

# **Введение в сжатие видео**

**Дмитрий Ватолин**

*Московский Государственный Университет  
CS MSU Graphics&Media Lab*

# 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.

# 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.

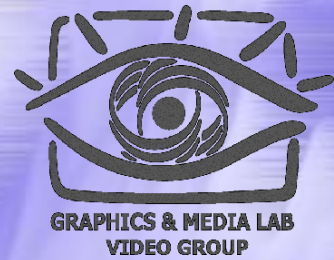
**Временная  
избыточность** –  
кадры весьма похожи

# 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

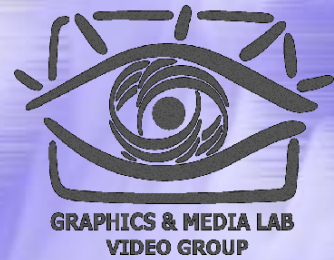


- ◆ **Не существует метода оценки качества кадра, полностью адекватного человеческому восприятию**
- ◆ **Не существует метода оценки пропущенных кадров, полностью адекватного человеческому восприятию**

**Следствие: Можно декларировать любую степень сжатия в маркетинговых материалах.**



# PSNR



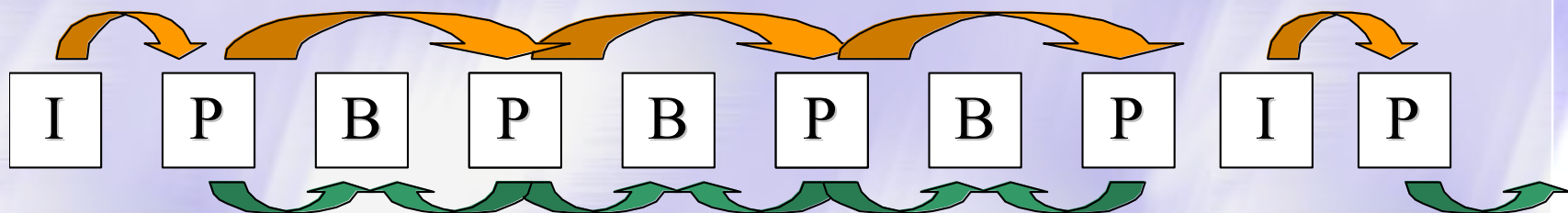
Базовые метрики –  
Y-PSNR, U-PSNR, V-PSNR

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

Хорошо работают только на высоком  
качестве.



# 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

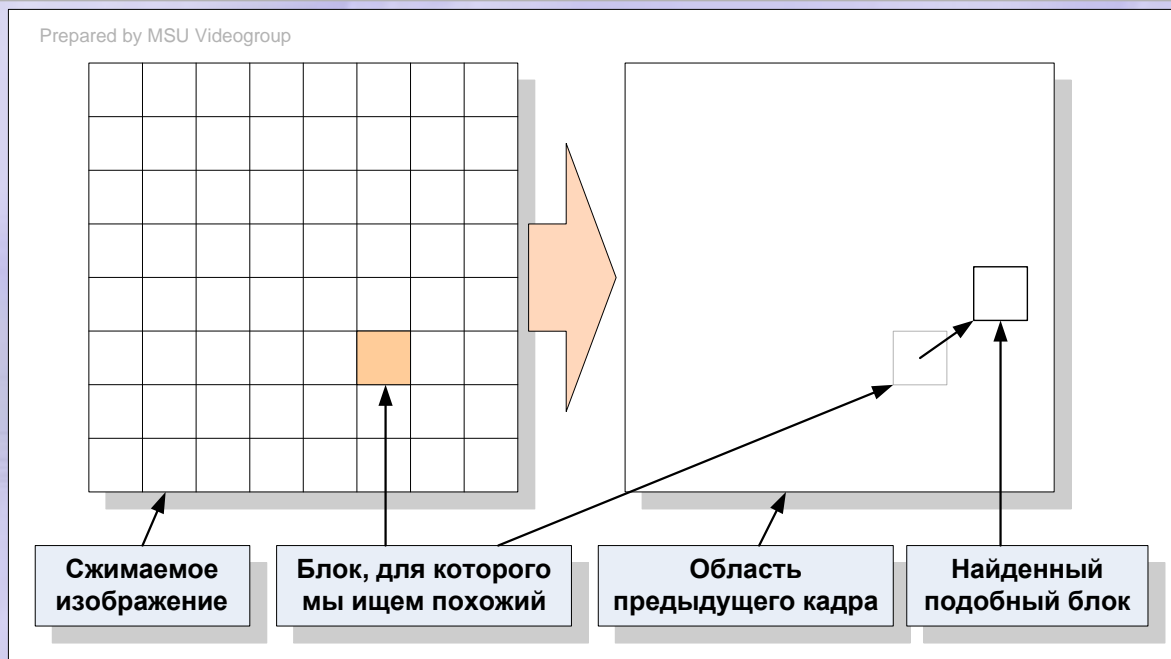
# Компенсация движения (2)



- ◆ 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)

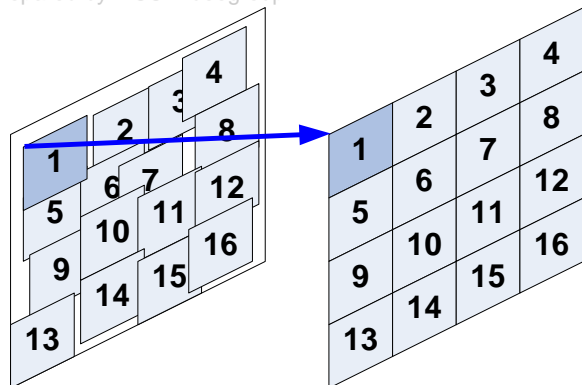


Для каждого блока в кадре мы находим похожий в предыдущем кадре в некоторой окрестности положения блока.

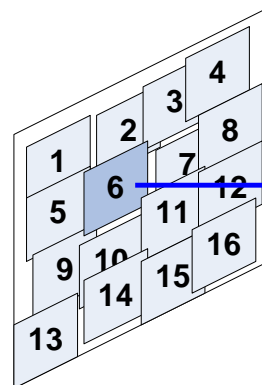
Если достаточно похожий блок в предыдущем кадре не найден – блок сжимается независимо (*Intra-Blocks*).

# Движение для В-кадра

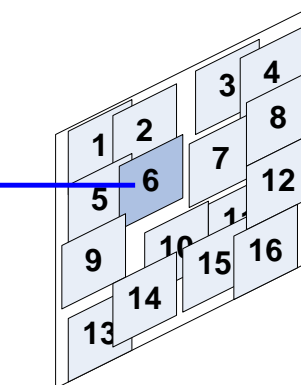
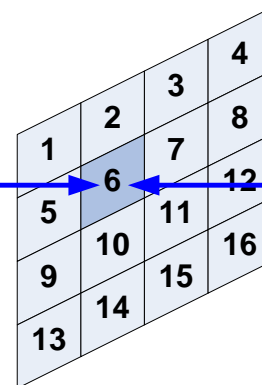
Prepared by MSU Videogroup



Предыдущий кадр P-кадр



Предыдущий кадр B-кадр

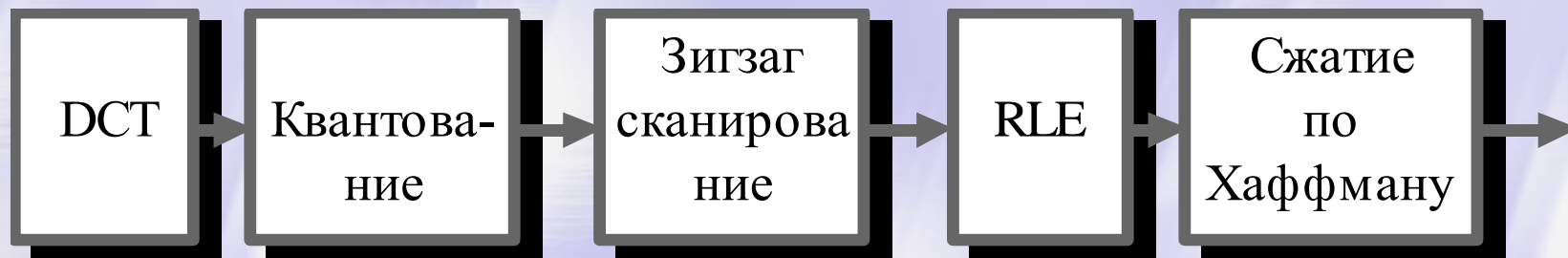


Следующий кадр

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 разницы очень похожа на сжатие JPEG: блоки 8x8 сжимаются помощью дискретного косинусного преобразования



# Схема простого ВИДЕОКОДЕКА

