

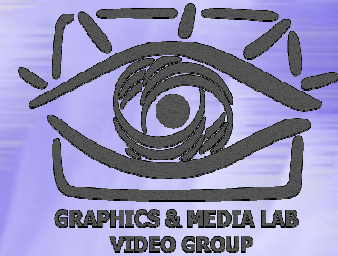
Введение в сжатие видео

Дмитрий Ватолин

*Московский Государственный Университет
CS MSU Graphics&Media Lab*

Version 2.4

Структура лекций

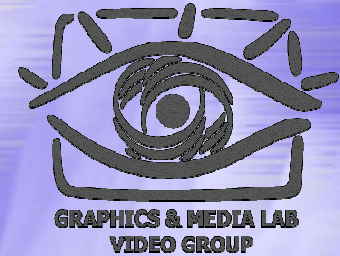


- ◆ **Раздел 1. Основные понятия в сжатии видео**
- ◆ Раздел 2. Построение простого видеокодека
- ◆ Раздел 3. Построение мощного видеокодека
- ◆ Раздел 4. Стандарты сжатия видео
- ◆ Раздел 5. Новое трехмерное видео

Причины сжатия видео



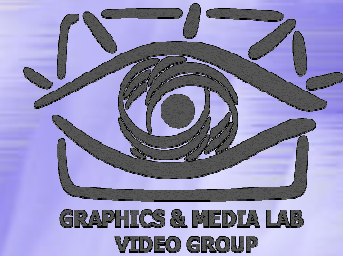
- ◆ Основные проблемы с видео:
 - Несжатые данные занимают очень много места
 - Каналы передачи и возможности хранения ограничены
- ◆ Пример: Видео 720x576 пикселей 25 кадров в секунду в системе RGB и прогрессивной развертке потребует потока данных примерно в 240 Мбит/сек (т.е. 1.8 Гб в минуту). На DVD-ROM диск размером 4.7Гб войдет всего 2.5 минуты. => Нужно сжатие в 35 раз для записи фильма.



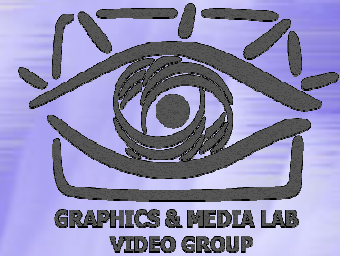
Характеристики ВИДЕОПОТОКА

- ◆ Видеопоток характеризуется *разрешением, частотой кадров (frames per second = fps) и системой представления цветов.*
- ◆ Например
 - PAL - 720x576, SECAM - 640x480, 25 fps
 - CIF - 352x288

Что используется при сжатии



- ◆ *Когерентность областей изображения* — малое локальное изменение цвета
- ◆ *Избыточность в цветовых плоскостях* — используется большая важность яркости для восприятия
- ◆ *Подобие между кадрами* — на скорости 25 кадров в секунду соседние кадры, как правило, изменяются незначительно



Что используется при сжатии

Используется избыточность:

- ◆ **Пространственная** (\Rightarrow используется DCT или Wavelet преобразования)
- ◆ **Временная** (между кадрами, \Rightarrow сжимается межкадровая разница)
- ◆ **Цветового пространства** (\Rightarrow RGB переводится в YUV и цветовые компоненты прореживаются)

Пространственная и временная избыточность

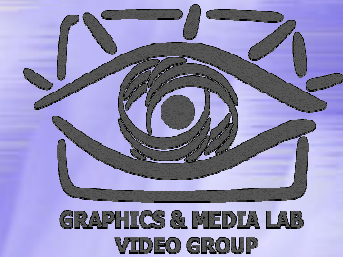


Соседние кадры
фильма
(Терминатор-2)

**Пространственная
избыточность** – цвет
большинства соседних
точек одинаков.

**Временная
избыточность** –
кадры весьма похожи

Межкадровая разница



Именно такие кадры (с учетом поправки на компенсацию движения) и сжимает кодек. Их больше 99% в потоке.
(Амплитуды – малы, изображение практически однородно)

Качество видео



- ◆ **Не существует метода оценки качества кадра, полностью адекватного человеческому восприятию**
- ◆ **Не существует метода оценки пропущенных кадров, полностью адекватного человеческому восприятию**

Следствие: Можно декларировать любую степень сжатия в маркетинговых материалах.

PSNR



Базовые метрики –
Y-PSNR, U-PSNR, V-PSNR

$$d(x,y) = 10 \cdot \log_{10} \frac{255^2 \cdot n^2}{\sum_{i=1, j=1}^{n, n} (x_{ij} - y_{ij})^2}$$

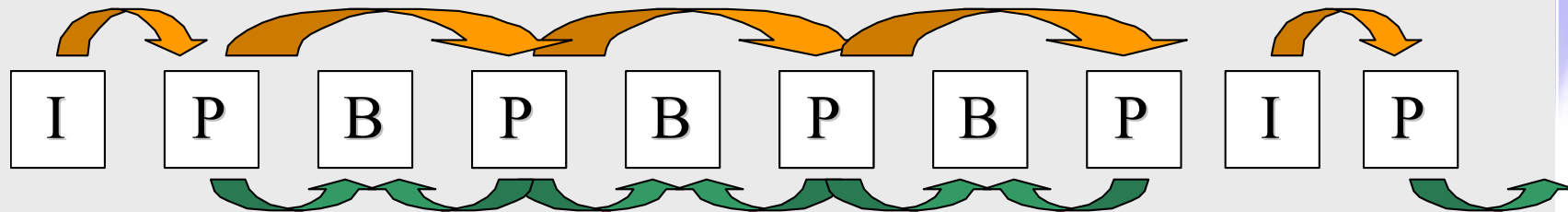
Хорошо работают только на **высоком**
качестве.

Структура лекций



- ◆ Раздел 1. Основные понятия в сжатии видео
- ◆ **Раздел 2. Построение простого видеокодека**
- ◆ Раздел 3. Построение мощного видеокодека
- ◆ Раздел 4. Стандарты сжатия видео
- ◆ Раздел 5. Новое трехмерное видео

Типы кадров в потоке



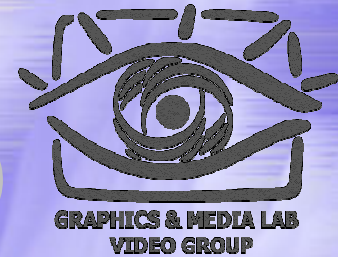
I-кадры — независимо сжатые (I-Intrapictures),
P-кадры — сжатые с использованием ссылки на одно изображение (P-Predicted),
B-кадры — сжатые с использованием ссылки на два изображения (B-Bidirection),

Компенсация движения



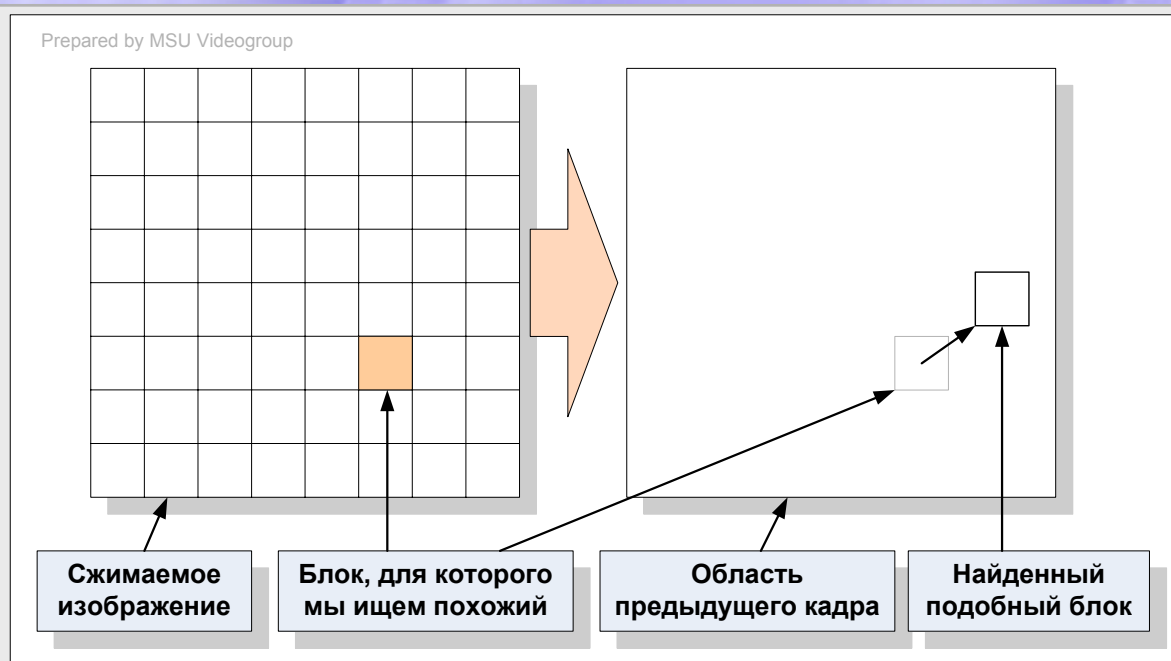
- ◆ Простая межкадровая разница работает плохо при сильном движении в кадре
- ◆ Алгоритмы компенсации движения отслеживают движение объектов в кадре
 - Уменьшение межкадровой разницы (увеличение ее степени сжатия)
 - Необходимость сохранения информации о движении в кадре
 - Существенно большее время, необходимое для сжатия

Компенсация движения (2)



- ◆ Идеальный алгоритм: **выделение в кадре объектов** и компактное описание их движений. Проблема: огромные объемы вычислений и весьма сложные алгоритмы.
- ◆ Реально используются квадратные блоки, с размером, кратным 8 и достаточно простая организация блоков.

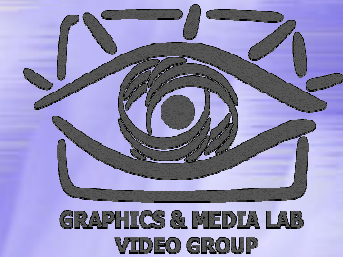
Компенсация движения (3)



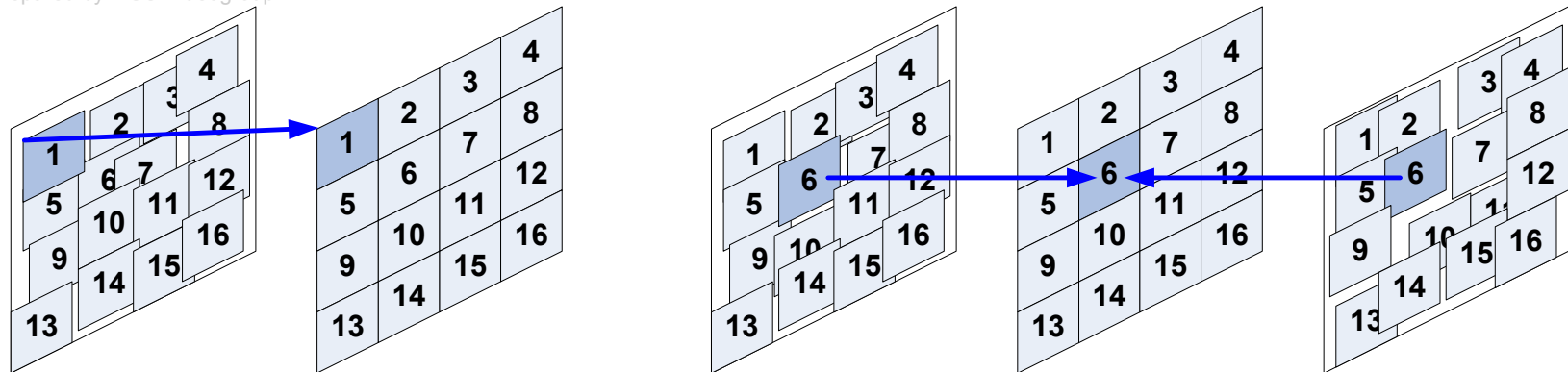
Для каждого блока в кадре мы находим похожий в предыдущем кадре в некоторой окрестности положения блока.

Если достаточно похожий блок в предыдущем кадре не найден – блок сжимается независимо (*Intra-Blocks*).

Движение для В-кадра



Prepared by MSU Videogroup



Предыдущий кадр Р-кадр

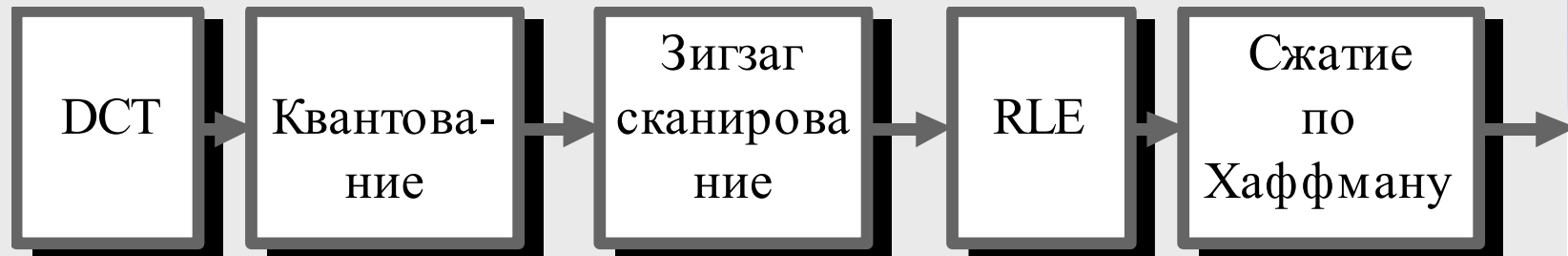
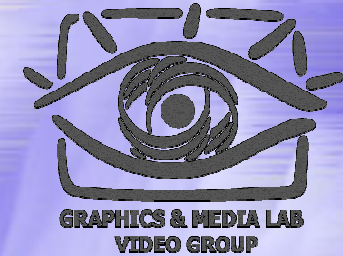
Предыдущий кадр В-кадр

Следующий кадр

Для В-кадров у нас появляется возможность выбирать как наиболее близкий блок из любого кадра, так и интерполировать блоки из двух кадров.

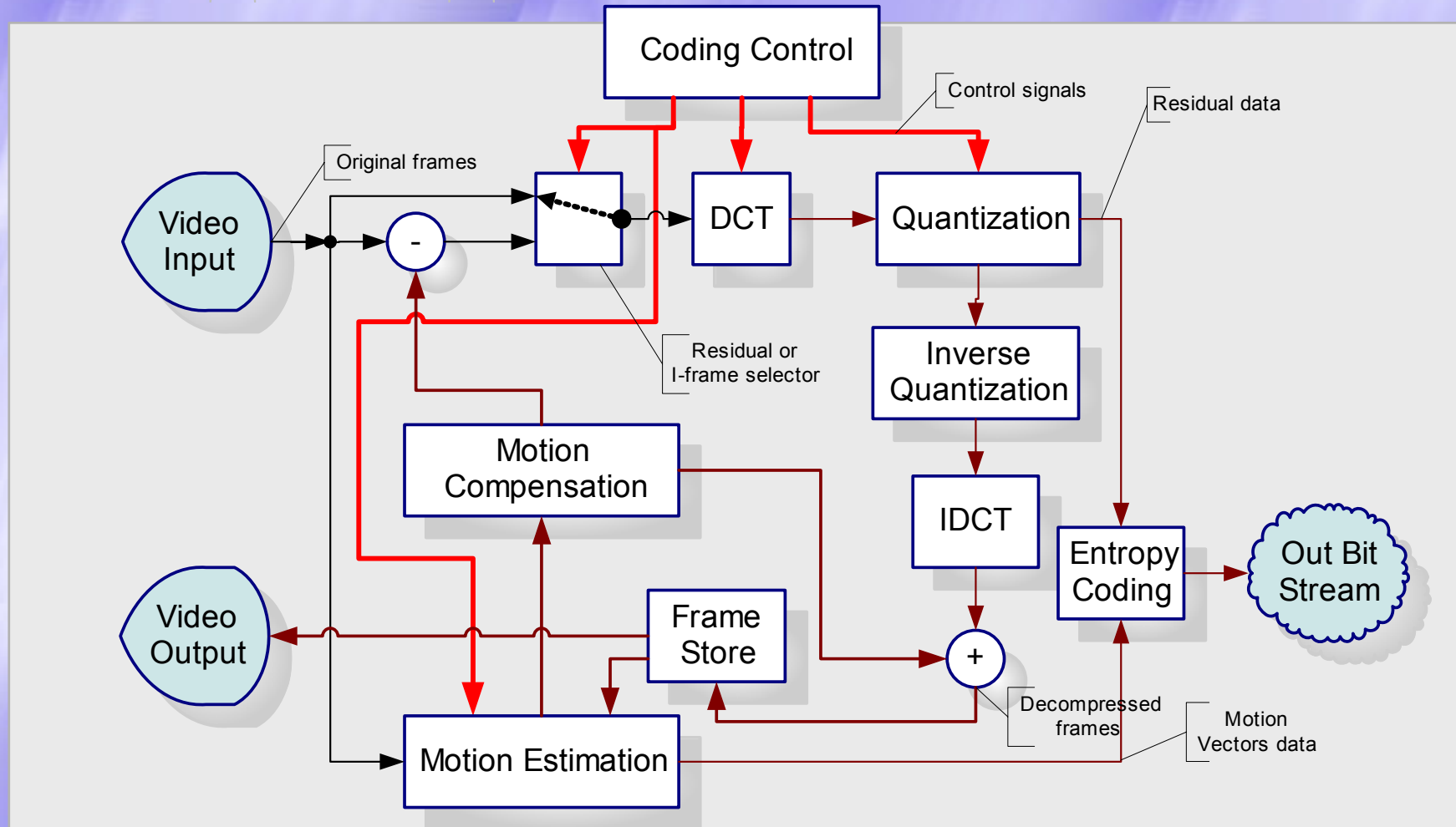


Сжатие межкадровой разности



Классическая схема сжатия межкадровой разности очень похожа на сжатие JPEG: блоки 8x8 сжимаются помощью дискретного косинусного преобразования

Схема простого ВИДЕОКОДЕКА

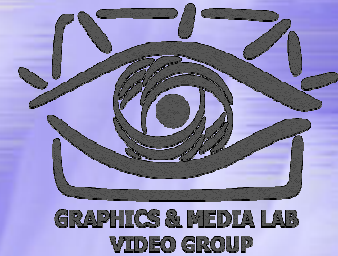


Структура лекций



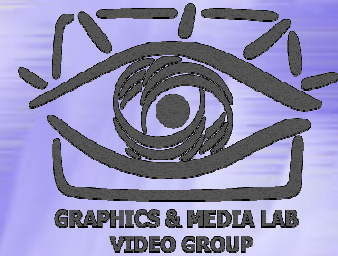
- ◆ Раздел 1. Основные понятия в сжатии видео
- ◆ Раздел 2. Построение простого видеокодека
- ◆ **Раздел 3. Построение мощного видеокодека**
- ◆ Раздел 4. Стандарты сжатия видео
- ◆ Раздел 5. Новое трехмерное видео

Пути повышения степени сжатия



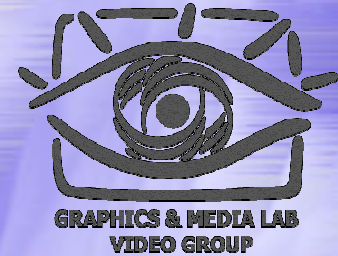
- ◆ **Изменение алгоритма сжатия I-кадров (DCT >> Wavelet)**
- ◆ **Изменение алгоритма сжатия без потерь (Huffman >> Arithmetic)**
- ◆ **Постфильтрация получающихся кадров (Deblocking, deringing и т.п.)**
- ◆ **Интра-фильтрация изображения (Loop-filtering)**
- ◆ **Улучшение качества более значимых объектов (face detection)**

Пути повышения степени сжатия (2)



- ◆ **Применение обработки коэффициентов**
(фильтрация DCT коэффициентов)
- ◆ **Предварительная обработка фильма**
(префильтрация, подавление шумов, дрожания, артефактов черезстрочной развертки)
- ◆ **Улучшение алгоритмов масштабирования видео** (билинейный >> бикубический)
- ◆ **Применение контекстного моделирования**
(SABAS)
- ◆ **Применение оценочного сжатия**

Пути повышения степени сжатия (3)



◆ Изменение алгоритмов компенсации движения

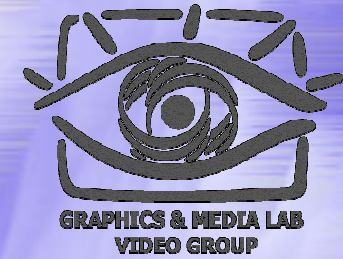
- Учет реальной степени сжатия получаемых блоков
- Изменение формы разбиения блоков
- Минимальный размер 4x4
- $\frac{1}{4}$ -пиксельная точность сдвига блока
- Несколько базовых изображений (больше 2)

Wavelet



- ◆ **Wavelet** – одна из самых современных технологий сжатия (используется, например в JPEG-2000)
- ◆ Визуальное преимущество за счет того, что пропадает блочность.

Wavelet



Сравнение с MPEG-4-like compressors.

Артефакты фильмов



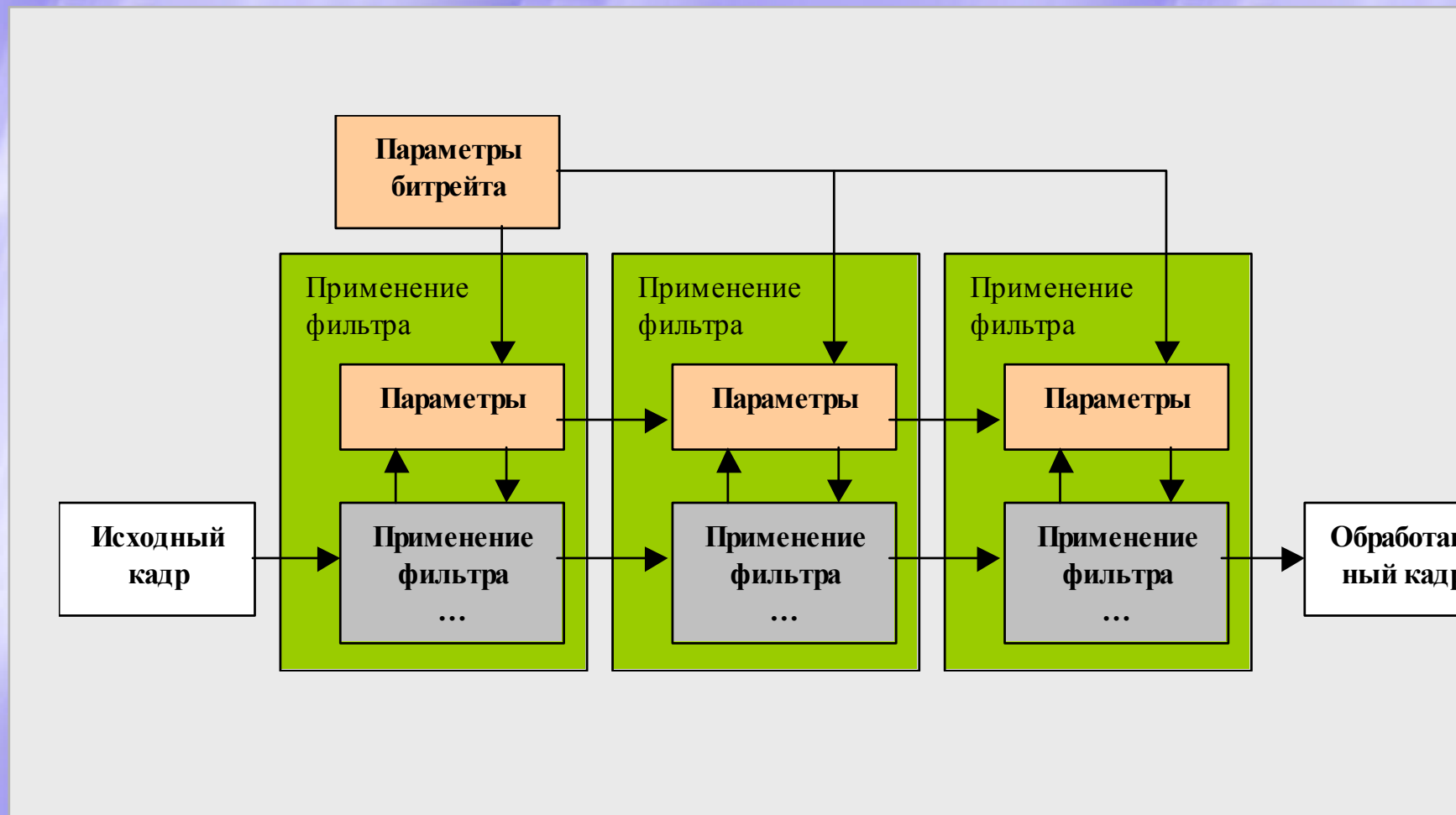
- ◆ Шум камеры (характерен для цифровых и веб-камер) – увеличивает размер
- ◆ Мусор, царапины, дрожжание кадра (для материала, перегоняемого с пленки) – увеличивает размер
- ◆ Блочность (для DVD, USB-1.0 веб-камер) – увеличивает размер, неприятна для глаз

Префильтрация



- ◆ Позволяет значительно поднять качество за счет подавления шумов
- ◆ Позволяет подавить шумы во временной и пространственной области
- ◆ Делает изображение более удобным для дальнейшей обработки

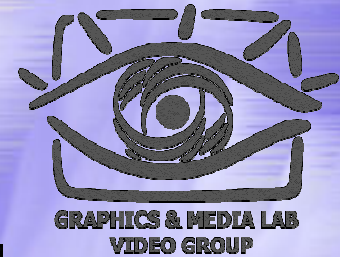
Схема работы цепочки фильтров



Префильтрация - было



Префильтрация - стало

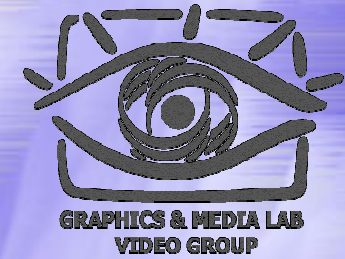


Префилтрация: межкадровая разница

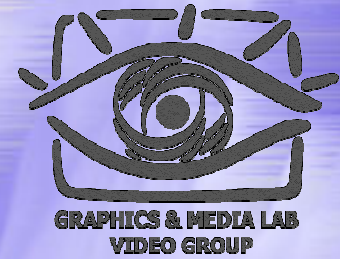


Уменьшение шума в простой межкадровой
разнице (без компенсации движения).

Интра-филтрация – было



Интра-филтрация – стало



Постфильтрация - деблокинг



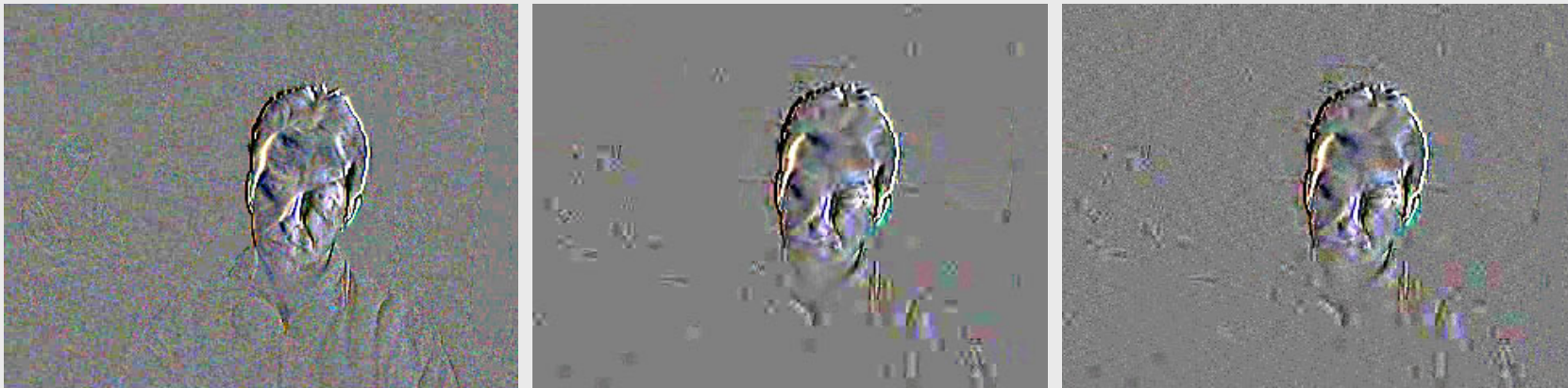
Пример визуального уменьшения блочности.

Постфильтрация - дерингинг



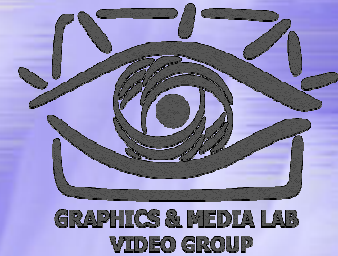
Пример визуального уменьшения рингинга.

Постфильтрация - перемешивание



Межкадровая разница: Оригинал, «обычный»
кодк, кадр с перемешиванием

ME: Компенсация движения (Motion Estimation)



Необходимо найти, как и куда сдвинулись блоки (объекты) в видео.

Для каждого блока мы рассматриваем возможный вектор сдвига, и для каждого вектора рассчитываем расстояние между блоками.

Задача: Если интерпретировать расстояние как высоту, то необходимо найти наиболее глубокую точку минимальным количеством замеров.

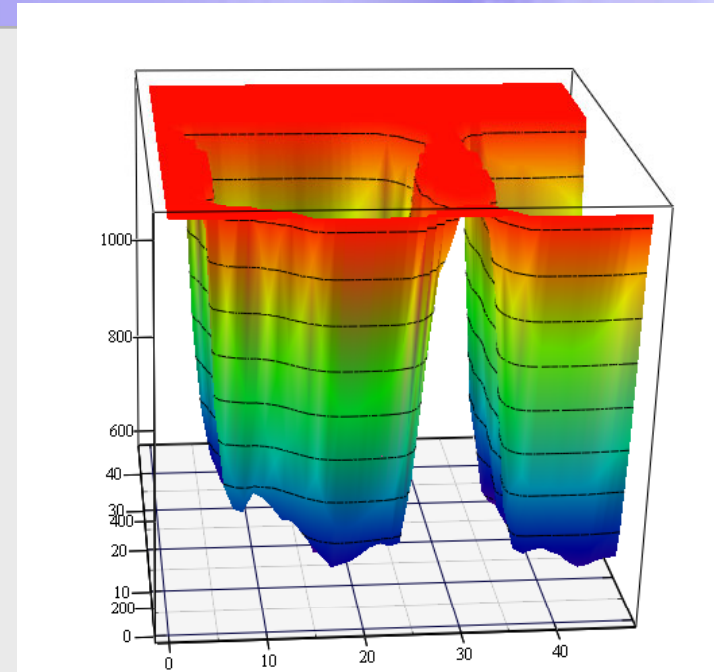
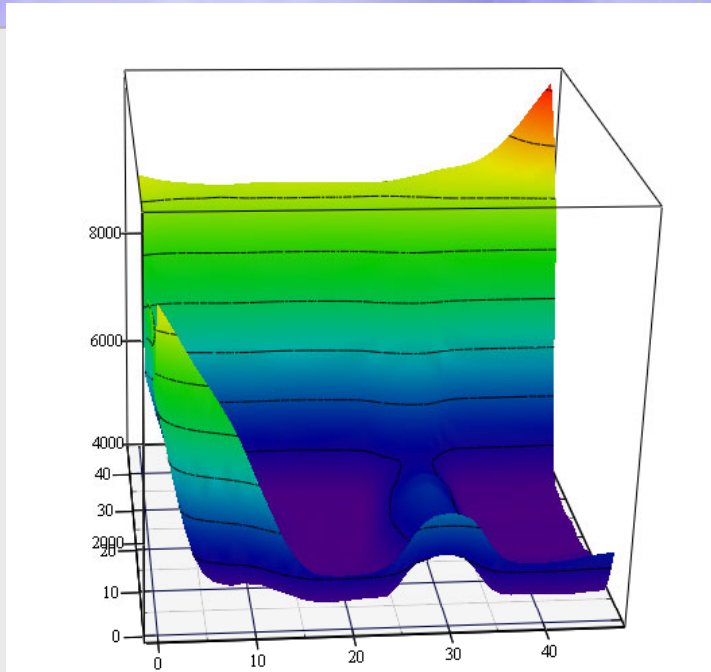
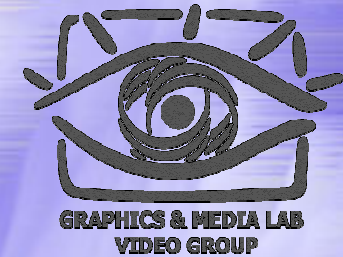
ME: Метрика блоков

Расстояние между блоками ищется как сумма модулей разности значений пикселов:

$$R(A, B) = \sum_{i=0, j=0}^{n, n} |a_{i,j} - b_{i,j}|$$

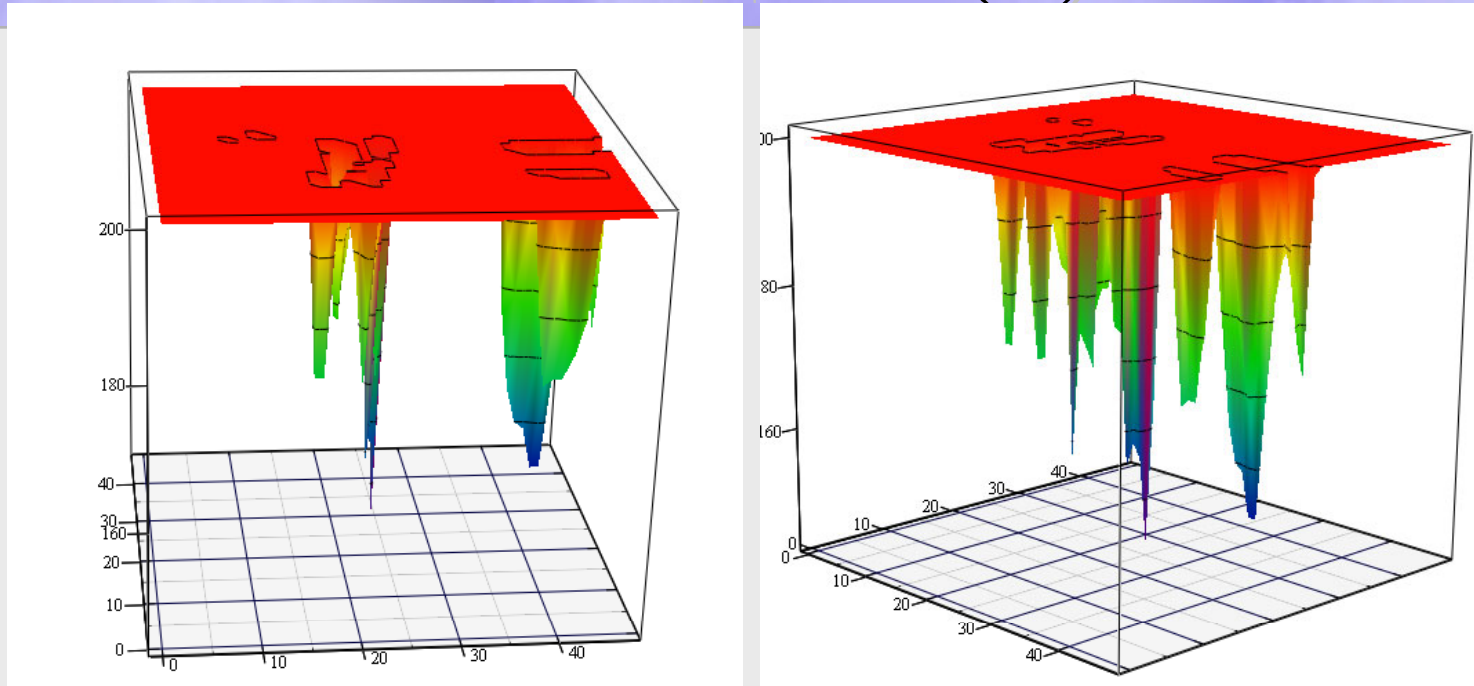
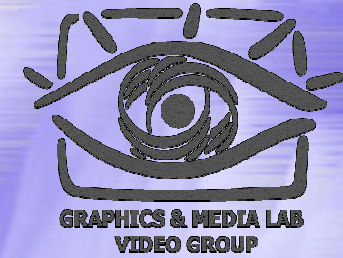
- + хорошо оптимизируется (важно, поскольку ME самая долгая операция при сжатии)
- неустойчива к шуму
- не учитывает изменения яркости

ME: Иллюстрация постановки задачи



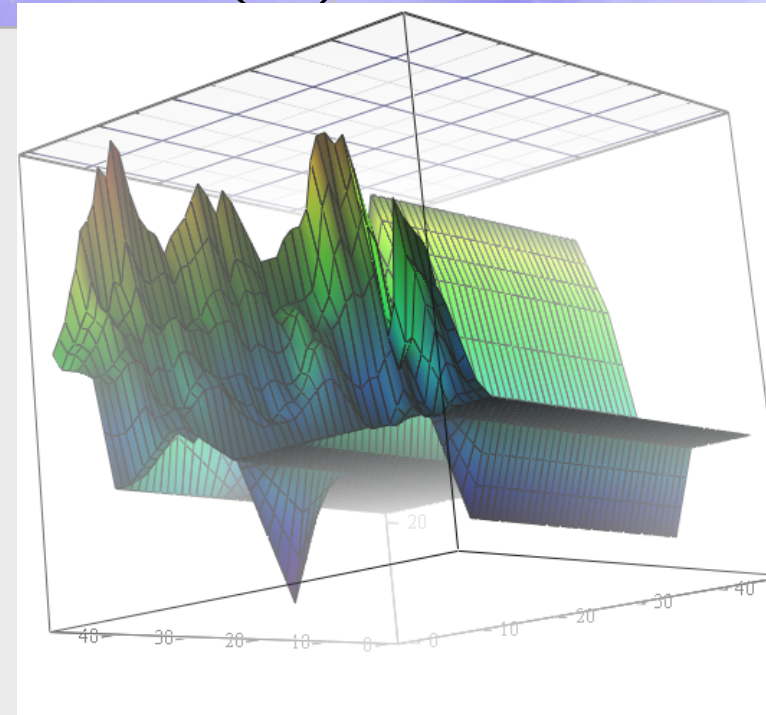
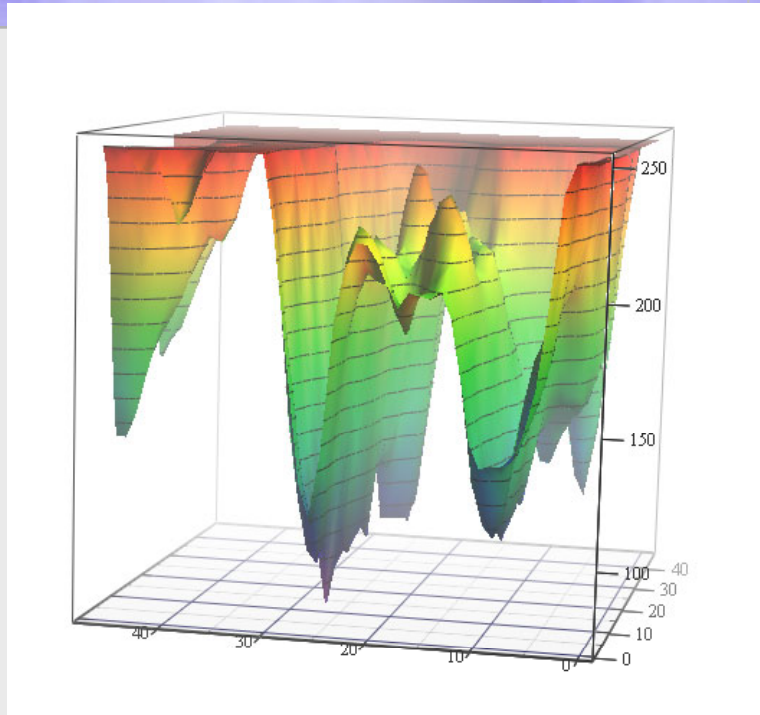
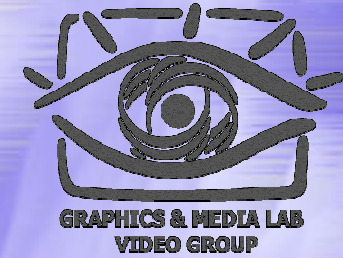
^f Величина ошибки от x, y . Взят более-менее типичный блок, сдвинувшийся на 2 пиксела в сравнительно ровной области (стена). На левом рисунке – исходный блок, на правом он же, но по границе 1000 обрезанный сверху (ракурс тот же).

ME: Иллюстрация постановки задачи (2)



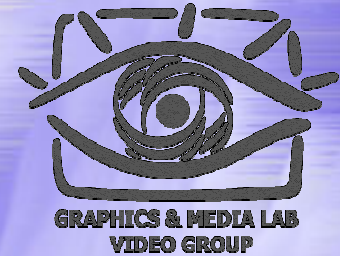
- ^f Тот же блок, обрезанный по границе 200 в том же ракурсе и в другом ракурсе. Хорошо видно, как много у этой функции (казавшейся на первом рисунке гладкой) локальных минимумов. Если алгоритм будет выбирать их, то блоки начнут «гулять»

ME: Иллюстрация постановки задачи (3)



- ^f Слева — блок сдвинувшийся на 6 пикселей по одной оси, и на 11 по другой, однако локальным минимумов у графика много.
Справа — также сильно сдвинувшийся блок (однако, минимум виден замечательно). «Ровный» график у дальней стенки — граница изображения

ME: Базовые стратегии поиска



**Задача: За минимальное число замеров
найти минимум.**

Используются комбинации стратегий:

- ◆ Cross поиск
- ◆ Ортогональный поиск
- ◆ Трехшаговый поиск (TSS)
- ◆ ME: Четырехшаговый поиск (FSS)

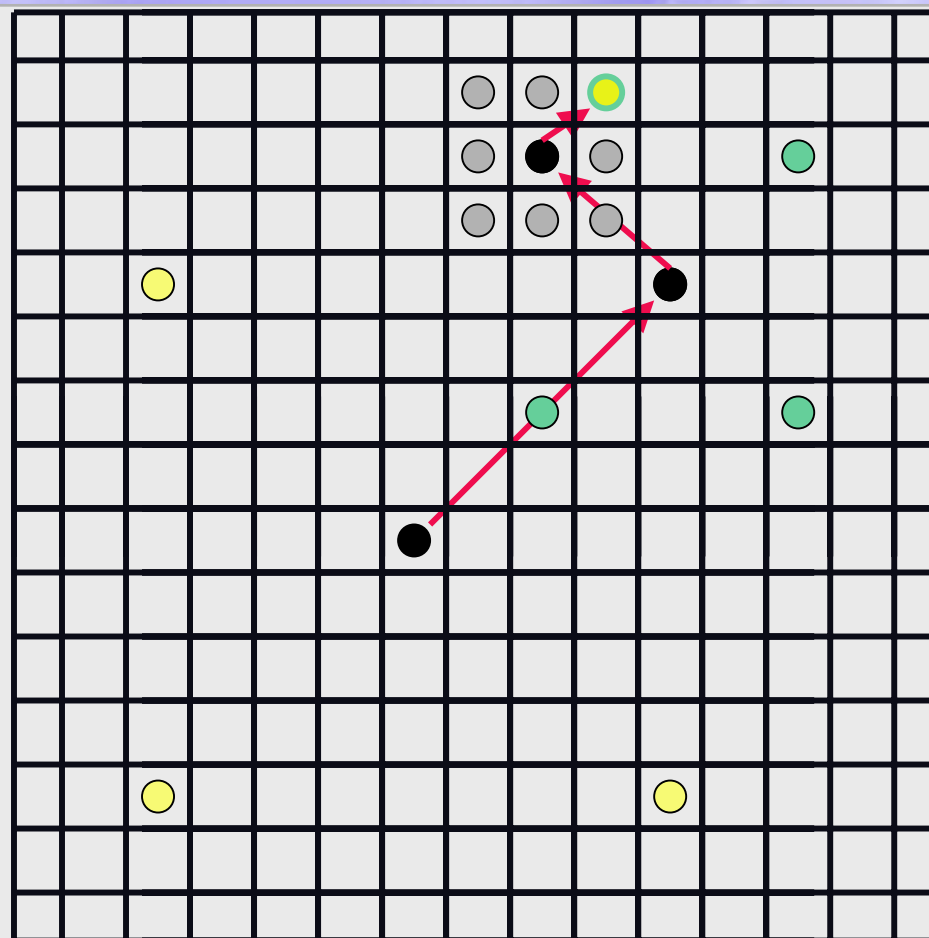
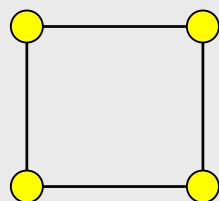
ME: Cross поиск



Cross поиск

Шаблон
уменьшается на
каждом шаге.

Шаблон:



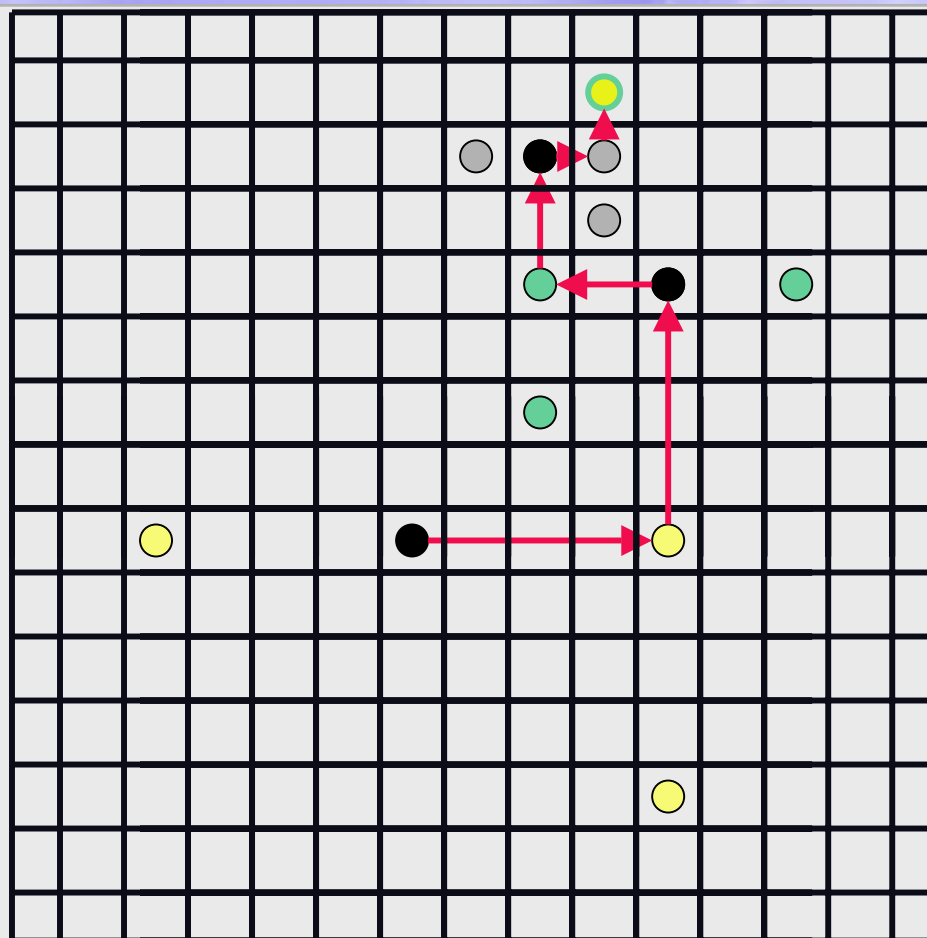
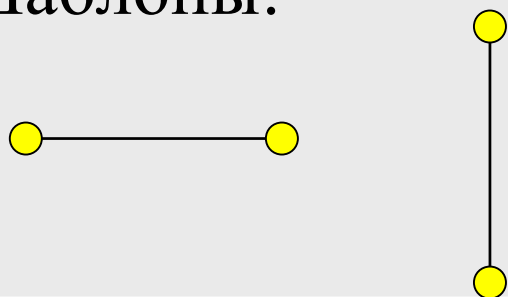
ME: Ортогональный поиск



Ортогональный поиск:

Пробуются попеременно вертикальное и горизонтальное направления.

Шаблоны:



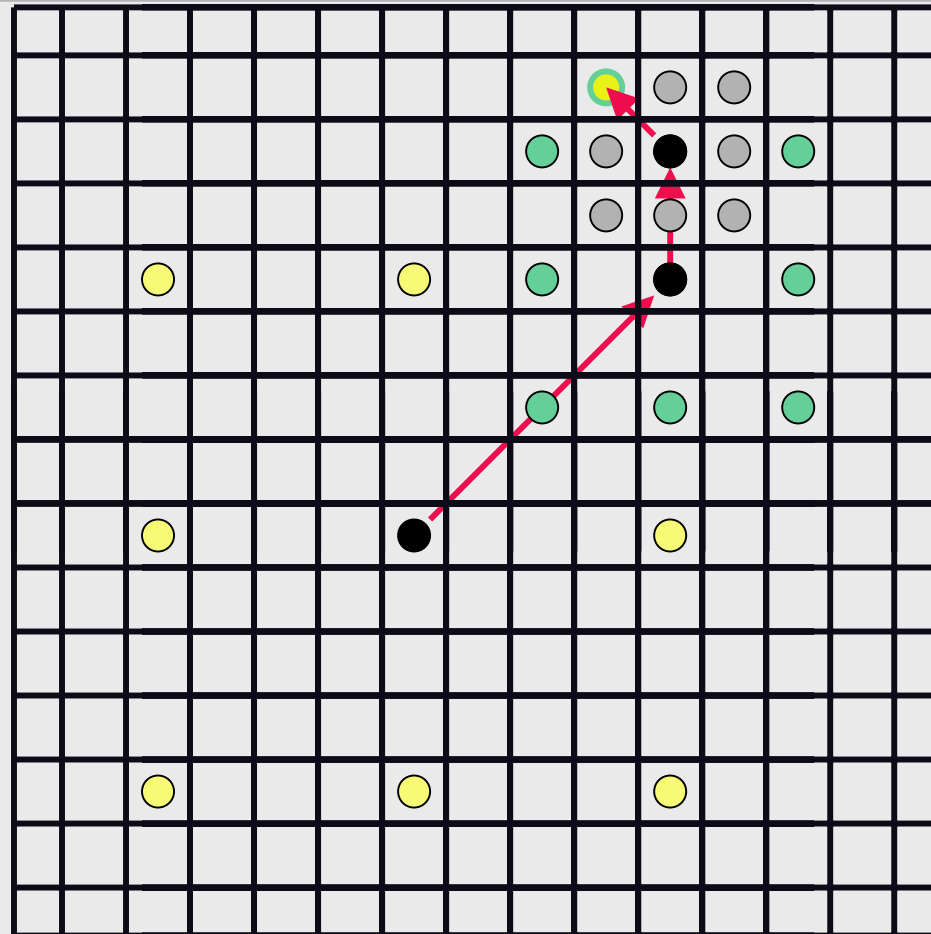
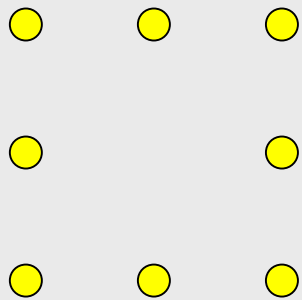
ME: Трехшаговый поиск (TSS)



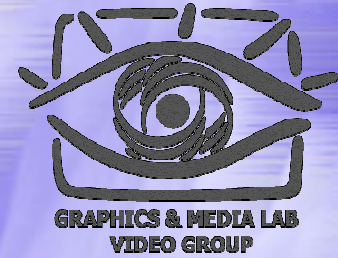
Three Step Search (TSS)

На каждом шаге размер шаблона уменьшается.

Шаблон:



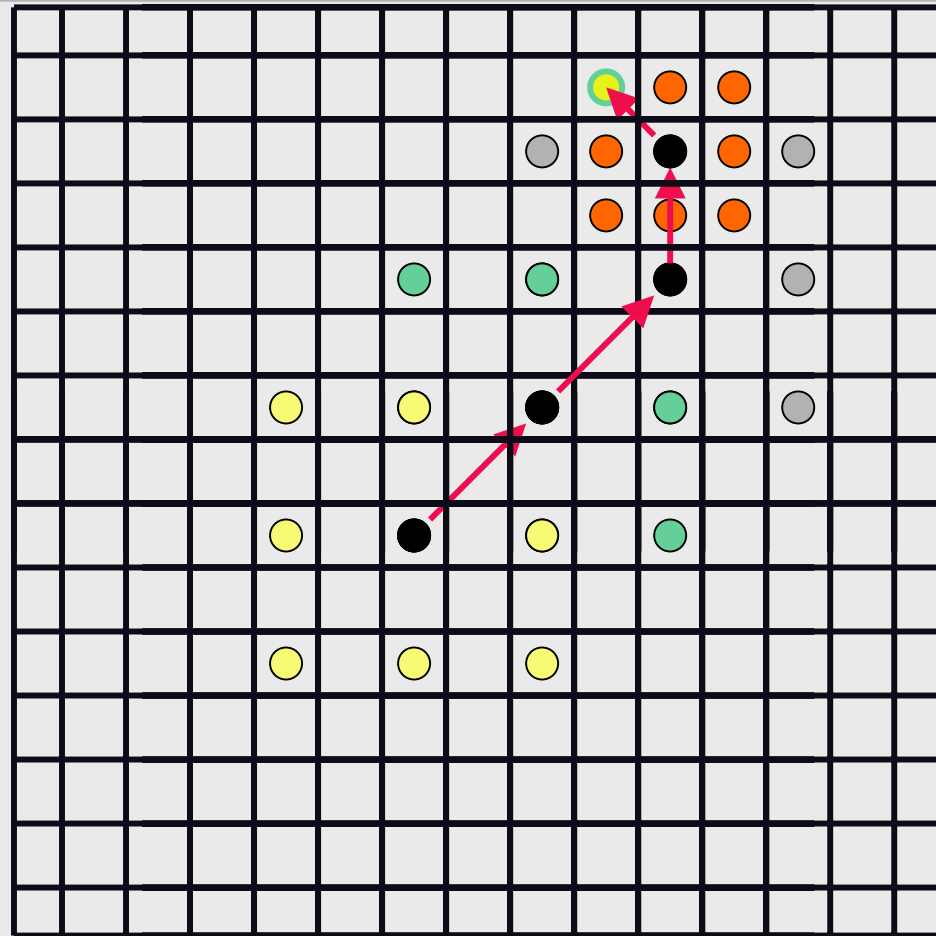
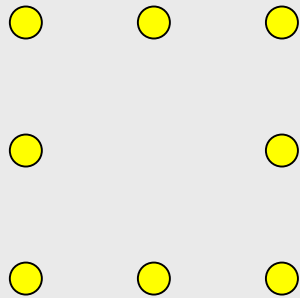
ME: Четырехшаговый поиск (FSS)



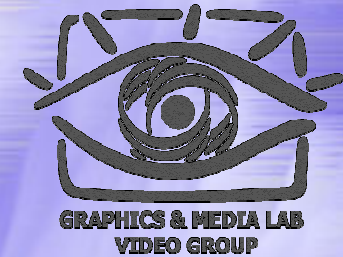
Four Step Search (FSS)

Шаблон не
уменьшается.

Шаблон:

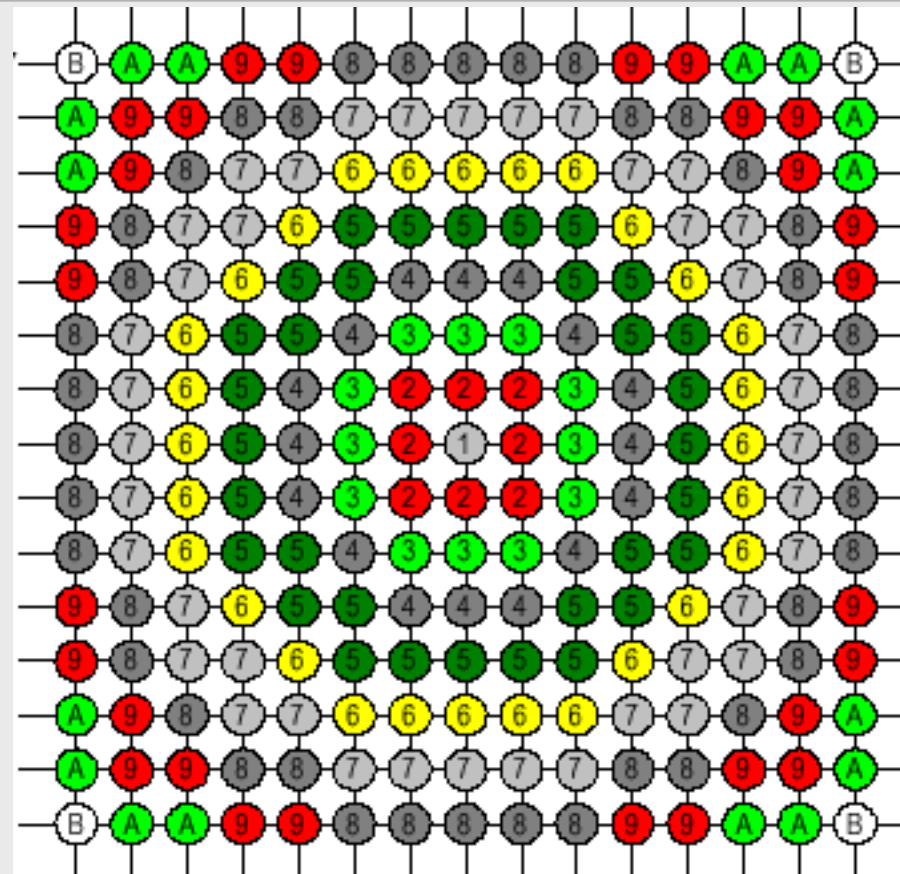


ME: Спиральный поиск



Спиральный поиск

Контр-пример:
Если мы будем
все время искать
с какого-то края,
то у нас будут
«ползти» ровные
куски кадра.



Пример MV-1

Кадр из фильма
«Кошки и собаки».

Векторами
показаны
распознанные
вектора движения
блоков.

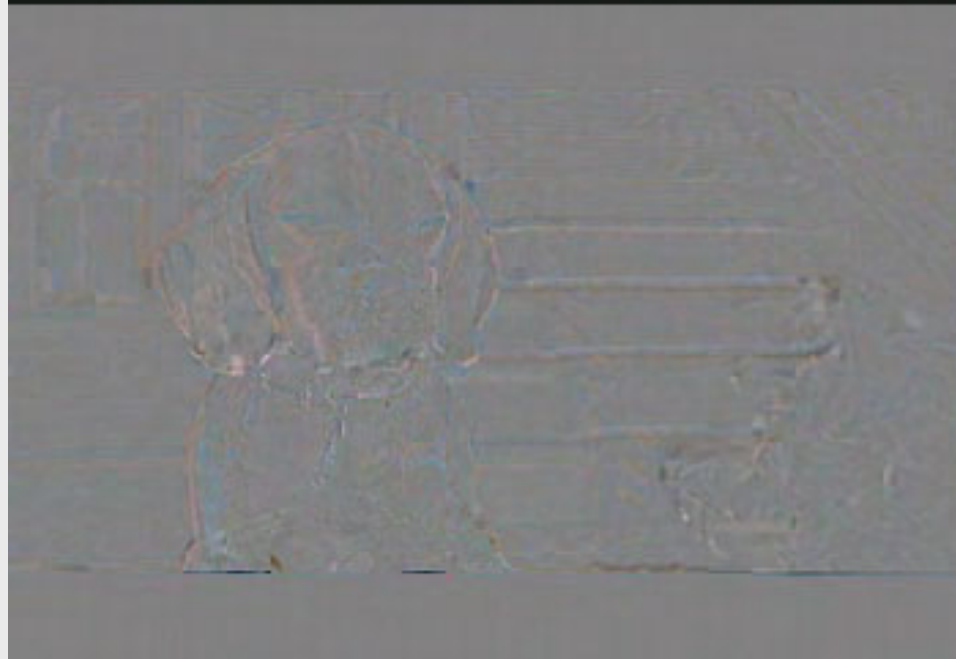


Пример MV-1

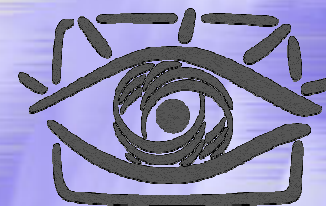
Кадр из фильма «Кошки и собаки».

Следующий кадр.

Показана межкадровая
разница, сравнительно
небольшая, благодаря
компенсации движения.



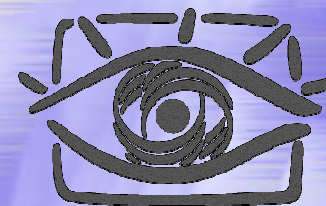
Пример MV-2



Кадр из
фильма
«Звездные
войны,
Эпизод 1».
Векторами
показаны
распознанные
вектора
движения
блоков.



Пример MV-2

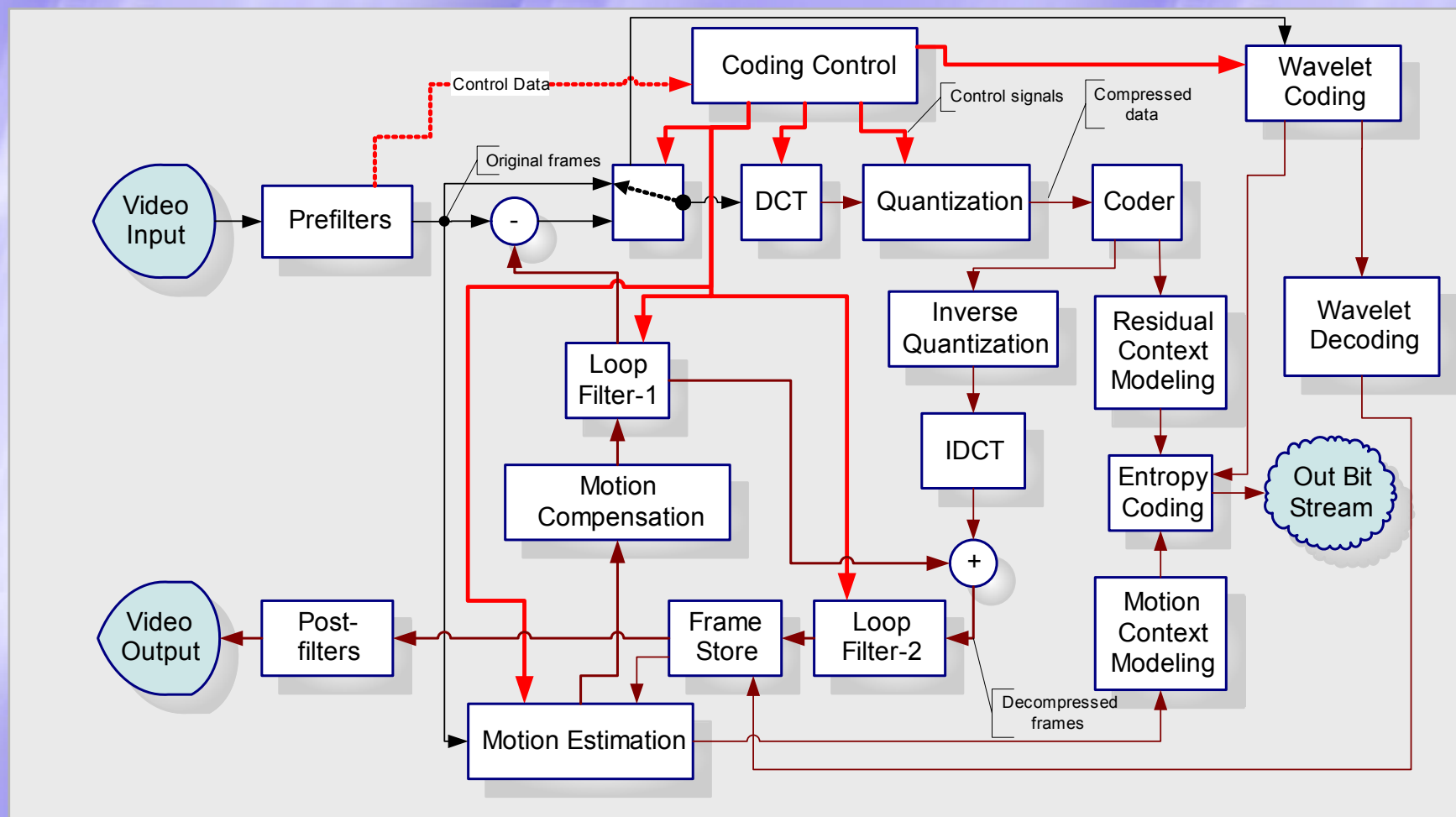


Следующий кадр

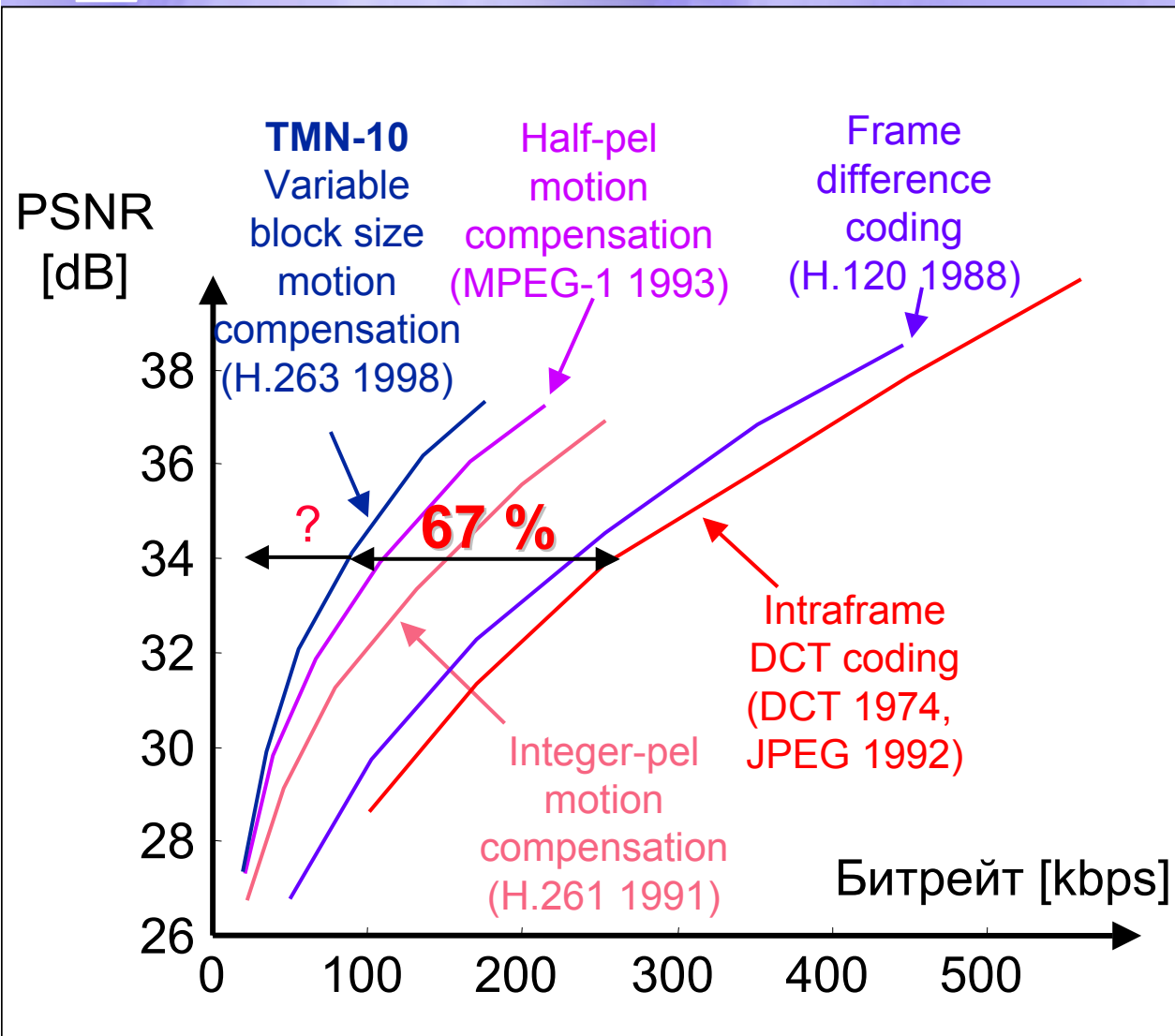
Хорошо видно, что хуже всего распознался «воротник за ЛОГОТИПОМ»



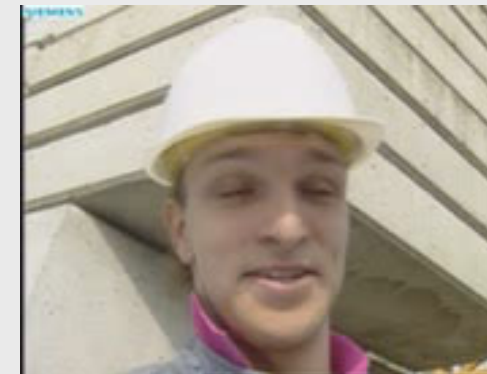
Схема видеокodeка



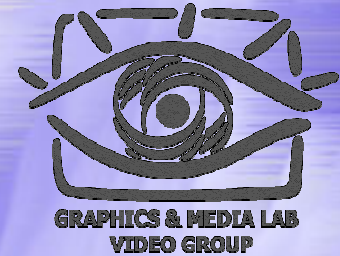
Развитие стандартов за 10 лет



Тестовая
последовательность
Foreman
10 кадров/с, QCIF
100 кадров всего

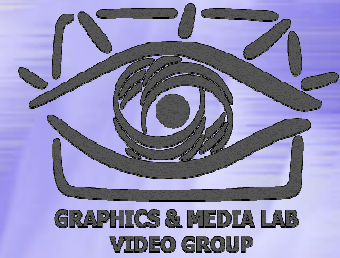


Структура лекций



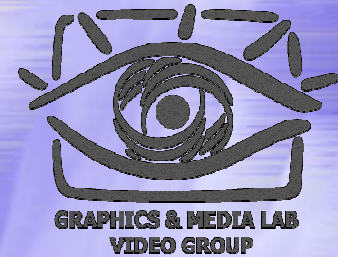
- ◆ Раздел 1. Основные понятия в сжатии видео
- ◆ Раздел 2. Построение простого видеокодека
- ◆ Раздел 3. Построение мощного видеокодека
- ◆ Раздел 4. Стандарты сжатия видео
- ◆ **Раздел 5. Новое трехмерное видео**

Благодарности



Автор выражает глубокую признательность Александру Жиркову за предоставленные слайды по трехмерному видео.

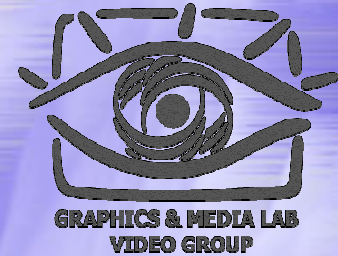
Сжатие 3d-видео



Трехмерное видео – новое направление развития алгоритмов обработки и сжатия видео (пока слабо распространены устройства показа)

- ◆ Стереоскопическое видео
- ◆ Интерактивное 3D Видео и MPEG-4

Определение 3D Видео (3DV)



‘3D Видео’ или ‘3D Анимация’ ?

- Видео получают путем съемки/захвата
- Анимацию получаю путем моделирования

Требования к представлению 3DV:

- Визуализация в реальном времени
- Произвольная сложность объектов и их движений
- Эффективная компрессия
- Интерактивность (*)

Подходы к 3DV



Представления

- Стереоскопическое видео
- Multi-View видео
- Поточковая передача 3D
- Поточковая передача параметров лица
- Видео Текстуры

Визуализация

- Обычный дисплей
- 3D очки
- 3D экран

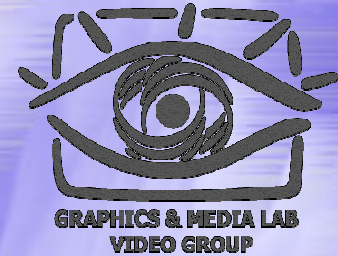
3D / Стандарты стереоскопического видео



Основанные на аналоговых 2D Видео:

- Система с чередованием полей (Field-Sequential):
 - ◆ Четные строки – левый канал, нечетные – правый
 - ◆ Наиболее простой, выгодно использует Interlacing в аналоговых форматах, но не преобразуем между:
 - NTSC(North America,Japan): 60Hz, 525lines
 - PAL, SECAM: 50Hz, 625
 - ◆ Необходимо: очки со ‘ставнями’ и устройство синхронизации очков с текущим изображением левого/правого канала

3D / Стандарты стереоскопического видео



Основанные на аналоговых 2D Видео:

- Система с чередованием строк (Field-Sequential):
 - ◆ Четные строки – левый канал, Нечетные – правый
 - ◆ Наиболее простой, выгодно использует Interlacing, но не преобразуем между:
 - NTSC(North America,Japan): 60Hz, 525lines
 - PAL, SECAM: 50Hz, 625
 - ◆ Необходимо: очки со ‘ставнями’ и устройство синхронизации очков с текущим изображением левого/правого канала

3DV / Стандарты стереоскопического видео



Менее распространенные представления:

- Разделение каналов по сторонам (Sidefields)
- Разделение каналов по высоте (Subfields)
- Отдельное хранение каналов (Separate Channels)

Анаглифы:

- Разделение каналов с использованием различных цветовых составляющих каналов, обычно синего и красного
- Недостаток: невозможность передачи полноцветного изображения
- Преимущество: необходимы только очки с цветовыми фильтрами

3DV / Стандарты стереоскопического видео



- ◆ Преимущества стереоскопического видео:
 - Совместим с традиционными форматами записи 2D видео
 - Нет необходимости в восстановлении 3D информации
- ◆ Недостатки:
 - Необходимость дополнительного оборудования
 - Фиксированная точка наблюдения, отсутствие интерактивности

3DV/ Интерактивное 3D Видео



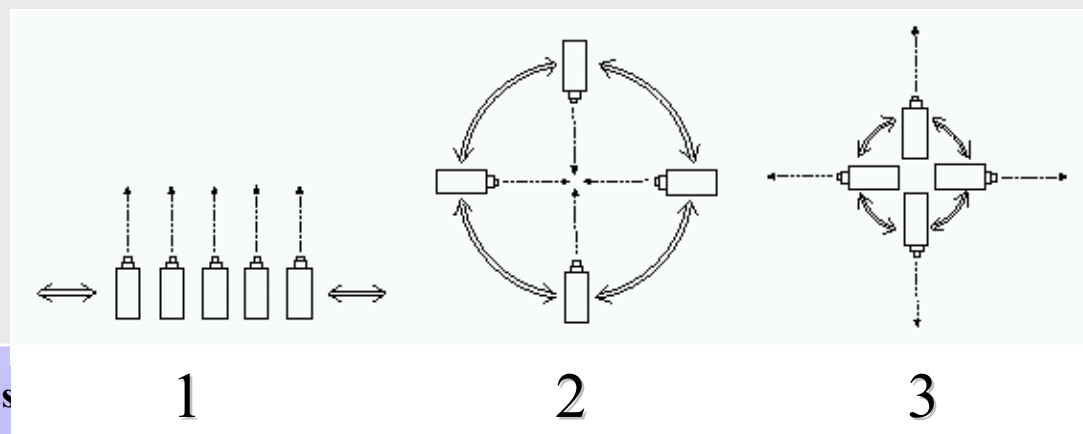
- ◆ Интерактивное 3D видео зависит от положения и ориентации наблюдателя.
- ◆ Для такого представления необходима группа калиброванных в пространстве и времени видео потоков + информация о третьем измерении (глубине).
- ◆ Для визуализации такого представления используется технологии Image-Based Rendering, а точнее к Video-Based Rendering

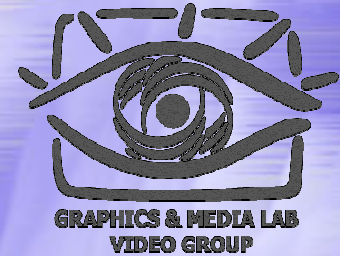
3DV/ Интерактивное 3D Видео



Примеры интерактивного видео:

- ◆ Одна камера с каналом глубины
- ◆ Стереоскопическое видео с восстановлением глубины (1)
- ◆ Свободные камеры (2) с глубиной– для представления 3D видео объектов
- ◆ Панорамное видео (3) – канал глубины не требуется





Существуют ли стандарты для интерактивного 3D видео?

◆ Multiview profile (MPEG-2)

Предназначен для сжатия множества видео потоков, но не для 3D Видео, отсутствие информации о калибровке камер

◆ Multiple auxiliary components (MPEG-4)

Позволяет кодировать канал глубины, но не приспособлен для этого, отсутствие информации о калибровке камер

◆ MPEG-4 Animation Framework eXtension (AFX) :

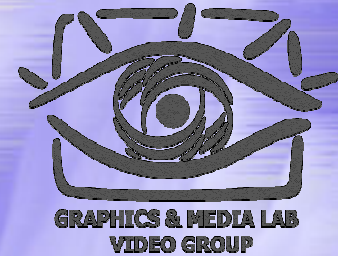
● Light Field Mapping (LFM):

Multi-View Images, но не Multi-View Video

● Depth-Image Based Representations (DIBR):

Позволяет представлять, компрессировать и интерактивно визуализировать 3D Видео

3DV / Depth-Image Based Representations (MPEG-4 AFX)



◆ Представление состоит из:

- Множество калиброванных видео потоков
- Два варианта сжатия информации о глубине:
 - ◆ Отдельные видео потоки глубины или альфа-каналы
 - ◆ Компрессированный поток Текстурированных Бинарных Волуметрических Октодеревьев
 - Большая степень сжатия по сравнению с хранением отдельных каналов глубины
 - Позволяет хранить в несколько раз меньшее число цветowych видеопотоков на объект

◆ Недостатки:

- Мощности современных компьютеров хватает только на воспроизведение 3D видео объектов с разрешением порядка 512^3 , на 3D видео фильмы мощности не хватает

Задания



- ◆ Задания по курсу расположены на странице курса:

<http://graphics.cs.msu.su/courses/mdc2004/>