q} k^s \left(\left\| \frac{x^s - x_i^s}{h_i^s} \right\|^2 \right) k^r \left(\left\| \frac{x^r - x_i^r}{h_i^r} \right\|^2 \right)$$

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h^r (H_i^s)^q} k^s (g(x^s, x_i^s, H_i^s)) k^r \left(\left\| \frac{x^r - x_i^r}{h^r (H_i^s)} \right\|^2 \right)$$

$$g(x^s, x_i^s, H_i^s) = (x_i^s - x^s)^T H_i^s{}^{-1} (x_i^s - x^s)$$

$$k^s(z) = 1 \text{ if } |z| < 1, \text{ and } 0 \text{ otherwise}$$

$$k^r(z) = 1 - |z| \text{ if } |z| < 1 \text{ and } 0 \text{ otherwise}$$

Anisotropic Mean-shift

Выбор ядра

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h^r (H_i^s)^q} k^s(g(x^s, x_i^s, H_i^s)) k^r \left(\left\| \frac{x^r - x_i^r}{h^r (H_i^s)} \right\|^2 \right)$$

$$g(x^s, x_i^s, H_i^s) = (x_i^s - x^s)^T H_i^s^{-1} (x_i^s - x^s)$$

Размер цветового ядра h^r зависит от размеров матрицы H_i^s .

H_i^s определяется локальной структурой видео (скалярная / диагональная / симметричная).

Anisotropic Mean-shift

Выбор ядра

- Выбор окрестности

$$k^s(g(x, x_i, H_i^s)) < 1; \quad k^r \left(\left\| \frac{x - x_i}{h^r(H_i^s)} \right\|^2 \right) < 1$$

- Матрица H_i^s сначала выбирается скалярной, затем постепенно уточняется
- Уточнение H_i^s

$$\bar{H}_i^s = \lambda D A D^T$$

Anisotropic Mean-shift

Уточнение пространственной матрицы

Новая матрица H_i^s :

$$\bar{H}_i^s = \lambda D A D^T$$

λ – скаляр

D – матрица нормализованных собственных векторов

A – диагональная матрица собственных значений

$$\prod_{i=1}^p a_i = 1 \quad a_i \geq a_j, \text{ for } i < j$$

Anisotropic Mean-shift

Уточнение пространственной матрицы

Новая матрица H_i^s :

$$\bar{H}_i^s = \lambda D A D^T$$

λ – общий объем пространственно-временной окрестности

D – ориентация ядра в пространстве

A – относительная длина осей эллипса

Anisotropic Mean-shift

Уточнение пространственной матрицы

- Выделение тонких объектов

$$a'_i = \begin{cases} a_i^{3/2} & : a_i \leq 1 \\ \sqrt{a_i} & : a_i > 1 \end{cases}, i = 2, \dots, p$$

- Большие сегменты для статичных объектов

$$s_t = \alpha + (1 - \alpha) \prod_{i=1}^{p-1} d_1(i)^2$$

$d_1(i)$ – компонент наибольшего собственного вектора (x, y and t)

$$\alpha = 0.25$$

Пересчитываем $a'_i = a_i \cdot s_t, i = 2, \dots, p$ $\lambda' = \lambda \prod_{i=1}^p \left(\frac{a_i}{a'_i} \right)$

Anisotropic Mean-shift

Уточнение цветовых параметров

- Если ядро сильно вытянуто, цвет в нем меняется сильнее
- Сохранение устойчивости по цвету

$$h^r(H_i^s) \leftarrow \sqrt{\frac{\lambda'}{\lambda}} \cdot h^r(H_i^s)$$

Anisotropic Mean-shift

Общая схема (1/3)

1. Инициализация

- Перевод данных в пространство feature points (5D/6D)
- Задание начальных значений h_0^s и h_0^r
- $H_i^s = (h_0^s)^2 I$
- $h^r(H_i^s) = h_0^r$

Anisotropic Mean-shift

Общая схема (2/3)

2. Вычисление анизотропного ядра

- Соседние пиксели

$$k^s(g(x_i, x_j, H_i^s)) < 1; k^r \left(\left\| \frac{x_i - x_j}{h^r(H_i^s)} \right\|^2 \right) < 1$$

- Обновление матрицы

$$H_i^s \leftarrow \frac{\sum_{j=1}^n \left\| \frac{x_i^r - x_j^r}{h^r(H_i^s)} \right\|^2 (x_j^s - x_i^s)(x_j^s - x_i^s)^T}{\sum_{j=1}^n \left\| \frac{x_i^r - x_j^r}{h^r(H_i^s)} \right\|^2}$$

- Модификация матрицы H_i^s и $h^r(H_i^s)$ с помощью разложения

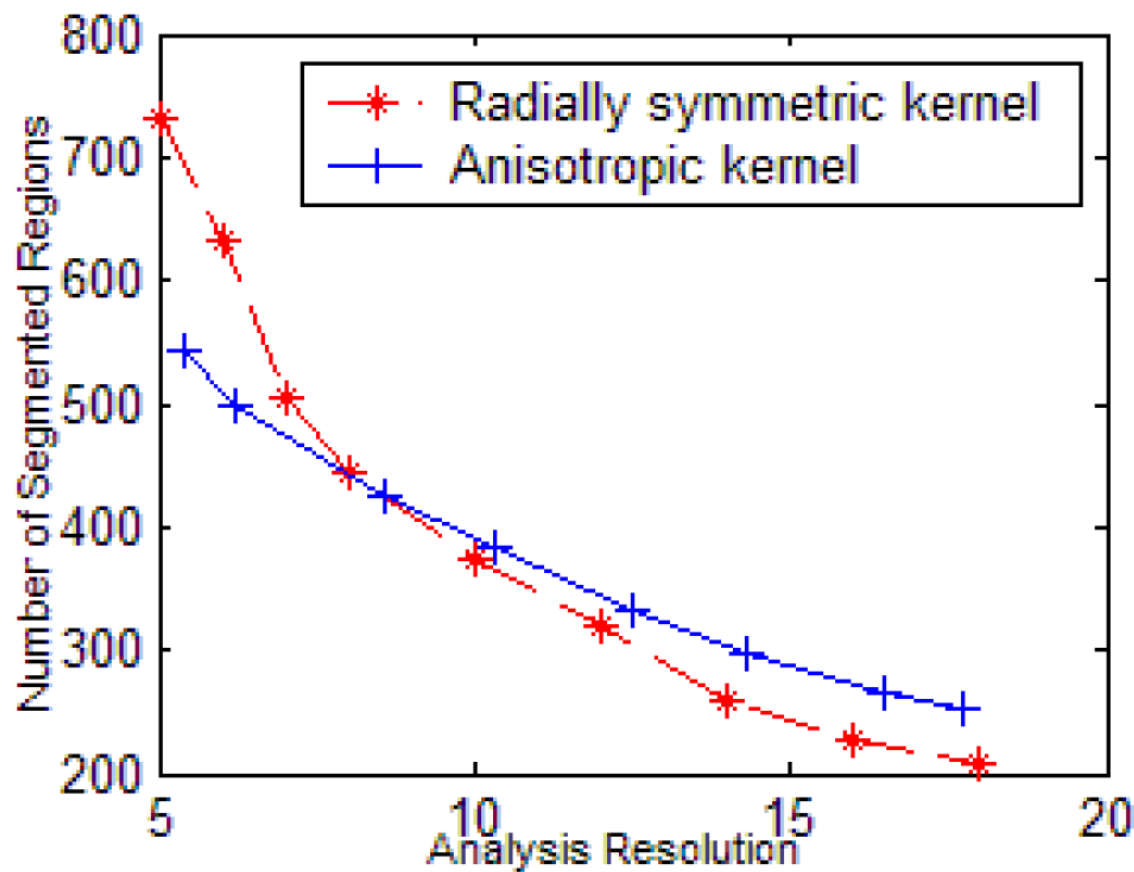
Anisotropic Mean-shift

Общая схема (3/3)

3. Провести несколько итераций вычисления анизотропного ядра
4. Запустить стандартную процедуру Mean-shift
5. Объединить пиксели с одинаковыми конечными точками траектории в сегменты
6. Удалить малые сегменты (опционально)

Anisotropic Mean-shift

Устойчивость относительно параметров



- Фиксирован начальный размер ядра по цвету
- Размер пространственного ядра варьируется

Результаты сегментации

Пример использования сегментации



Результаты сегментации

2D симметричная



Результаты сегментации

3D симметричная

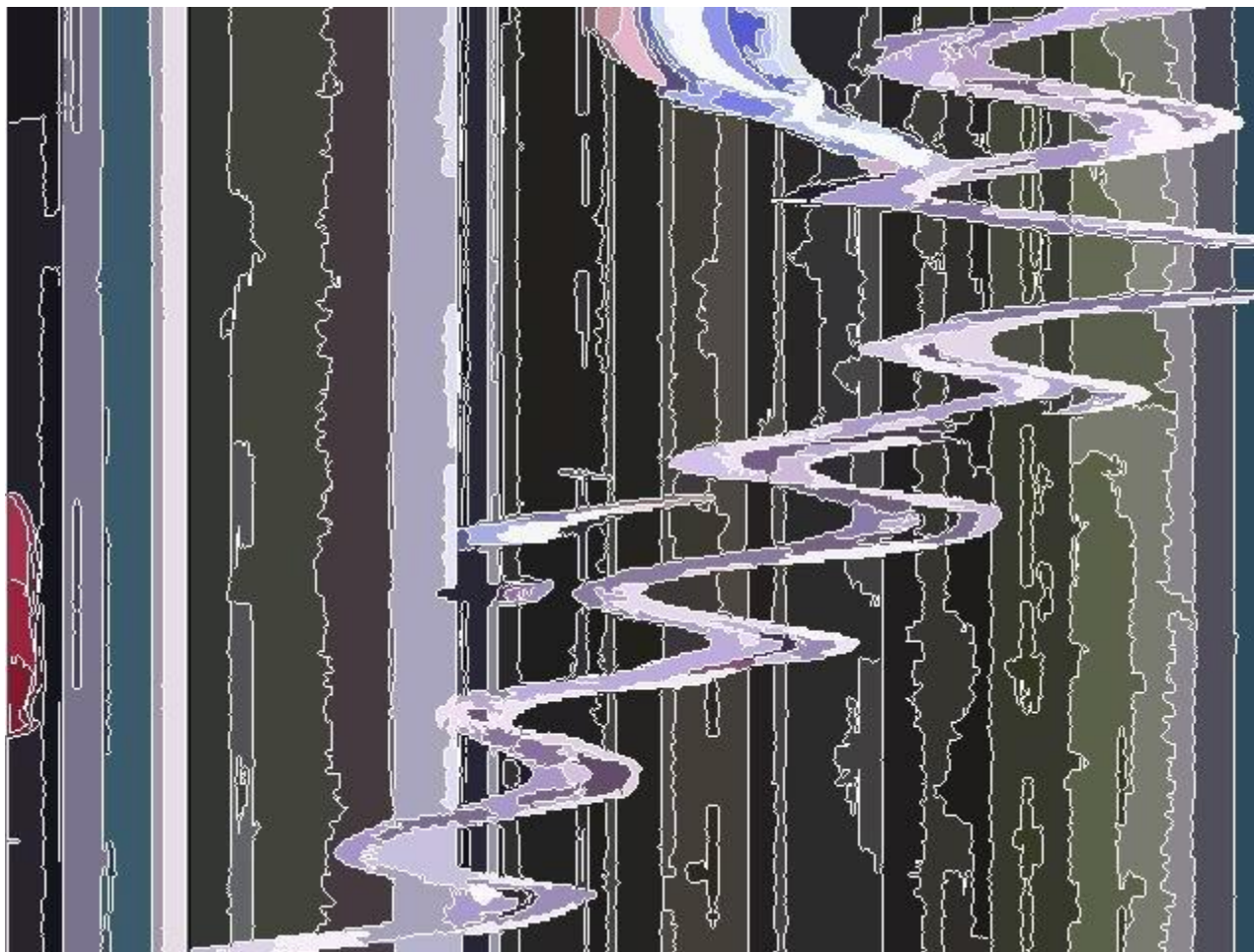


Результаты сегментации

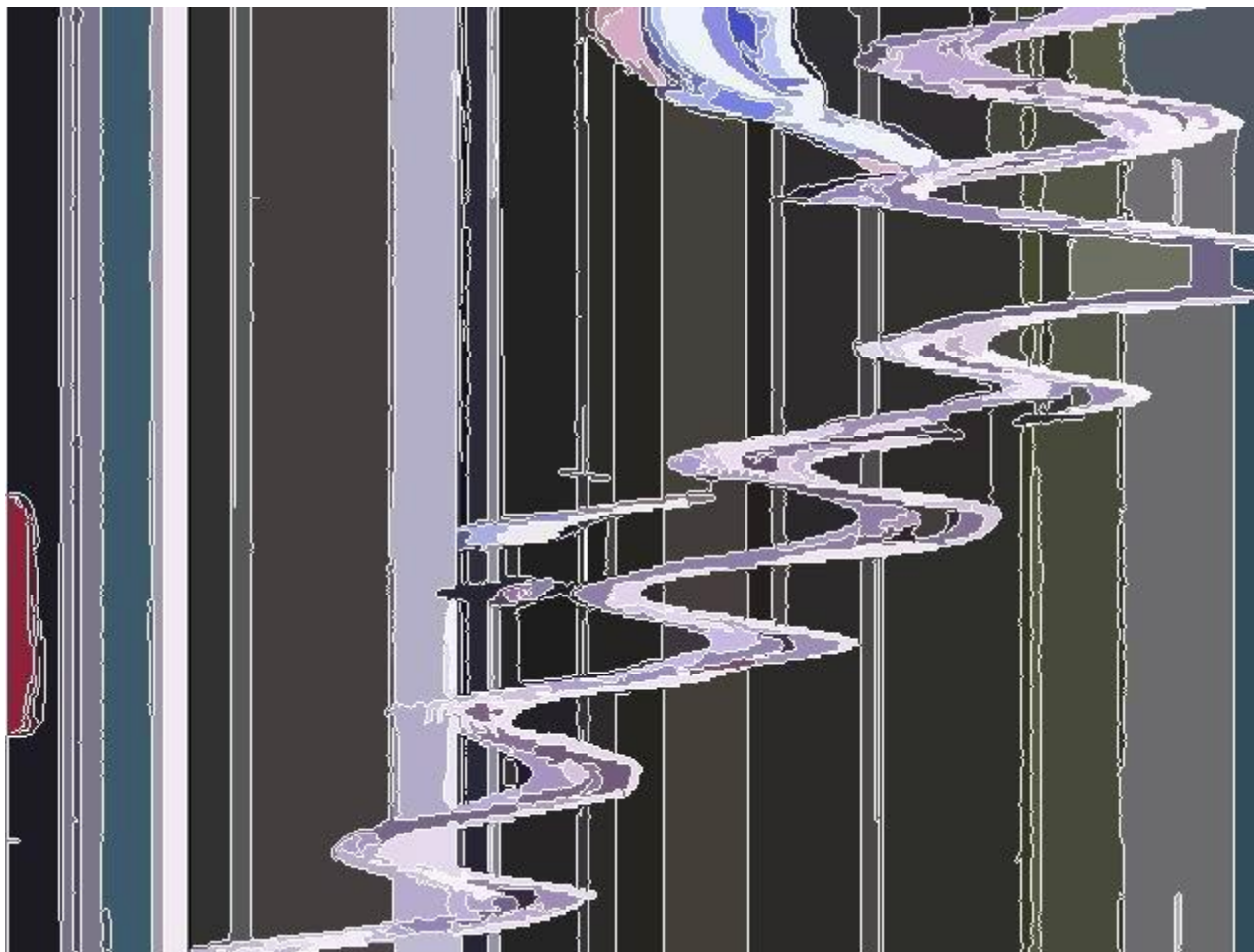
3D анизотропная



Временная согласованность 3D симметричная сегментация



Временная согласованность 3D анизотропная сегментация



Еще результаты

Исходный кадр



Еще результаты

Симметричная сегментация



Еще результаты

Анизотропная сегментация



Anisotropic Mean-shift

Выводы

- Достоинства
 - Простая реализация
 - Более устойчив к выбору входных параметров (относительно исходного алгоритма)
 - Возможно параллельное вычисление
- Недостатки
 - Вычислительная сложность
 - Нужно выбирать начальные параметры
 - Нужно сегментировать сразу набор кадров
 - Не используется оценка движения



Содержание

- Введение
- Анизотропный подход
- **Обновление Mean-shift**
- Ускорение Mean-shift
- Заключение

Обновление Mean-shift



- Базовый алгоритм – Mean-shift
- Сегментация в пространстве $V = [x, y, t, L, u, v]$
- Идея: разбить пространство на неперекрывающиеся регионы

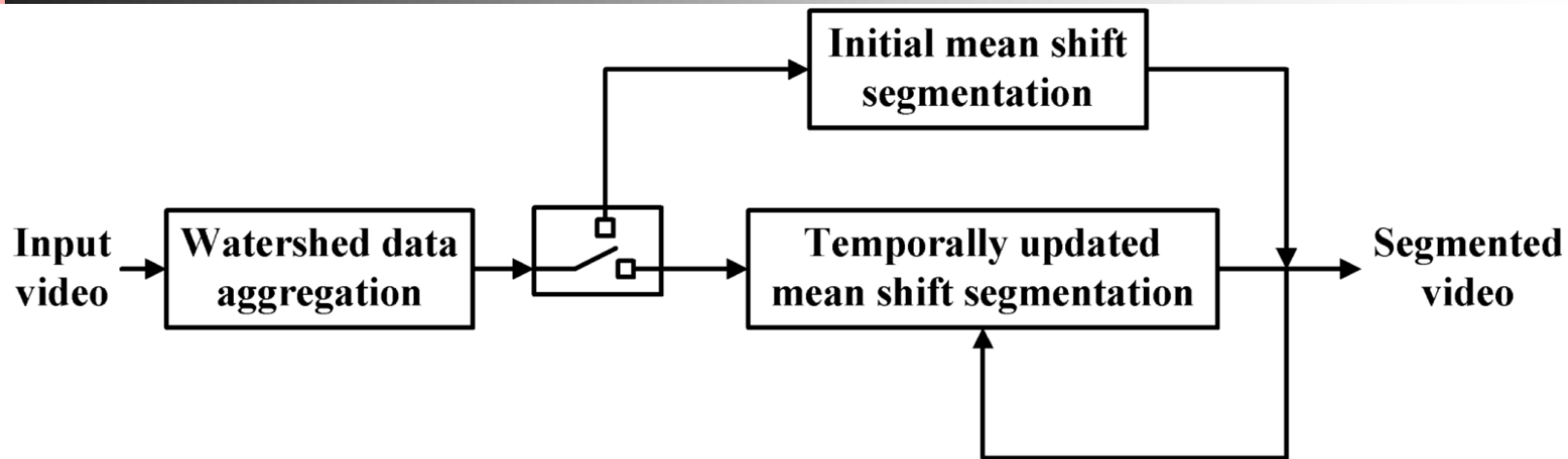
$$V_{seg(i,t)} = [x_{seg(i,t)}, y_{seg(i,t)}, t, L_{seg(i,t)}, u_{seg(i,t)}, v_{seg(i,t)}]$$

$$r_{seg(i,t)} = \frac{1}{n(seg(i,t))} \sum_{k \in seg(i,t)} r(k) \quad r \in \{L, u, v\}$$

$$s_{seg(i,t)} = \frac{1}{n(seg(i,t))} \sum_{k \in seg(i,t)} s(k) \quad s \in \{x, y\}$$

Обновление Mean-shift

Начальный этап



- Сегментация по методу водораздела (watershed) по L компоненте (цветовое пространство Luv)
- Начальная сегментация Mean-shift

$$S_{\text{init}} = \{V_{\text{seg}(i,t)} \mid 1 \leq i \leq n(\text{seg}(i,t)) \wedge 1 \leq t \leq N, \forall i, \forall t, i \in \mathcal{N}, t \in \mathcal{N}\}$$

Обновление Mean-shift

Выбор ядра

$$K(\mathbf{x}) = \frac{c}{h_t h_s^2 h_r^p} k\left(\left\|\frac{\mathbf{x}^t}{h_t}\right\|^2\right) k\left(\left\|\frac{\mathbf{x}^s}{h_s}\right\|^2\right) k\left(\left\|\frac{\mathbf{x}^r}{h_r}\right\|^2\right)$$

- Epanechnikov kernel

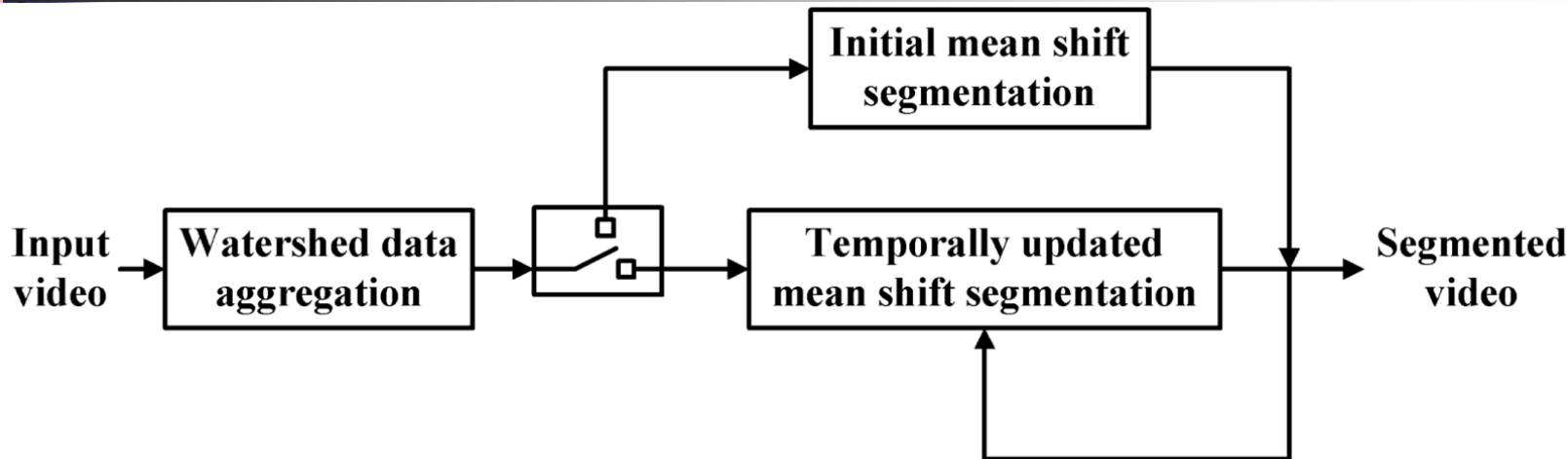
$$k_E(x) = \begin{cases} 1 - x & 0 \leq \|x\| \leq 1 \\ 0 & \|x\| > 1. \end{cases}$$

- Gaussian kernel

$$k_N(x) = e^{-\frac{1}{2}x} \quad x \geq 0.$$

Обновление Mean-shift

Основной режим



- Обновление пространства

$$S(m) = \{V_{\text{seg}(i,t)} | 1 \leq i \leq n(\text{seg}(i,t)) \wedge m + 1 - N \leq t \leq m, \forall i, \forall t, i \in \mathcal{N}, t \in \mathcal{N}\}$$

- Запускаем Mean-shift только для добавленных точек пространства (неточное соответствие)

Обновление Mean-shift

Скорость

Реализация: C++

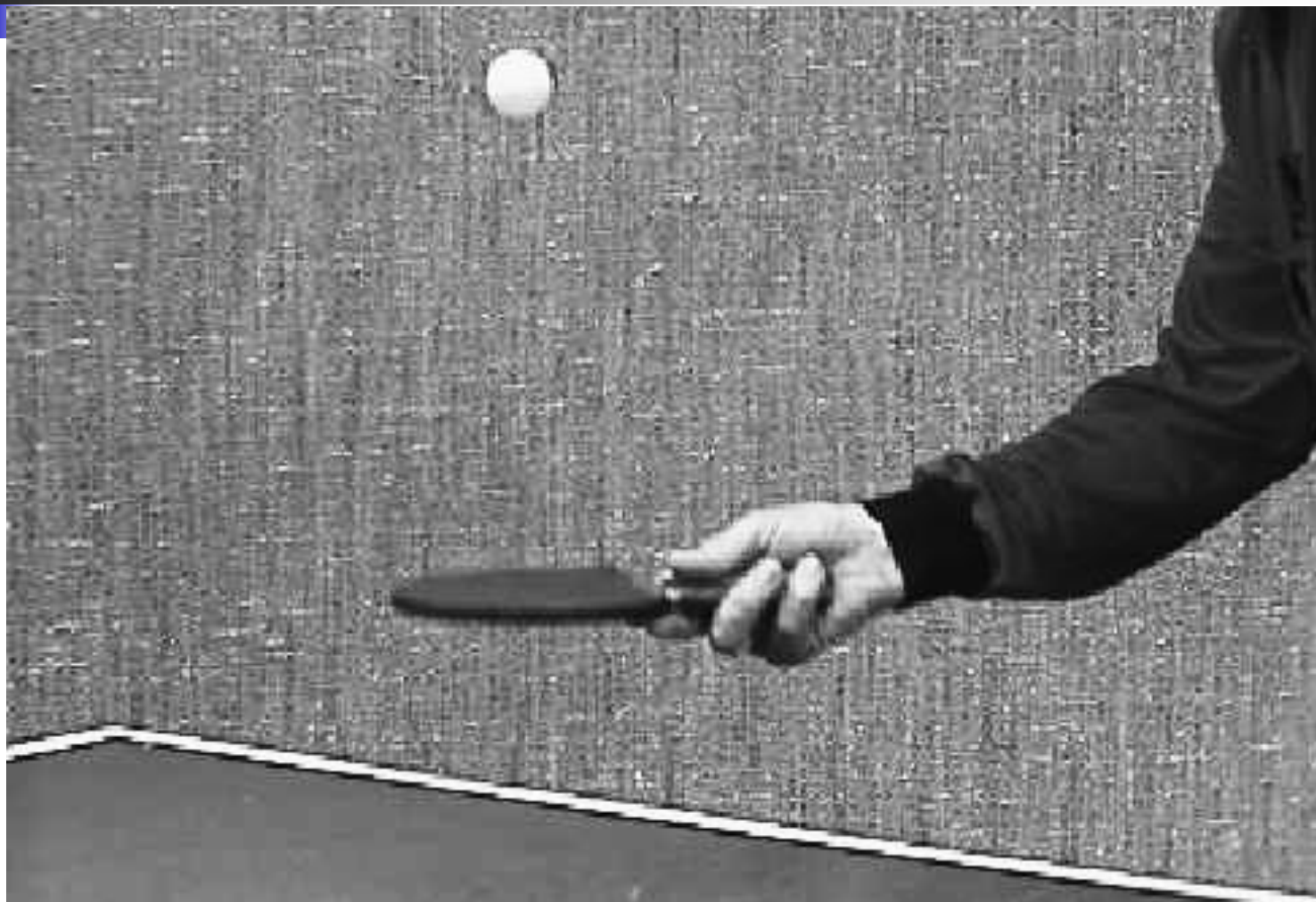
Процессор: AMD Athlon64 X2 Dual Core 4400+

Видео: Tennis 352 × 240

- Начальный этап 18.6 fps
- Основной этап 18.7 fps

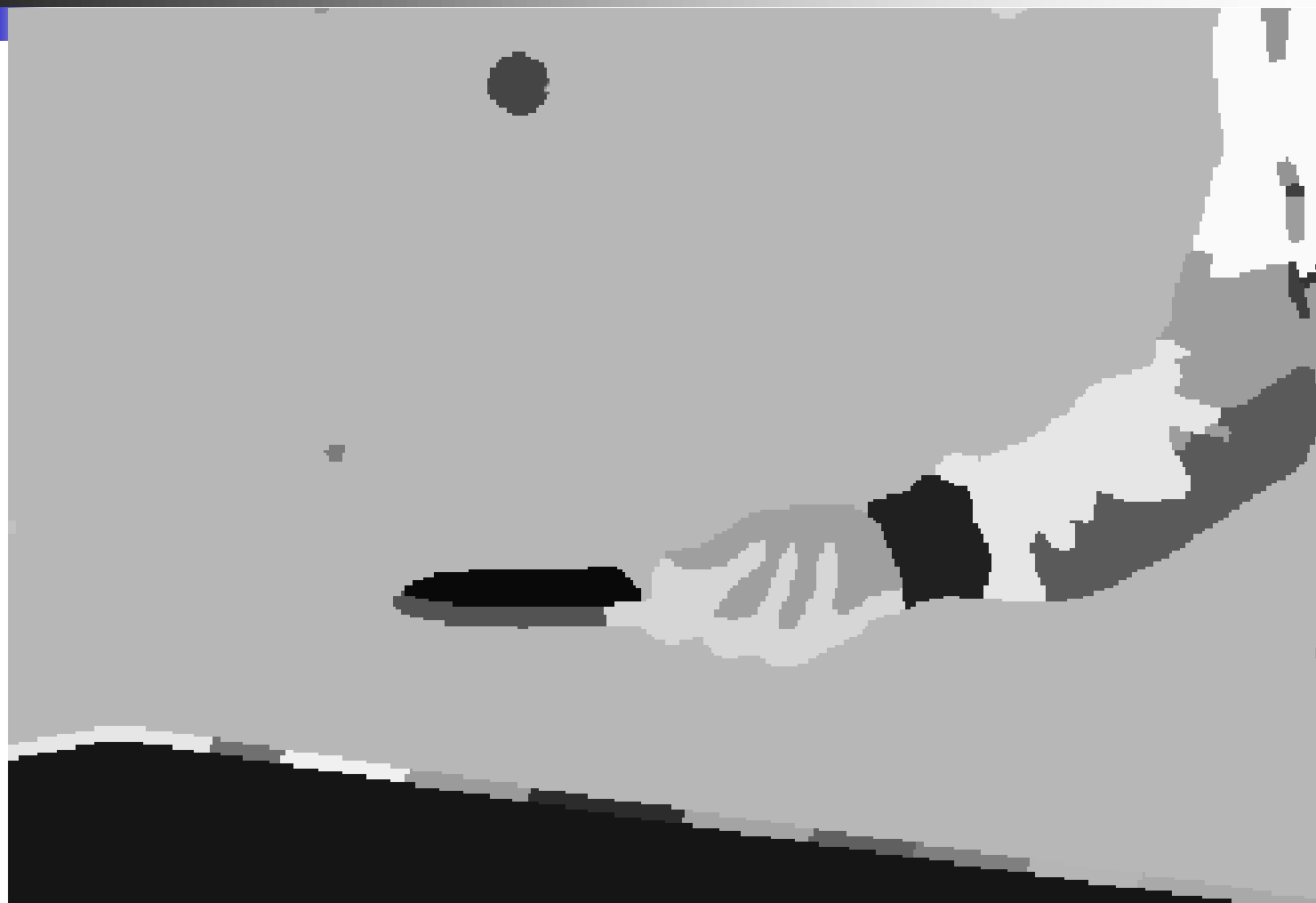
Обновление Mean-shift

Результаты (кадр 18)



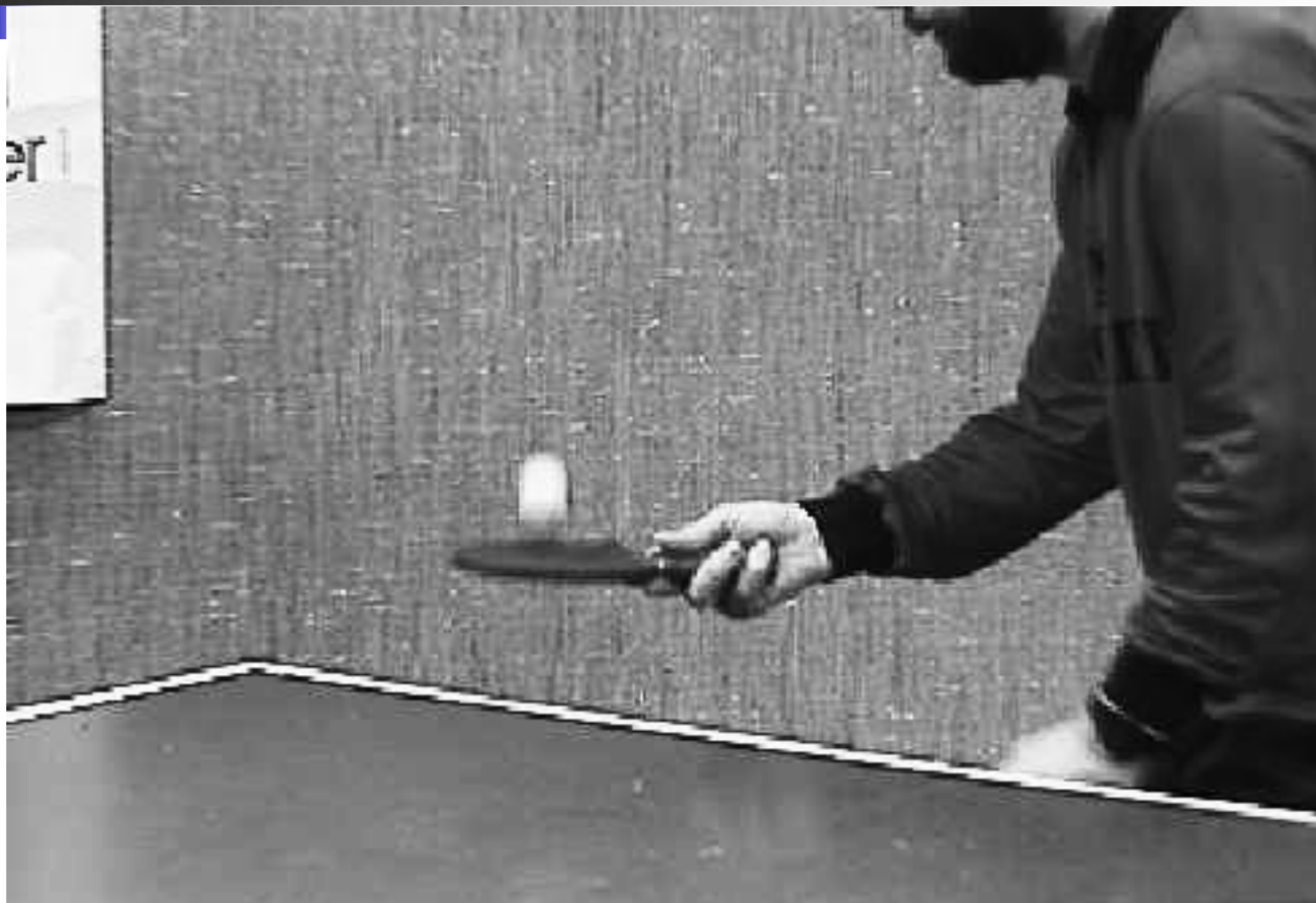
Обновление Mean-shift

Результаты (кадр 18)



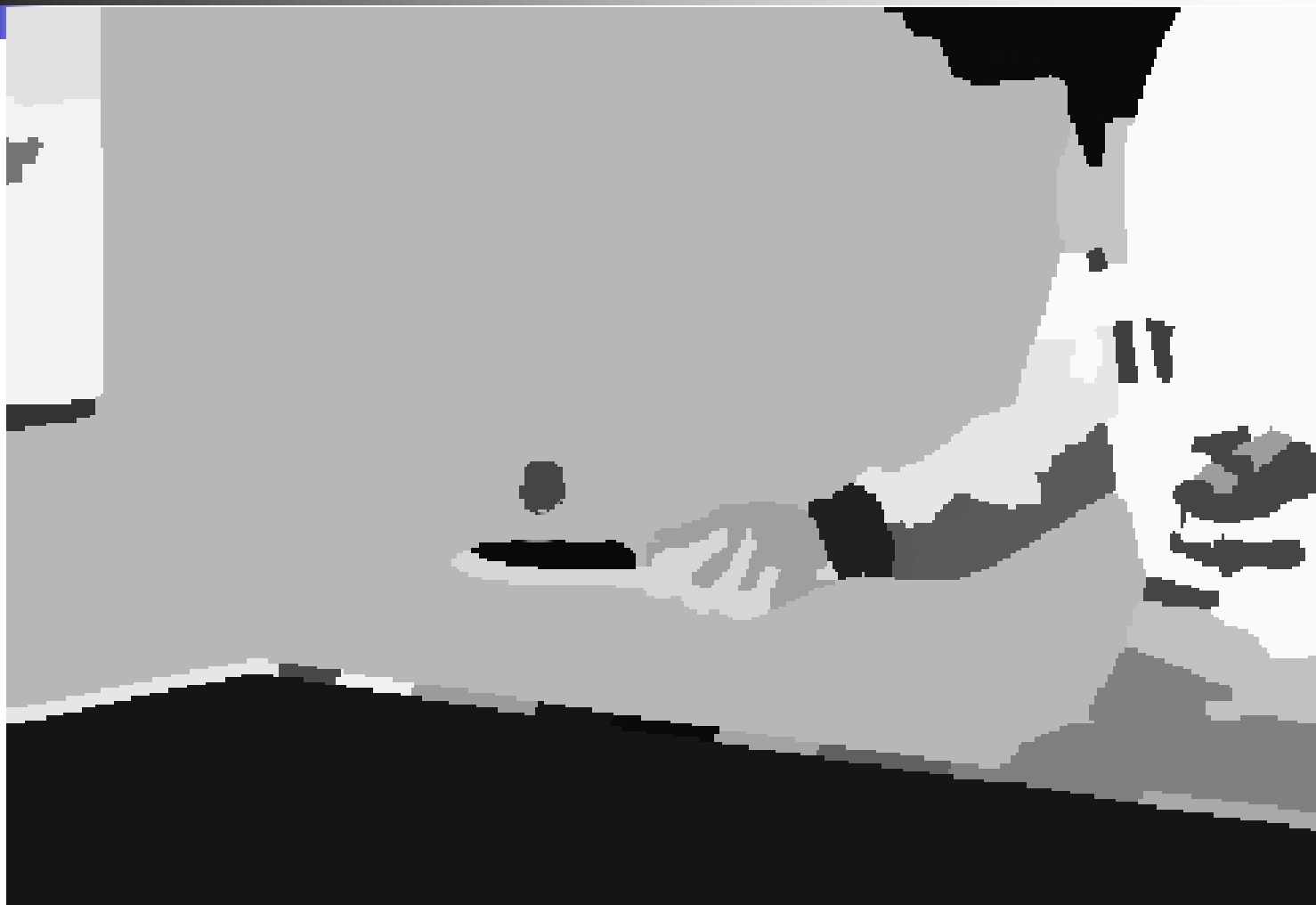
Обновление Mean-shift

Результаты (кадр 42)



Обновление Mean-shift

Результаты (кадр 42)



Обновление Mean-shift

Результаты (кадр 69)



Обновление Mean-shift

Результаты (кадр 69)



Обновление Mean-shift

Результаты (MissAmerica, кадр 35)



Обновление Mean-shift

Результаты (MissAmerica, кадр 35)



Обновление Mean-shift

Выводы

- Достоинства
 - Простота реализации
 - Скорость
- Недостатки
 - Непонятен выбор параметров
 - Невнятные результаты
 - Нет исходников
 - Не используется компенсация движения



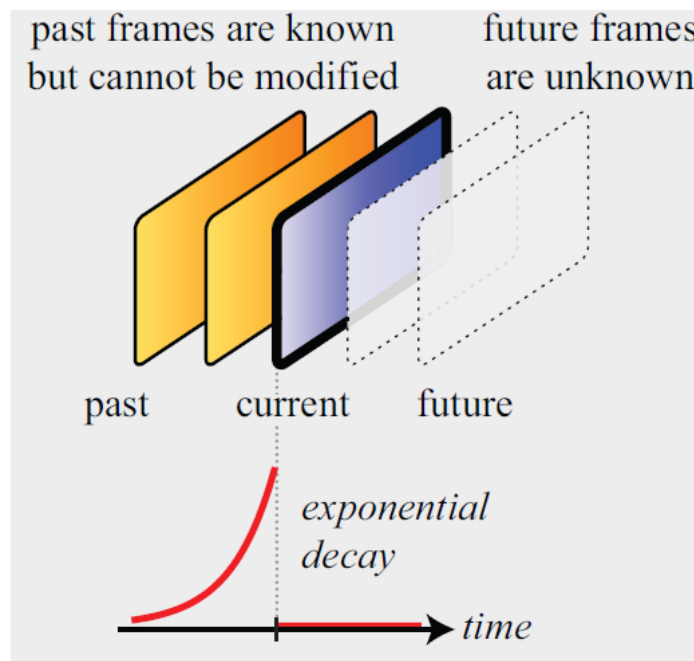
Содержание

- Введение
- Анизотропный подход
- Обновление Mean-shift
- **Ускорение Mean-shift**
- Заключение

Ускорение Mean-shift

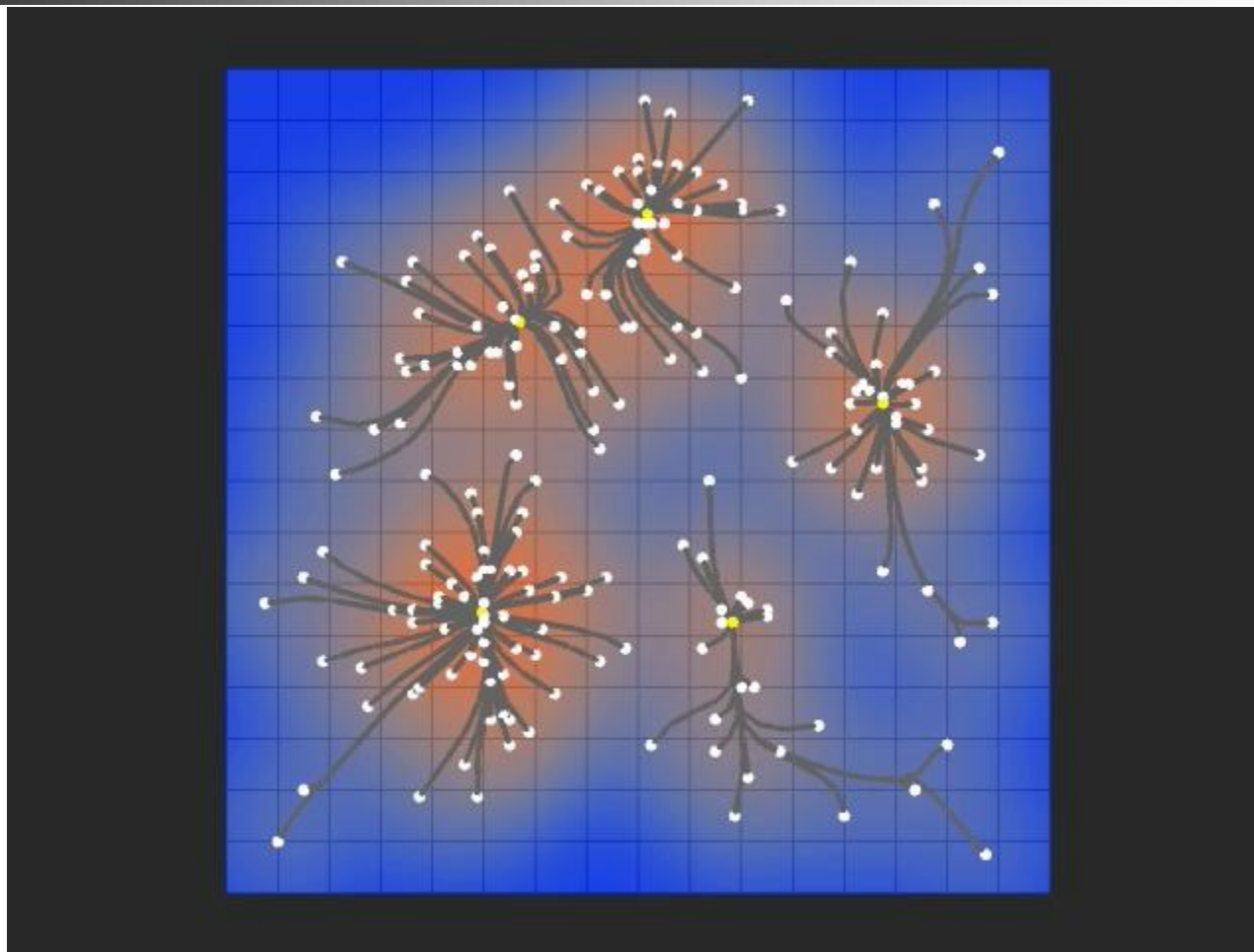
Введение

Обработка в реальном времени без задержки
(нет информации со следующих кадров)



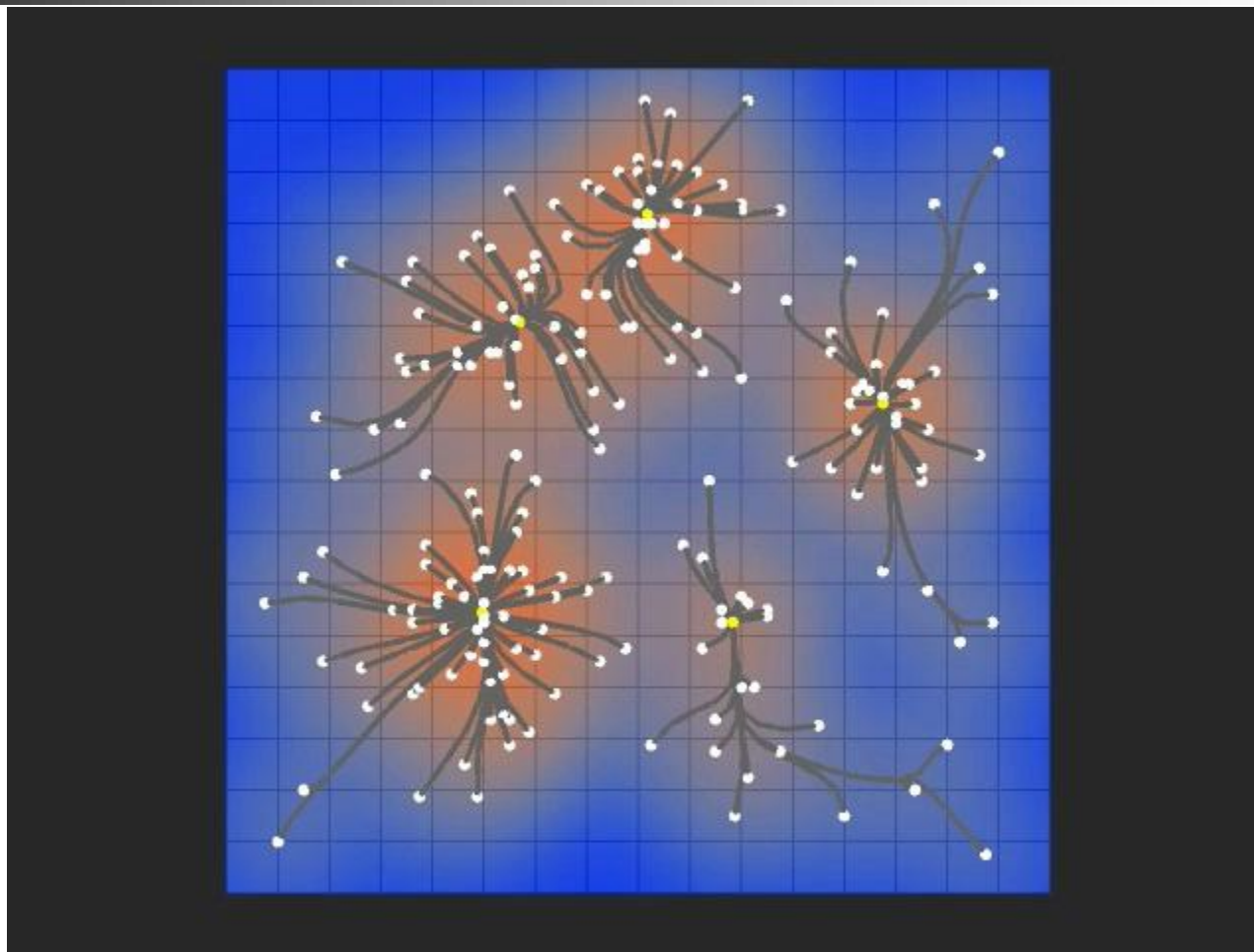
Ускорение Mean-shift

Переход к плотности распределения



Ускорение Mean-shift

Переход к плотности распределения



Ускорение Mean-shift

Обработка одного изображения

- Пользуемся тем, что ядро Гаусса сепарабельно
- Сортируем точки по плотности распределения (нужно найти максимумы)
- Рассматриваем в порядке убывания
- Рассматриваем наличие максимумов (меток) в окрестности
 - нет максимумов – создаем метку
 - одна – копируем
 - много – помечаем как *граничную* точку

Ускорение Mean-shift

Обработка видео

- Предыдущий кадр обработан
- Рассматриваем пространство (x, y, t, I)
 - x, y – координаты
 - t – время (0 или 1)
 - I – интенсивность

Ускорение Mean-shift

Распространение метки

- Находим локальные максимумы (как для изображений)
- Для каждого локального максимума на предыдущем кадре (метка L)
Проверяем, есть ли максимум на текущем кадре с большей плотностью. Если есть, то по скорейшему спуску переходим и распространяем метку L
- Если на текущем кадре максимум, а на предыдущем – максимум с большей плотностью, то далее не учитываем этот текущий
- Далее рассматриваем максимумы в окрестности (как для изображений)
- Ускорение – копировать метку для граничных точек из ближайшего сегмента с наибольшей плотностью

Обработка граничных точек

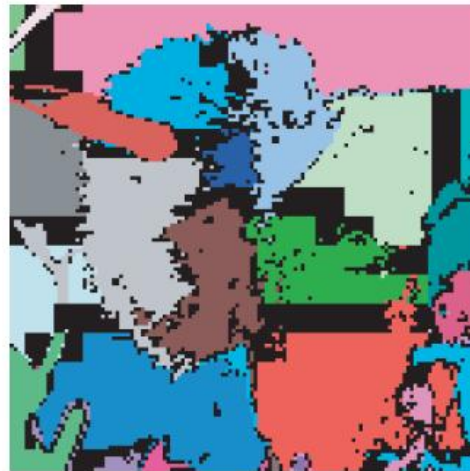
Простой подход

- Запускаем стандартный Mean-shift
- Копируем ту метку, к которой быстрее придём

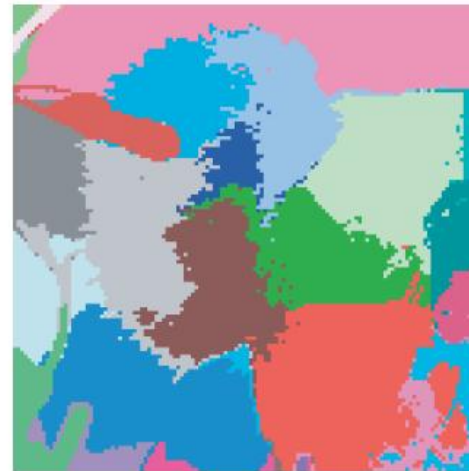
input



before refinement



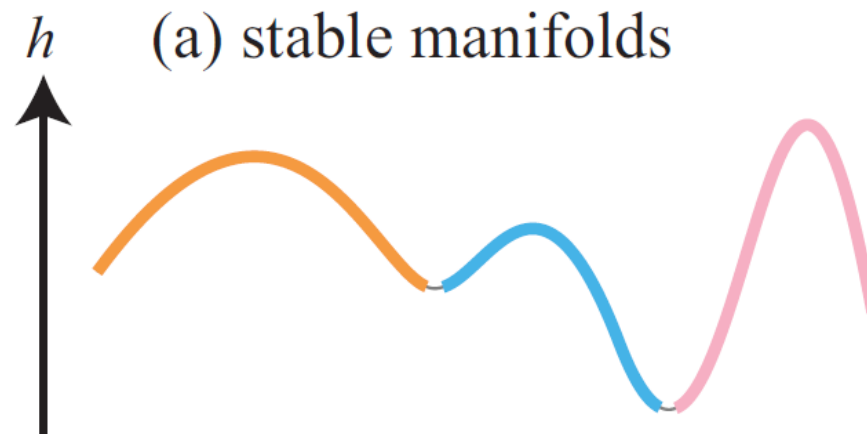
after refinement



Обработка граничных точек

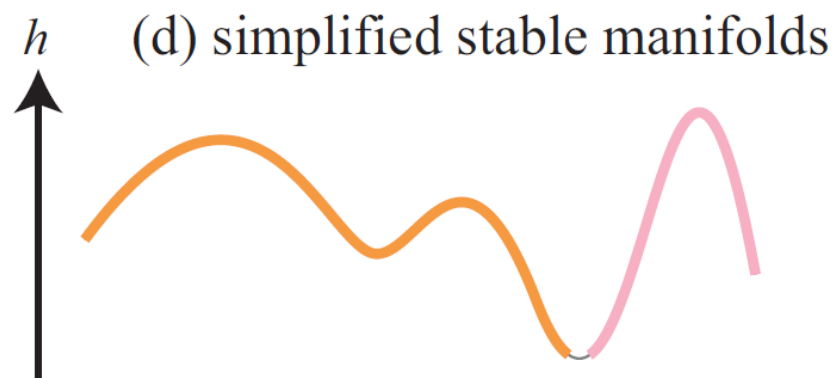
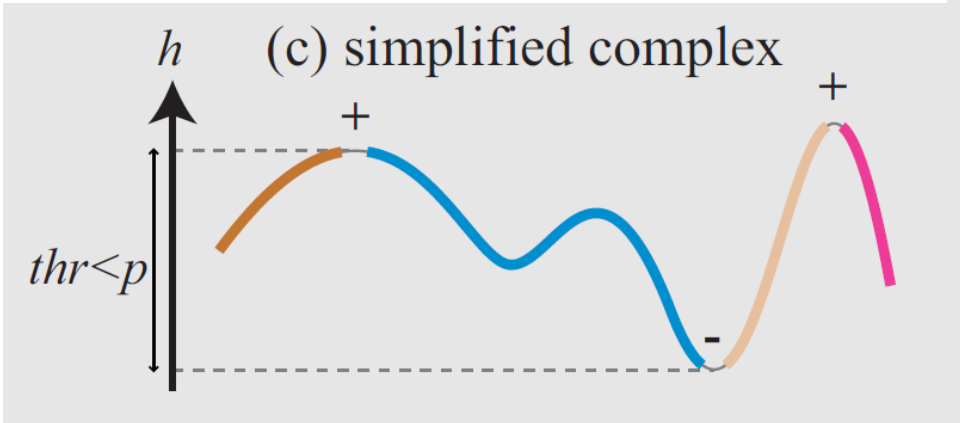
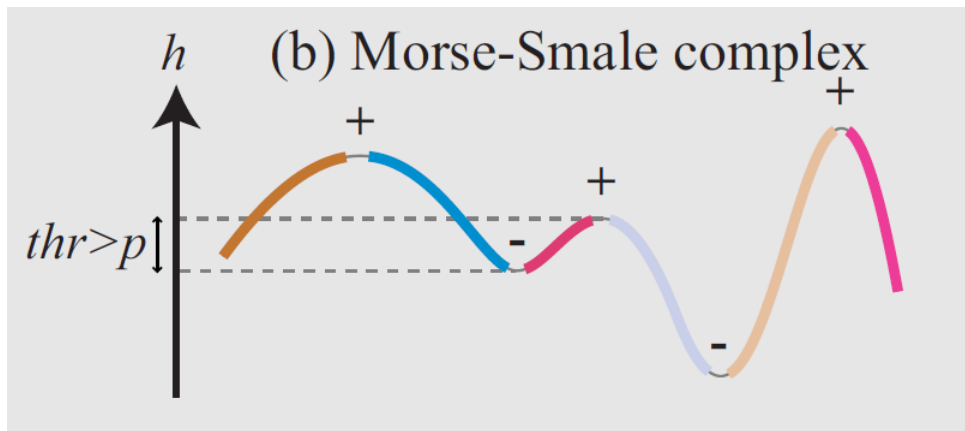
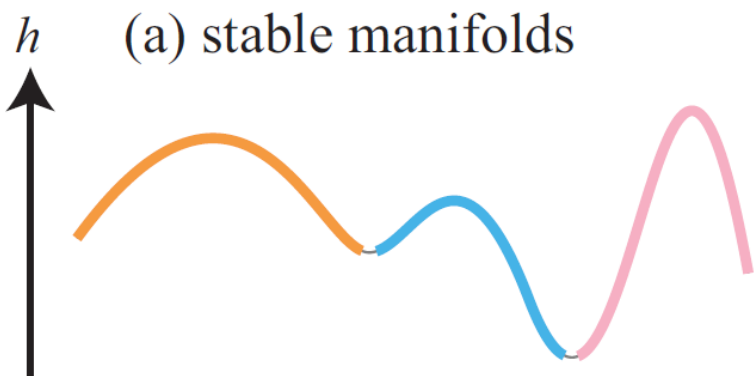
Быстрый подход

- m_1 и m_2 – максимумы $D(m_1) > D(m_2)$
- s_{12} – седловая точка
- Стойкость границы $p_b(m_1, m_2) = D(m_2) - D(s_{12})$



Обработка граничных точек

Слияние сегментов



Ускорение Mean-shift

Результаты



Results

Video Segmentation

Результаты

Исходный кадр



Результаты 4D пространство

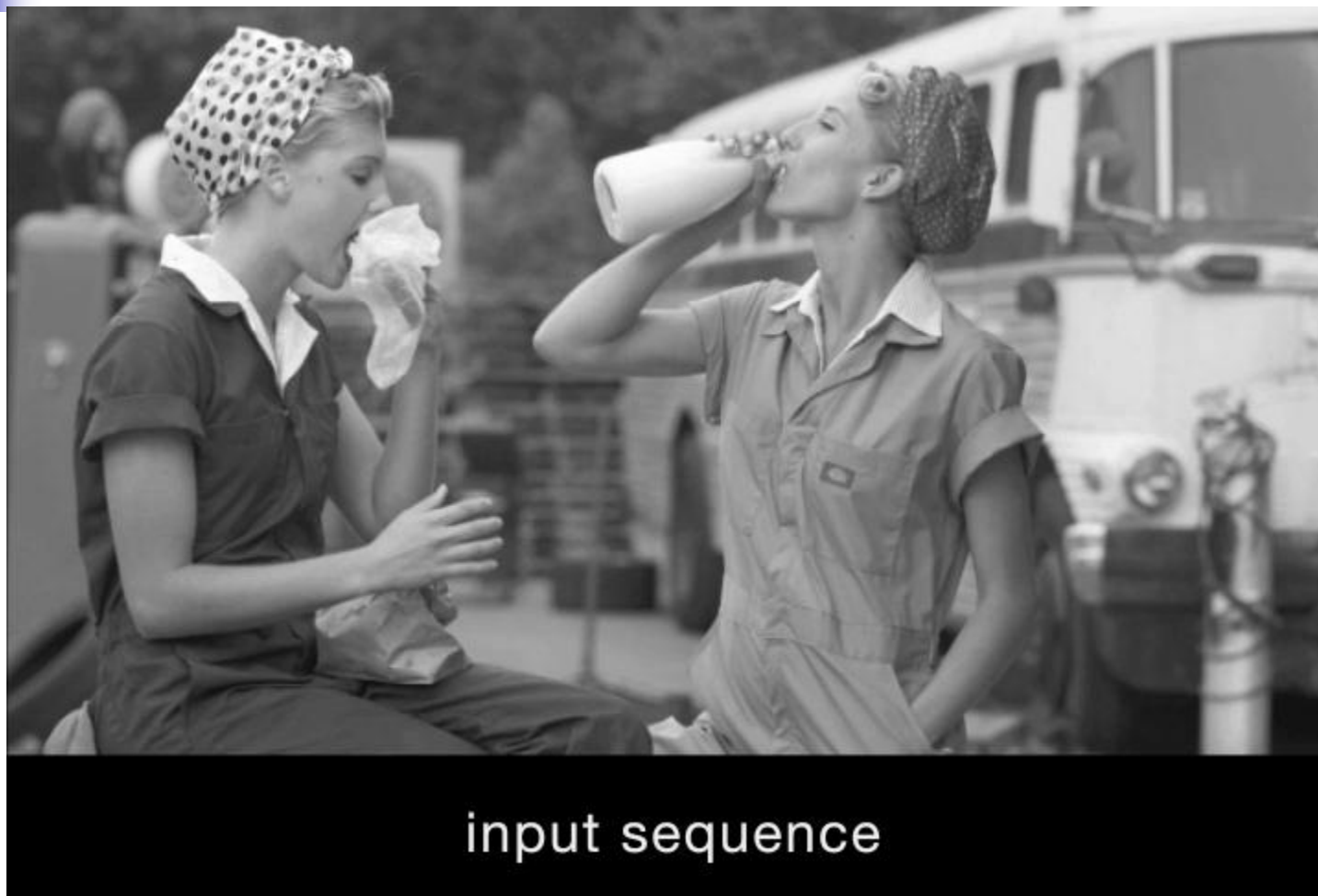


Результаты 5D пространство



Результаты

Исходный кадр

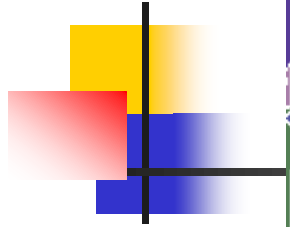


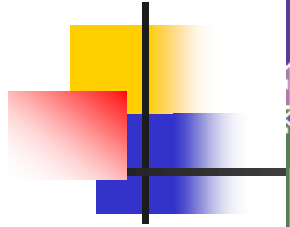
Результаты Сравнение



frame by frame
(no temporal coherence)

our approach
(temporally coherent)





Ускорение Mean-shift

Скорость и потребление памяти

Обработка видео 640x360 grayscale

Реализация: C++, GCC
4.0.1

Процессор: Intel Xeon
3GHz

	frame-by-frame	our approach
bilateral filter	29.1Hz (3.8MB)	28.4Hz (3.9MB)
mean shift	59.5Hz (1.1MB)	52.9Hz (1.3MB)
Perona-Malik	14.6Hz (8.8MB)	11.7Hz (9.7MB)

Обработка изображения 8 мегарixel

Размер ядра: 64
пикселя

Процессор: AMD
Opteron 2.6GHz, 1MB of
cache

	Off-line	On-the-fly <i>thr = 1</i>	On-the-fly <i>thr = 1, PCA to 4D</i>
Gaussian	6.08s	6.07s	0.59s
Sort	0.67s	0.67s	0.03s
Mode ext.	1.73s	1.81s	0.22s
Boundaries	15.58s	1.04s	1.02s
Other	5.23s	5.24s	1.15s
Total	29.29s	14.83s	3.01s

Ускорение Mean-shift

Выводы

- Достоинства
 - Простота
 - Скорость
 - Частично доступны исходники (C++)
 - Можно улучшить
- Недостатки
 - Нужно подбирать параметры
 - Не понятны некоторые моменты реализации
 - Используется только один предыдущий кадр



Содержание

- Введение
- Анизотропный подход
- Обновление Mean-shift
- Ускорение Mean-shift
- **Заключение**



Заключение

Основные идеи:

- Взять за основу ускоренный вариант, основанный на поиске максимумов плотности
- Подключить ME
- Разобраться с параметрами ядра
- Попробовать использовать анизотропное ядро
- Попробовать подключить пред-сегментацию

Литература

1. S. Paris, "*Edge-preserving Smoothing and Mean-shift Segmentation of Video Streams*", Processing of the European Conference on Computer Vision (ECCV), 2008.
2. N. Petrovic, L.J. Jovanov, A. Pižurica, W. Philips, "*Efficient Video Segmentation Using Temporally Updated Mean Shift Clustering*", in Proc. SPIE Conf. on Applications of Digital Image Processing XXXI, 2008, Aug 11-14, San Diego, CA, vol. 7073, pp 70731R-1 - 70731R-10.
3. D. Comaniciu, P. Meer, "*Mean Shift: A Robust Approach Toward Feature Space Analysis*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 24, Issue 5, May 2002.
4. J. Wang, B. Thiesson, Y. Xu, and M. Cohen, "*Image and Video Segmentation by Anisotropic Kernel Mean Shift*", Computer Vision - ECCV 2004.

Лаборатория компьютерной графики и мультимедиа



Видеогруппа — это:

- Выпускники в аспирантурах Англии, Франции, Швейцарии (в России в МГУ и ИПМ им. Келдыша)
- Выпускниками защищено 5 диссертаций
- Наиболее популярные в мире сравнения видеокодеков
- Более 3 миллионов скачанных фильтров обработки видео