Средства и приемы обработки видео

Дмитрий Ватолин

Московский Государственный Университет
CS MSU Graphics&Media Lab

Version 1.2

Содержание:

GRAPHICS & MEDIA LAB
VIDEO GROUP

- MMX технология
- Программа VirtualDub
- Программа AviSynth
- Программа Mathcad

MMX™ Technology

Потоковая обработка данных.
Средство существенного увеличения
скорости работы видеофильтров.

Курс по Intel MMX

GRAPHICS & MEDIA LAB
VIDEO GROUP

В слайдах использованы рисунки из курса по MMX компании Intel, который настоятельно рекомендуется пройти.

Курс можно скачать по адресу:

http://graphics.cs.msu.su/courses/mdc2004/library/mintro.exe

(размер: 14 МБ)

http://graphics.cs.msu.su/courses/mdc2004/library/runcbt.exe

(размер: 2,7 МБ)

История возникновения

GRAPHICS & MEDIA LAB
VIDEO GROUP

Технология MMX была разработана компанией Intel и является своего рода развитием команд процессора.

Технология базируется на архитектуре процессора Pentium® и позволяет **ускорить вычисления** за счет **параллельной обработки данных**.

Где применяется технология MMX™ ?

GRAPHICS & MEDIA LAB
VIDEO GROUP

Технология MMX используется во многих мультимедийных приложениях, например **при обработке видео, звука и графики** (ускорение цифровой обработки сигналов и данных).

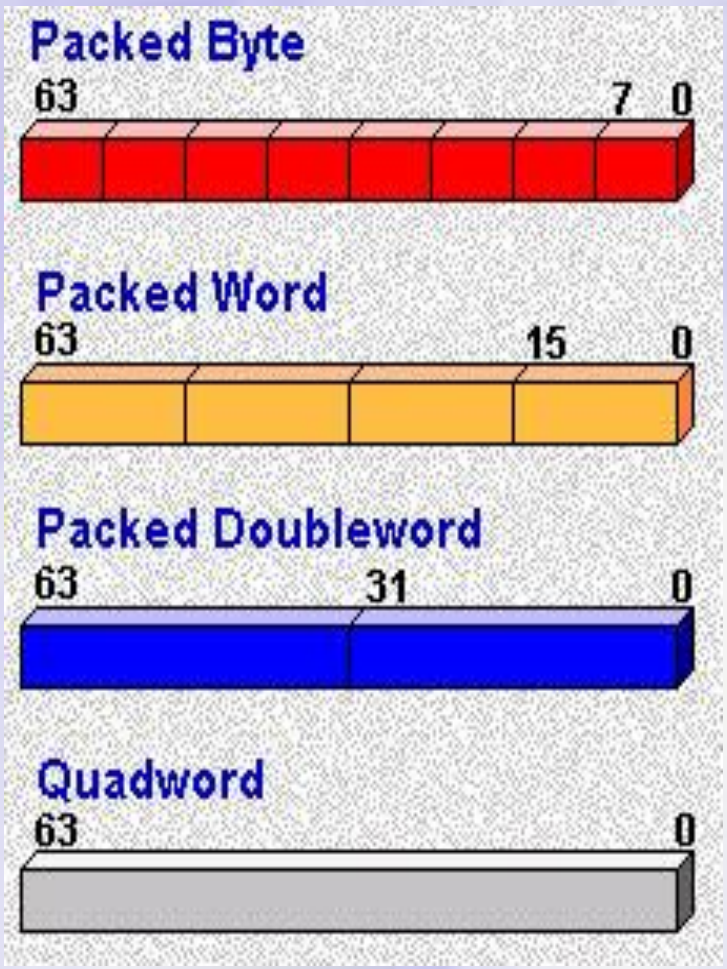
Забавно, что MMX команды сегодня применяются даже **при заполнении и копировании** буферов операционной системы.

MMX™ технология

GRAPHICS & MEDIA LAB
VIDEO GROUP

1. В технологии MMX применяются инструкции использующие особенности архитектуры нового процессора.
2. MMX инструкции работают как со знаковой так и с беззнаковой арифметикой.
3. Появились 8 новых регистров с соответствующими именами MM0…MM7.

Типы данных в MMX™



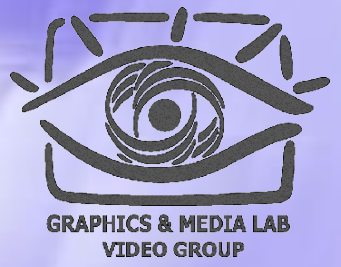
В 64 бита можно поместить от 8 «переменных» размером по 8 бит (байтов) и до одной «переменной» размером в 64 бита.

Система команд MMX™

Система команд MMX состоит из 57 команд, сгруппированных в следующие категории:

- Команды передачи данных
- Арифметические команды
- Команды сравнения
- Команды преобразования
- Логические команды
- Команды двига
- Команда освободить MMX™ состояние (EMMS)

Семантика инструкций MMX™



<команда>[dest,src]

<команда> записывается по следующим правилам:

1) Команда начинается с “P” (кроме movd, movq)
2) “US” работа с без знаковой арифметикой
3) “S” или “SS” работа со знаковой арифметикой
4) “B”, “W”, “D”, “Q” соответственно обозначают тип с которым работает инструкция

Арифметика с насыщением

MMX технология поддерживает арифметику с насыщением (saturated arithmetics).

- В **режиме с насыщением**, результаты операции, которые переполняются сверху или снизу отсекаются к границе *datarange* соответствующего типа данных
- В **режиме без насыщения**, результаты, которые переполняются как в обычной процессорной арифметике (см. курсы по C и ассемблеру).

Таблица

GRAPHICS & MEDIA LAB
VIDEO GROUP

Тип данных	Нижний предел		Верхний предел	
	Шестн адцат.	Десяти чн.	Шестн адцат.	Десати чн.
Знаковый байт	80H	-128	7FH	127
Знаковое слово	8000H	-32768	7FFFH	32767
Беззнаковый байт	00H	0	FFH	255
Беззнаковое слово	0000H	0	FFFFH	65535

Команды передачи данных (пример)

MOVD (Переместить 32 Бита) передает 32 бита упакованных данных из памяти в регистры MMX и обратно, или из целочисленных регистров в регистры MMX и обратно.

MOVQ (Переместить 64 Бита) передает 64 бита упакованных данных из памяти в регистры MMX и обратно, или между регистрами MMX.

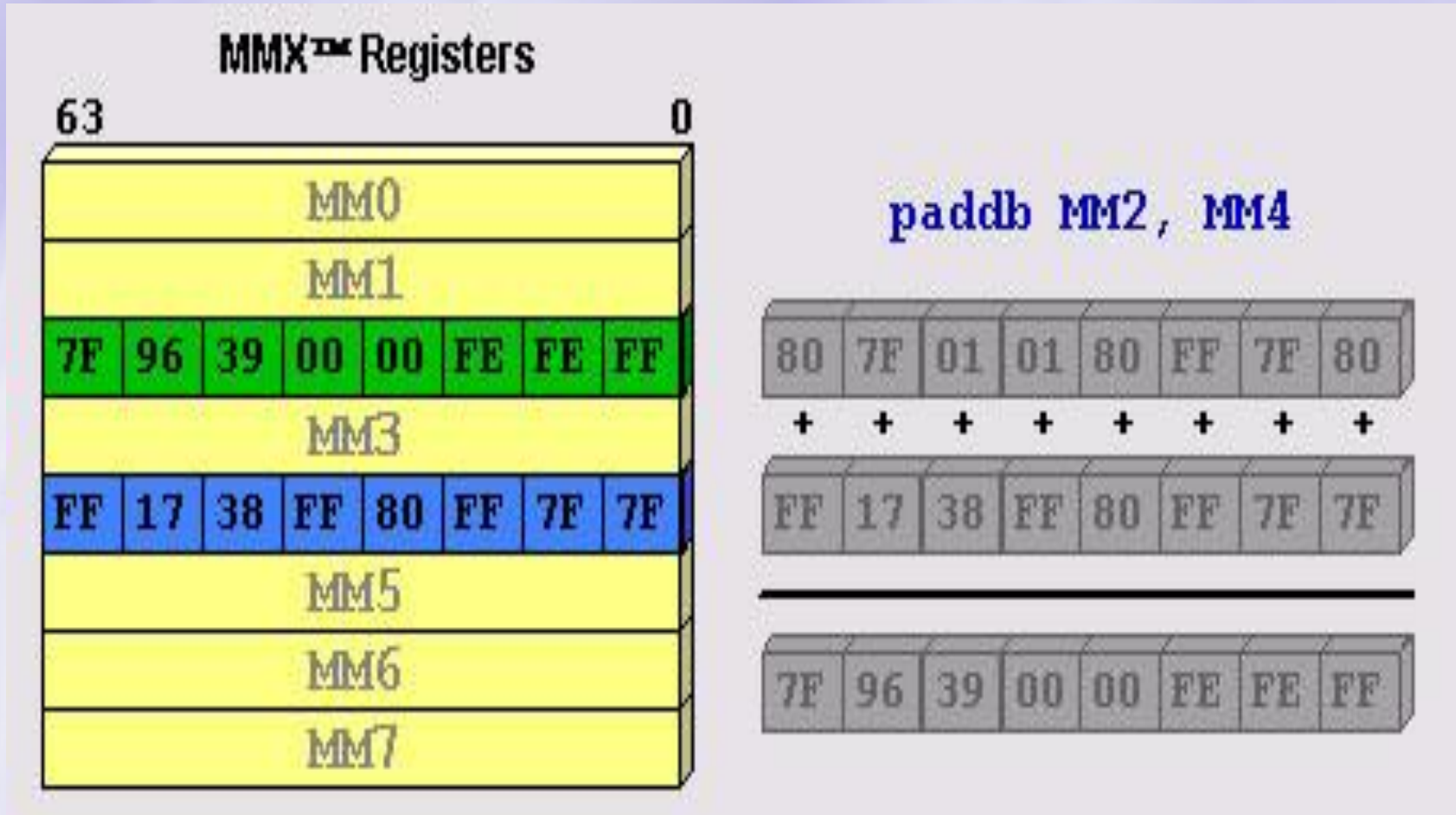
Пример арифметических инструкций

GRAPHICS & MEDIA LAB
VIDEO GROUP

Арифметичес-кие	Wraparound	Знаковая	Без знаковая
Сложение	PADD	PADDS	PADDUS
Вычитание	PSUB	PSUBS	PSUBUS
Умножение	PMULL/H		
Умножение и сложение	PMADD		

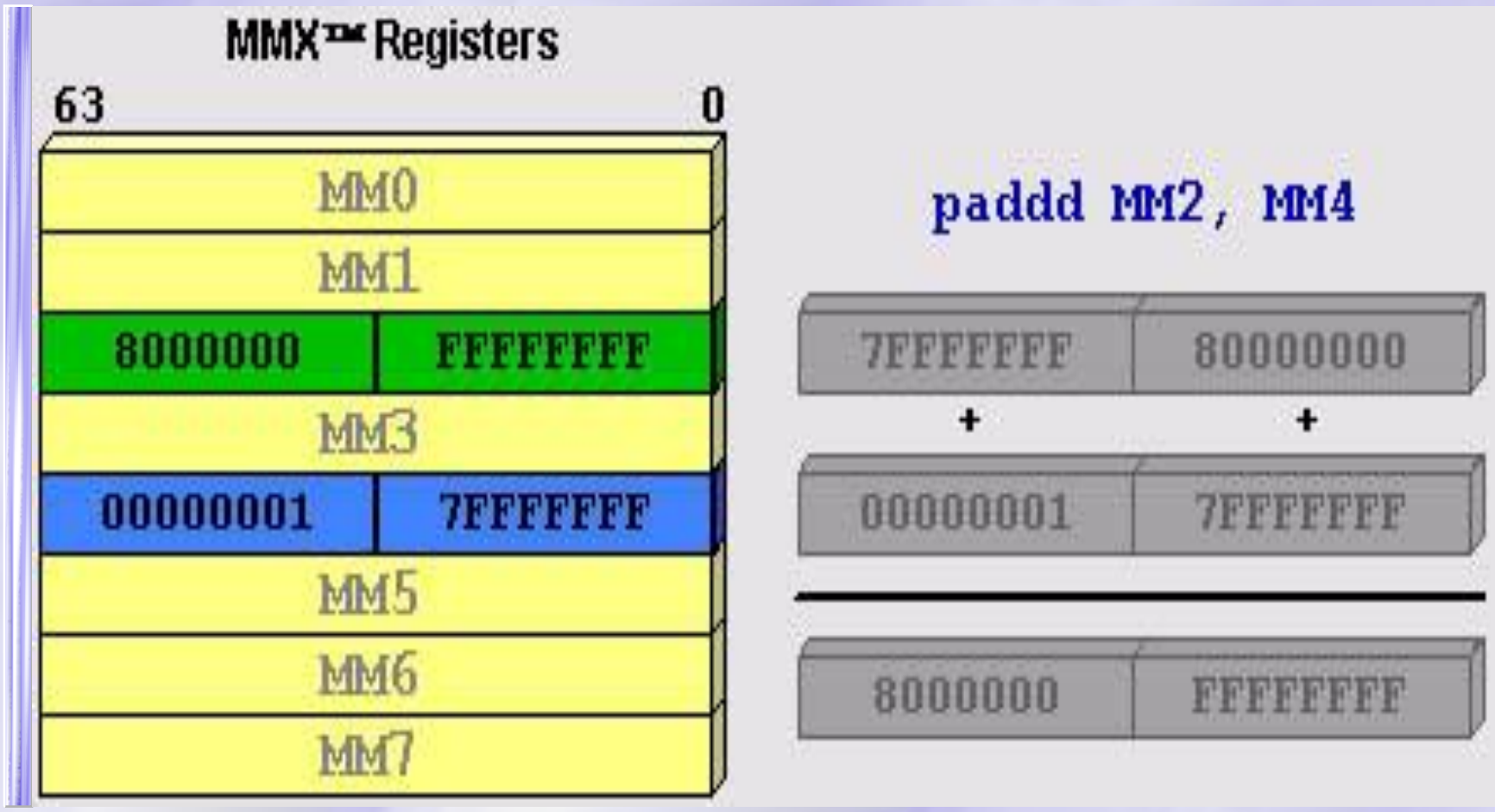
Пример для сложения типа Byte

GRAPHICS & MEDIA LAB
VIDEO GROUP

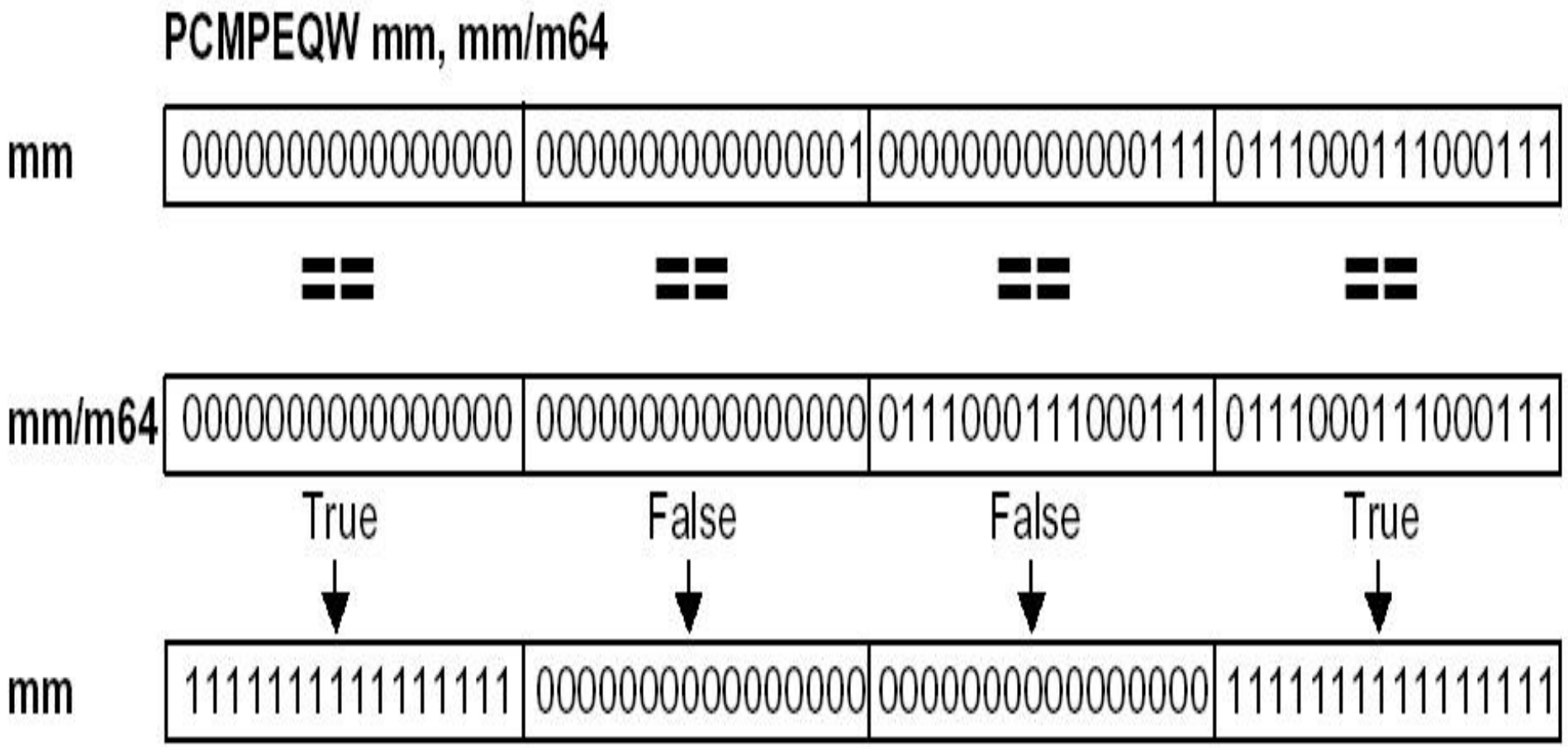


Пример для сложения типа Word

GRAPHICS & MEDIA LAB
VIDEO GROUP



Пример для сравнения



Пример кода с MMX™

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
1   movq MM0, [a_vector]
2   movq MM1, [b_vector]
3   pmaddwd MM0, MM1
4   paddd MM7, MM0
5   add [a_vector], 8
6   add [b_vector], 8
7   sub [count], 4
8   jnz loop
9   movq MM0, MM7
10  psrlq MM7, 32
11  paddd MM7, MM0
12  movd mem_vdp, MM7
```

Расчет скалярного произведения:

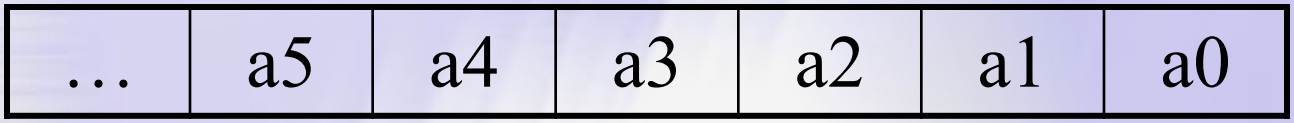
$$\sum a(i) * b(i)$$

Пример кода с MMX™

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
1 movq MM0, [a_vector]
2 movq MM1, [b_vector]
```

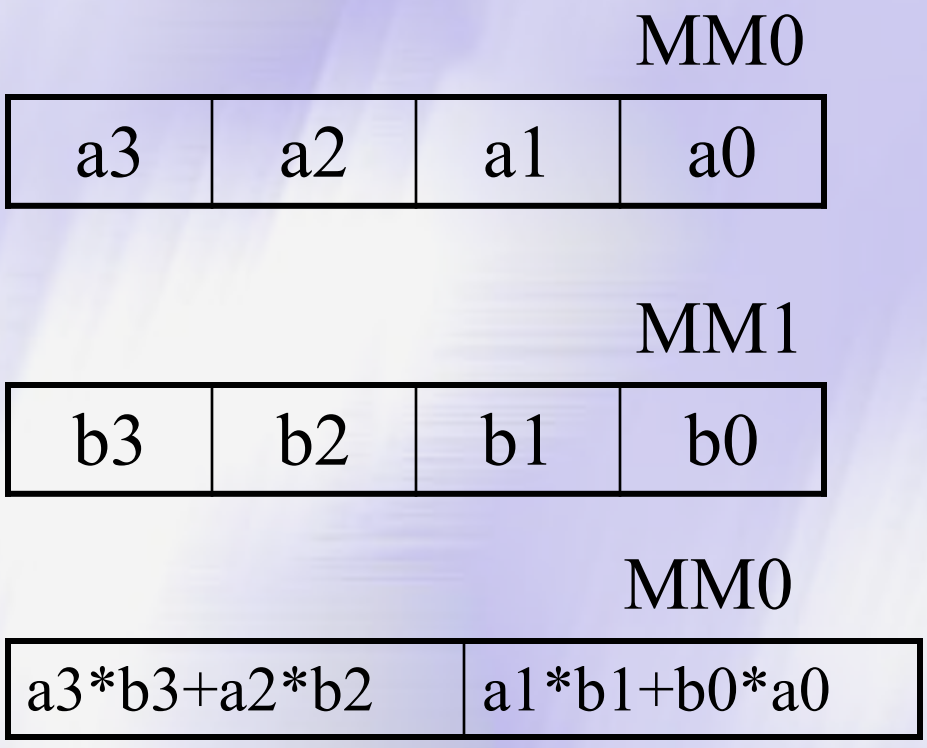
A_vector



MM0

Пример кода с MMX™


```
3 pmaddwd MM0, MM1
```



Пример кода с MMX™


```
4 paddd MM7, MM0
```

MM7

00000000	00000000

paddd

MM0

a3*b3+a2*b2	a1*b1+b0*a0

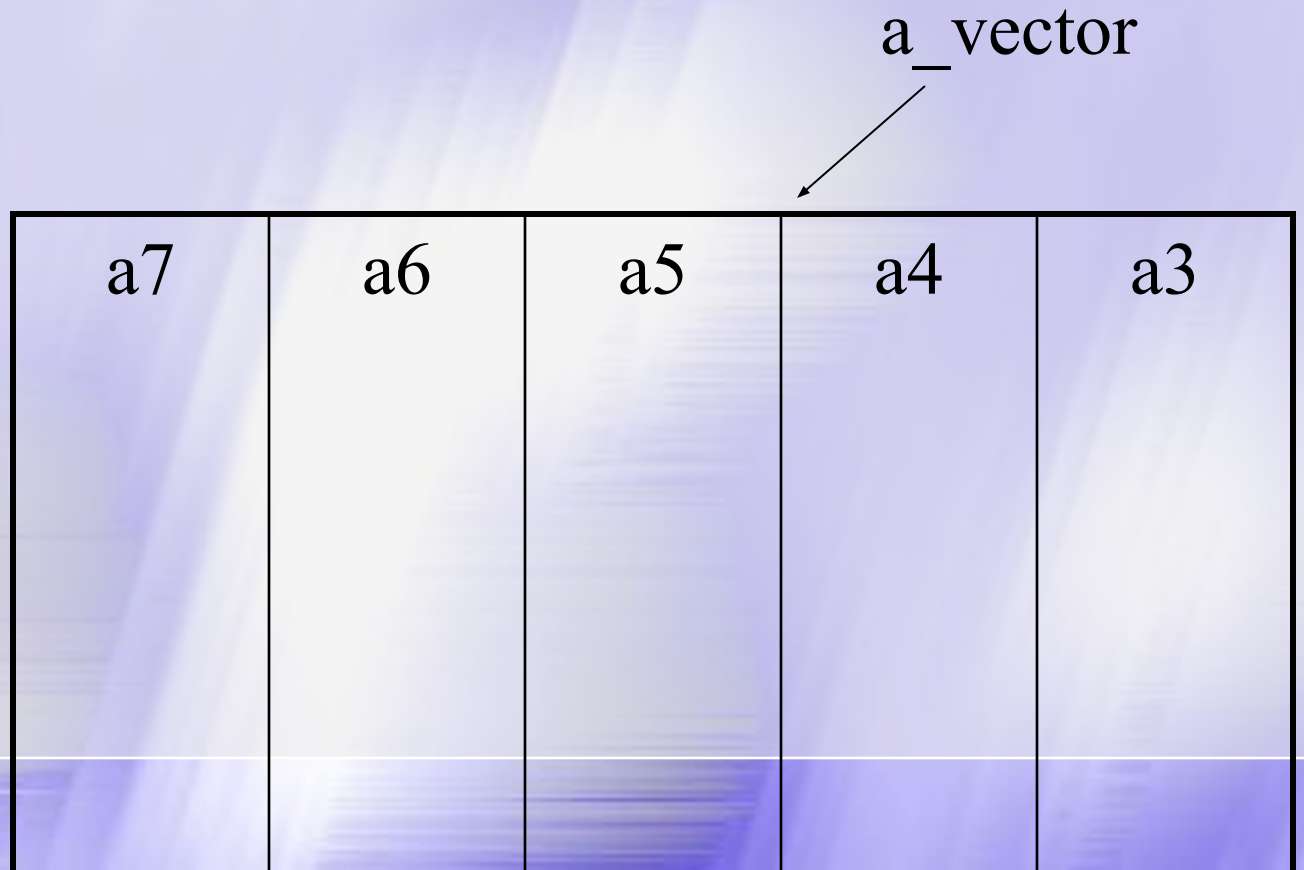
MM7

a3*b3+a2*b2	a1*b1+b0*a0

Пример кода с MMX™

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
5 add [a_vector], 8
6 add [b_vector], 8
```



Пример кода с MMX™

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
7 sub [count], 4
```

Счетчик уменьшаем на 4.
Уже обработано 4 элемента

```
8 jnz loop
```

Продолжается цикл если ещё осталось что обрабатывать

Пример кода с MMX™

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
9 movq MM0, MM7
```

MM0

a11*b11+a10*b10+a7*b7 +a6*b6+a3*b3+a2*b2	A9*b9+a8*b8+a5*b5+a4 *b4+a1*b1+a0*b0

Пример кода с MMX™


```
10 psrlq MM7, 32
```

MM7

a11*b11+a10*b10+a7*b7 +a6*b6+a3*b3+a2*b2	a9*b9+a8*b8+a5*b5+a4* b4+a1*b1+a0*b0

shift

MM7

00000000	a11*b11+a10*b10+a7*b7 +a6*b6+a3*b3+a2*b2

Пример кода с MMX™

GRAPHICS & MEDIA LAB
VIDEO GROUP

11 paddd MM7, MM0

MM7

a11*b11+a10*b10+a7*b7 +a6*b6+a3*b3+a2*b2	A9*b9+a8*b8+a5*b5+a4 *b4+a1*b1+a0*b0

paddd MM0

00000000	a11*b11+a10*b10+a7*b7 +a6*b6+a3*b3+a2*b2

MM7

a11*b11+a10*b10+a7*b7 +a6*b6+a3*b3+a2*b2	Наш результат

Содержание:

GRAPHICS & MEDIA LAB
VIDEO GROUP

- MMX технология
- Программа VirtualDub
- Программа AviSynth
- Программа Mathcad

VirtualDub

Лучшая программа для работы с потоковым видео

План

GRAPHICS & MEDIA LAB
VIDEO GROUP

- ✔ О программе VirtualDub
- ✔ Как писать фильтры
- ✔ Пример
- ✔ Итоги

Что это такое?

GRAPHICS & MEDIA LAB
VIDEO GROUP

VirtualDub является бесплатно распространяемой программой.

Это САМАЯ распространенная программа для поточной обработки видео (в т.ч. Подготовки MPEG-4 фильмов с DVD).

У неё открытые исходники, что позволяет модифицировать исходный код программы.

Где можно скачать и узнать о VirtualDub

GRAPHICS & MEDIA LAB
VIDEO GROUP

http://virtualdub.org/

Это официальный сайт VirtualDub. Здесь можно скачать последние версии и документацию по использованию.

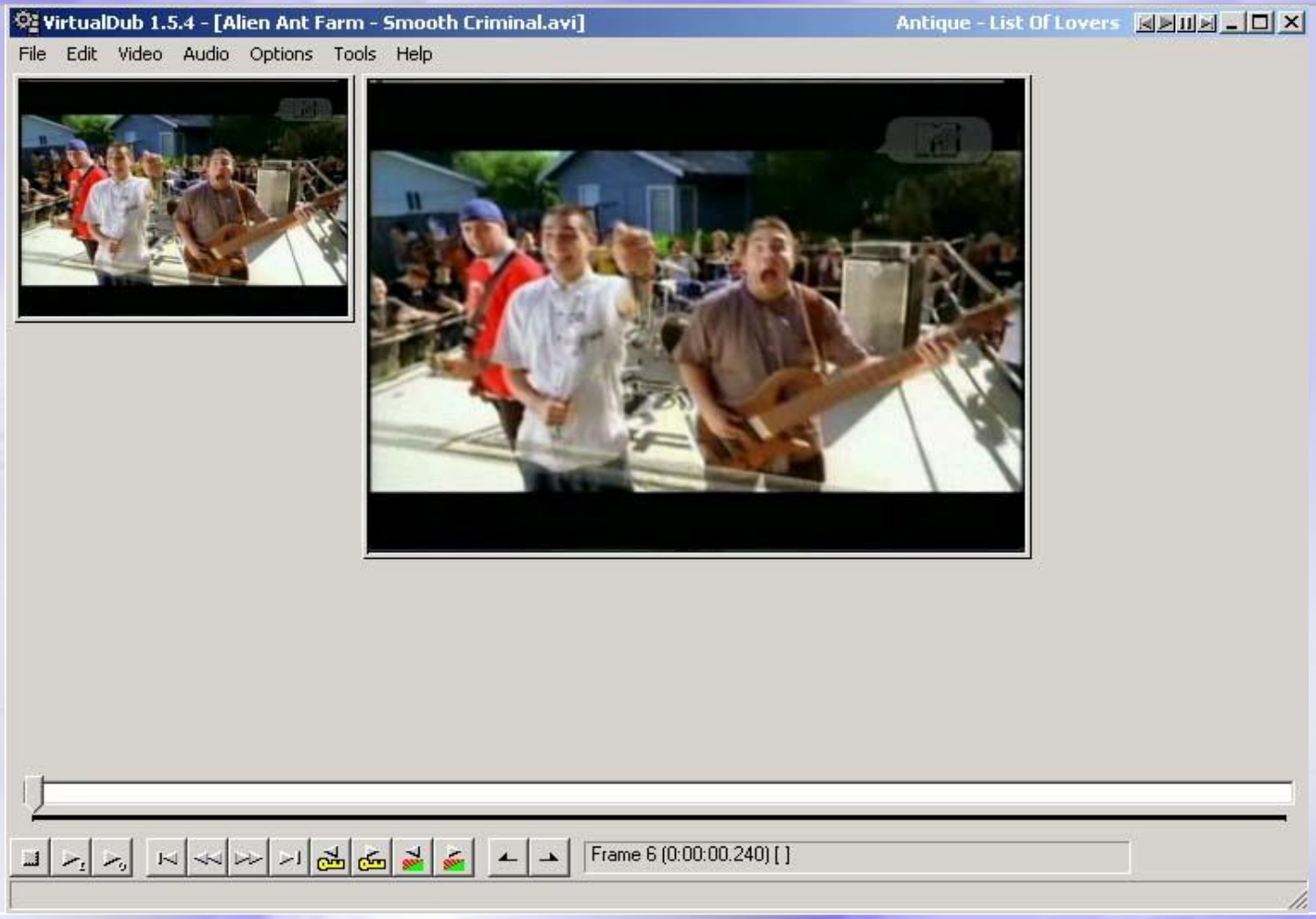
Область применения

GRAPHICS & MEDIA LAB
VIDEO GROUP

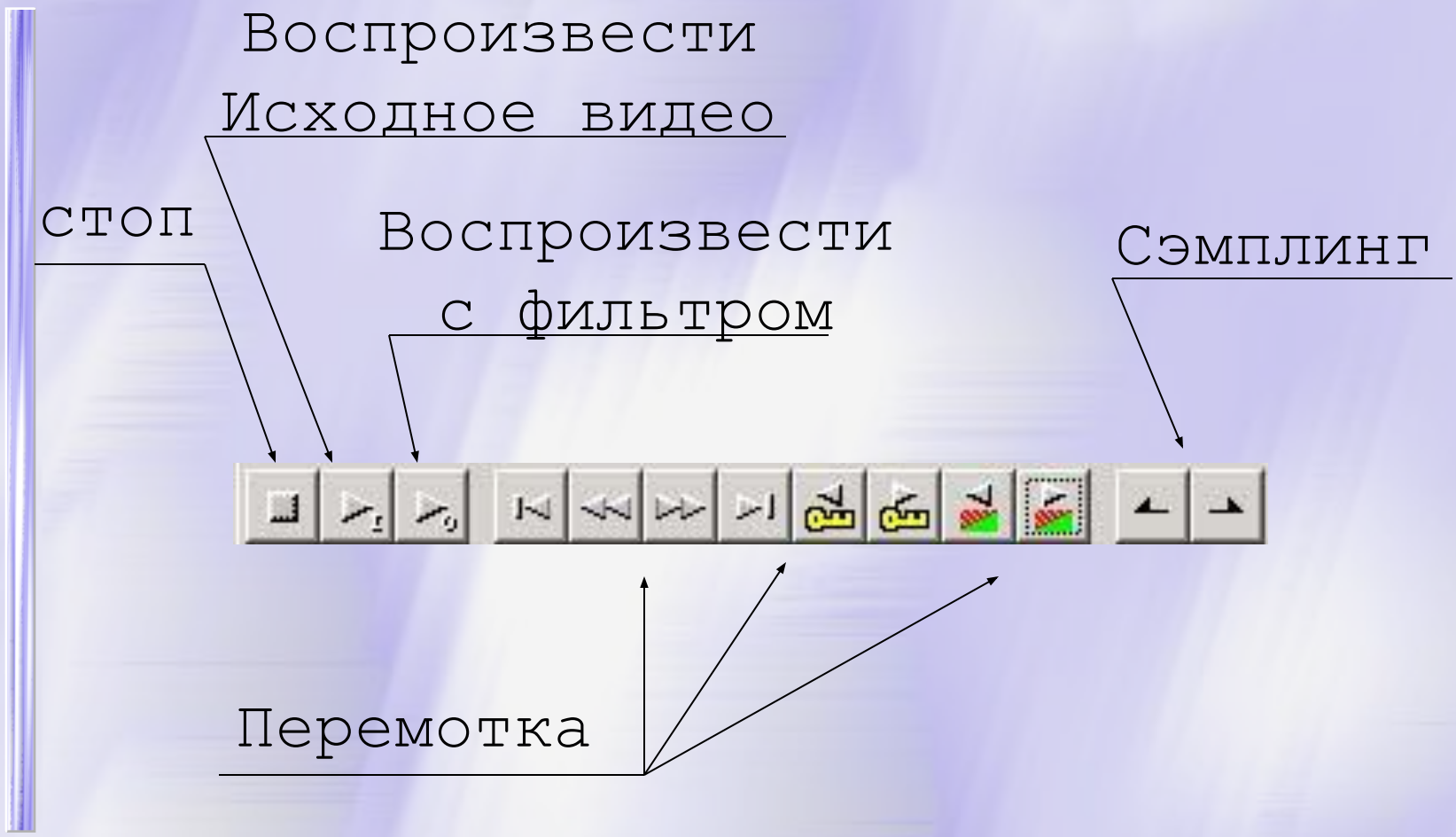
- Осуществляет просмотр и базовое редактирование видео
- Позволяет конвертировать в разные форматы видео и аудио треки
- Обрабатывает видео (и аудио) с использованием фильтров
- Осуществляет восстановление файлов
- Позволяет указывать в скрипте автоматическую обработку фильмов
- Осуществляет качественный захват видео с камеры

Внешний вид VirtualDub

GRAPHICS & MEDIA LAB
VIDEO GROUP

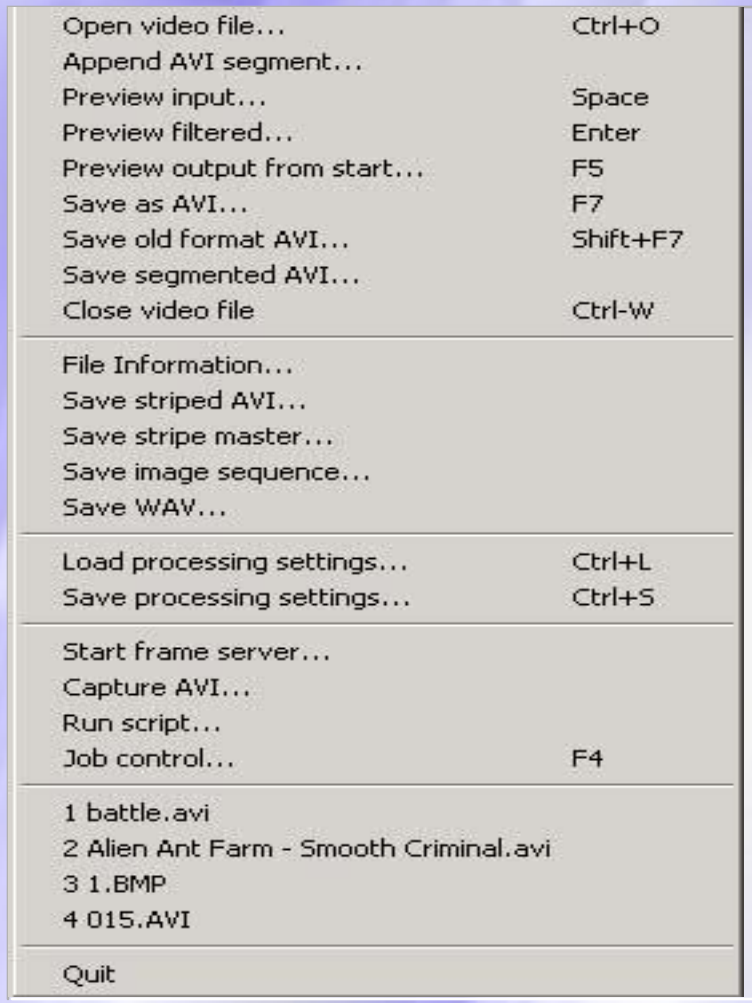


Панель управления



Меню File

GRAPHICS & MEDIA LAB
VIDEO GROUP

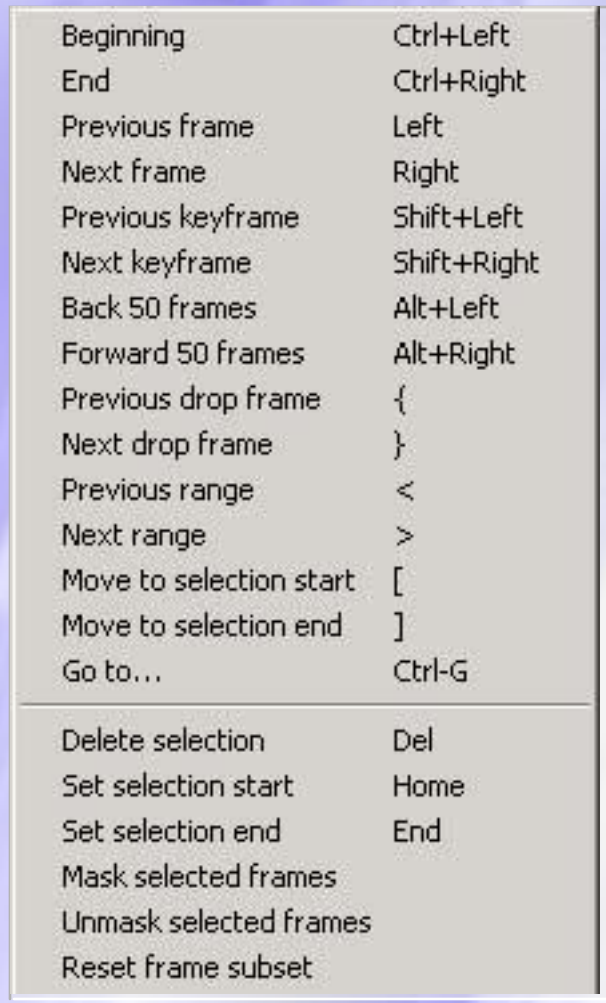


Работа с файлами: сохранение, загрузка, сохранение скриптов и т. п.

Работа Job – создание последовательных указаний для VirtualDub

Меню Edit

GRAPHICS & MEDIA LAB
VIDEO GROUP

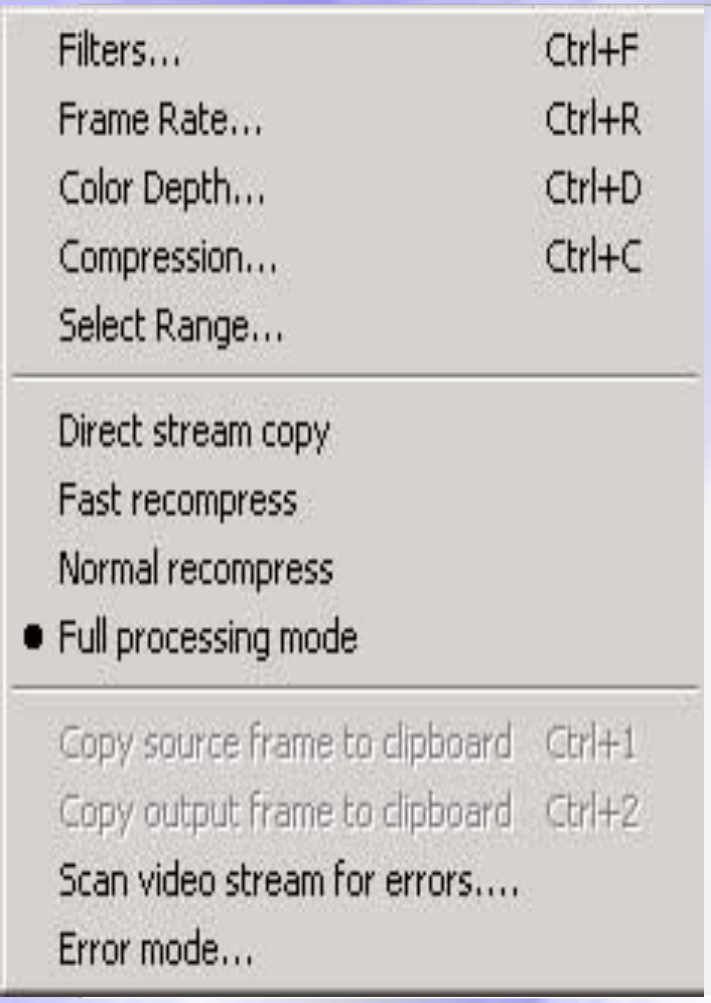


Более обширные возможности по перемотки видео вплоть до перехода на указанный номер кадра.

Работа с сэмплингами в расширенном режиме.

Меню Video

GRAPHICS & MEDIA LAB
VIDEO GROUP



Обработка видео с использованием фильтров. Подключение и указание параметров

Проверка на наличие ошибочных кадров в видео потоке.

Меню Audio

GRAPHICS & MEDIA LAB
VIDEO GROUP

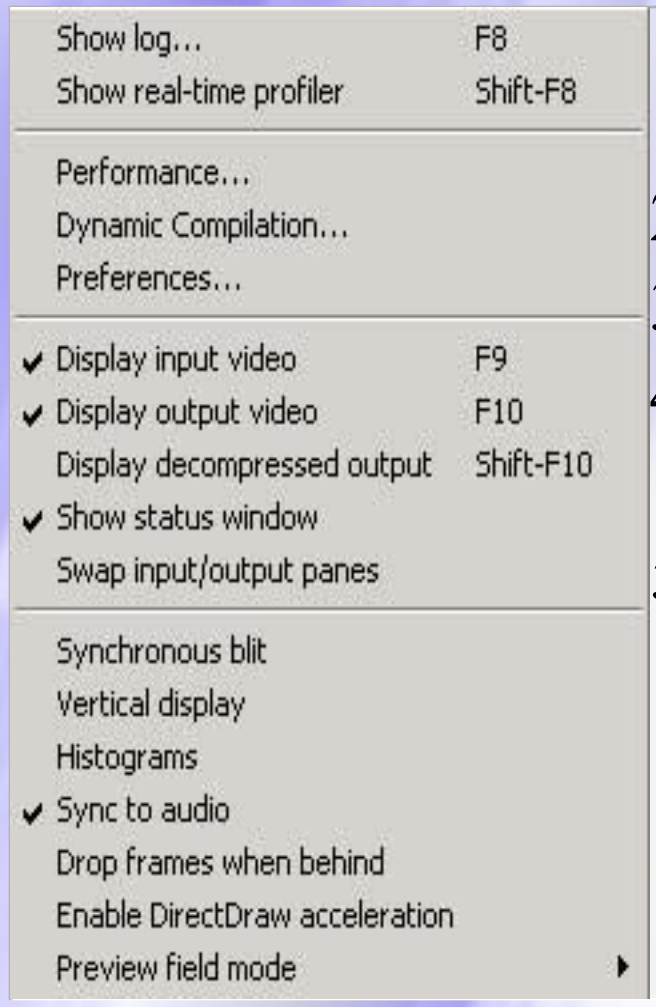
Установка параметров звука и фильтров которые будут подключены в момент обработки.

Установка режима:

- Прямое копирование потока
- Режим полной обработки

Меню Option

GRAPHICS & MEDIA LAB
VIDEO GROUP



1. Просмотр Log файлов
2. Установка параметров кодеков
3. Отображение потоков видео
4. Формат отображаемой информации
5. Расположения окон

Как писать фильтры для VirtualDub

GRAPHICS & MEDIA LAB
VIDEO GROUP

Фильтр для VirtualDub представляет собой DLL библиотеку которая имеет вид:

<имя>. vdf

После этого можно скопировать его в папку Plugins и подключить в программе как фильтр.

Структура файла *.vdf

GRAPHICS & MEDIA LAB
VIDEO GROUP

Файл должен содержать минимальный набор функций для работы. Те функции которые используются должны быть описаны в специальной структуре. Если функция не используется, то в поле должно стоять NULL.

Структура

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
typedef struct FilterDefinition{
   . . .
   FilterInitProc    initProc;
   FilterDeinitProc    deinitProc;
   FilterRunProc     runProc;
   FilterParamProc     paramProc;
   FilterConfigProc    configProc;
   FilterStringProc    stringProc;
   FilterStartProc     startProc;
   FilterEndProc     endProc;
   FilterScriptStrProc    fssProc;
   . . .
} FilterDefinition;
```

Пример

Разберем пример программы которая Blue компоненту уменьшает в два раза, а Green оставляет без изменения.

```
int runProc(const FilterActivation *fa,
const FilterFunctions *ff);
```

Это аналог процедуры main() в C++, VirtualDub начнет действия с вызова этой функции при обработке кадра (не учитываем вызов инетфейса).

RunProc

Фильтр, уменьшающий Blue в два раза

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
{   . . .
    src = (Pixel32 *)fa->src.data;
    dst = (Pixel32 *)fa->dst.data;
    h = fa->src.h;
    do {
        w = fa->src.w;
        do {   old_pixel = *src++;
            new_pixel = (old_pixel & 0xFF0000) +
    ((old_pixel & 0x0000FE)>>1) +            0x008000;
            *dst++ = new_pixel; }
        while(--w);
        src = (Pixel32 *)((char *)src +              fa   ->
src.modulo);
        dst = (Pixel32 *)((char *)dst +
        fa -> dst.modulo);
    }
    while(--h);
    return 0;
}
```

Служебные функции

Для VirtualDub надо включить 2 функции они служебные и не несут большой смысловой нагрузки, но их надо указывать для совместимости с VirtualDub:

```
extern "C" int __cdecl
   VirtualdubFilterModuleInit2(FilterModule *fm,
   const FilterFunctions *ff, int& vdfd_ver, int&
   vdfd_compat);
extern "C" void __cdecl
   VirtualdubFilterModuleDeinit(FilterModule *fm,
   const FilterFunctions *ff);
```

Описание структуры

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
struct FilterDefinition {
NULL,NULL,NULL,   // next, prev
"tutorial",  // name
"blue color",   // desc
"anyone",     // maker
NULL,        // private_data
0,         // inst_data_size
. . .
```

Описание атрибутов относящиеся к интерфейсу фильтра, подсказка для пользователя при подключении фильтра.

Описание структуры

GRAPHICS & MEDIA LAB
VIDEO GROUP

//Описание используемых функций.

```
. . .
NULL,           // initProc
NULL,           // deinitProc
tutorialRunProc,  // runProc
NULL,           // paramProc
NULL,           // configProc
NULL,           // stringProc
NULL,           // startProc
NULL,           // endProc
NULL,           // script_obj
NULL,           // fssProc };
```

Компиляция

GRAPHICS & MEDIA LAB
VIDEO GROUP

- После компиляции файл будет иметь вид *.dll
- Переименовать файл в *.vdf (мы создавали проект для написания DLL библиотеки)
- Поместить его в папку Plugins
- Подключить в опциях VirtualDub фильтр

Итоги

GRAPHICS & MEDIA LAB
VIDEO GROUP

Плюсы:

- Программа является бесплатной и с открытыми исходниками
- Все проблемы с открытием видео и его сохранением уже решены
- Возможность загрузки фильтров
- Программирование ведется на C++ что позволяет пользоваться весьма гибким аппаратом для реализации своих алгоритмов

Содержание:

GRAPHICS & MEDIA LAB
VIDEO GROUP

- MMX технология
- Программа VirtualDub
- Программа AviSynth
- Программа Mathcad

AviSynth

Скриптовый язык потоковой обработки видео

План

GRAPHICS & MEDIA LAB
VIDEO GROUP

- ✔ О программе AviSynth
- ✔ Операторы
- ✔ Семантика и прагматика операций:
  - логические
  - математические
- ✔ Классификаций функций AviSynth
- ✔ Использование Plugins
- ✔ Итоги

Что это такое?

GRAPHICS & MEDIA LAB
VIDEO GROUP

AviSynth является программой-источником (FrameServer) которую используют различные приложения для обработки видео.

Также AviSynth обладает развитым скриптовым языком и механизмом Plug-In, позволяющим в потоке обрабатывать фильмы.

Как работает AviSynth

GRAPHICS & MEDIA LAB
VIDEO GROUP

Использование AviSynth состоит из двух этапов:

- создается простой текстовый документ который содержит последовательность команд – скрипт;
- запускается приложением обрабатывающее видео, например, можно запустить его VirtualDub или Windows Media Player.

Почему удобен AviSynth

GRAPHICS & MEDIA LAB
VIDEO GROUP

AviSynth является открытым и свободно распространяющимся проектом. Исходники можно исправлять и вносить в них те изменения, которые вам нужны. Этот проект только стартовал, и есть уникальная возможность поучаствовать в нём.

Где можно скачать и узнать о AviSynth

GRAPHICS & MEDIA LAB
VIDEO GROUP

Официальный сайт AviSynth:

http://www.avisynth.org/

Здесь можно вносить свои предложения, а также предложена весьма интересная идея – корректировать сайт вместе с разработчиками.

Типы доступные в AviSynth

GRAPHICS & MEDIA LAB
VIDEO GROUP

Семантика типов данных	Прагматика
`clip`	Переменная хранящая параметры видео/аудио клипа
`string`	Строковая переменная
`int`	Целочисленная переменная
`float`	Переменная с плавающей точкой
`bool`	Булевская переменная

Логические операции AviSynth

GRAPHICS & MEDIA LAB
VIDEO GROUP

Семантика	Прагматика
==	Равенство
!=	Неравно
||	Логическое Или (OR)
&&	Логическое И (AND)

Логические операции AviSynth

GRAPHICS & MEDIA LAB
VIDEO GROUP

Семантика	Прагматика
+	Сложение
-	Вычитание
*	Умножение
/	Деление
%	Mod (Операция в кольце)
>=(<=)	Больше/меньше или равно
>	Больше
<	Меньше

Пример

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
a = 20 - 5 - 5
```

Результатом является `10`

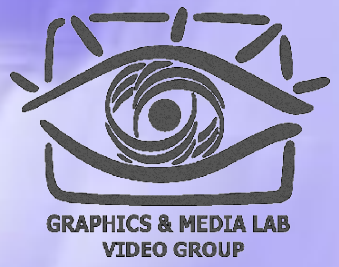
```
b = (a==true) ? 1 : 2
```

Аналог операции в C++ «:?»

В эквивалентном виде:

```
if (a==true) then b=1 else
b=2
```

Классификация функций в AviSynth



1. Численные функции
2. Строковые функции
3. Функции перевода
4. Проверочные функции
5. Другие типы функций

Остановимся на некоторых из их подробнее. Все эти функции а также многие другие приведены на сайте.

Численные функции

GRAPHICS & MEDIA LAB
VIDEO GROUP

`Floor (float)` Переводит `float` в `int` до ближайшего снизу

```
Floor(1.2) = 1
Floor(1.6) = 1
Floor(-1.2) = -2
Floor(-1.6) = -2
```

Численные функции

GRAPHICS & MEDIA LAB
VIDEO GROUP

`Round (float)` Переводит `float` в `int` округляя результат

```
Round(1.2) = 1
Round(1.6) = 2
Round(-1.2) = -1
Round(-1.6) = -2
```

Численные функции

GRAPHICS & MEDIA LAB
VIDEO GROUP

Стандартные математические функции:

```
Sin (float)
Cos (float)
Pi ()
Log (float)
Exp (float)
Pow (float base, float power)
Sqrt (float)
```

Численные функции


```
Spline (float X,  x1,y1, x2,y2, ....,
bool "cubic")
```

Пример:

```
Spline(5, 0,0, 10,10, 20,0, false) = 5
Spline(5, 0,0, 10,10, 20,0, true) = 7
```

Работа со строками

GRAPHICS & MEDIA LAB
VIDEO GROUP

`UCase("AviSynth")`	Буквы в верхний регистр
`LCase("AviSynth")`	Буквы в нижний регистр
`RevStr("AviSynth")`	Инверсия букв
`StrLen("AviSynth")`	Длина строки

Пример

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
UCase("AviSynth") = "AVISYNTH"

LCase("AviSynth") = "avisynth"

RevStr("AviSynth") = "htnySivA"

StrLen("AviSynth") = 8
```

Функция перевода

GRAPHICS & MEDIA LAB
VIDEO GROUP

`Value(string)`- Переводит строку в `int`

Пример:

```
Value( "-2.7" ) = -2.7
```

Функции проверки типа

Функции проверки типа переменных:

```
IsBool (var)
IsInt (var)
IsFloat (var)
IsString (var)
IsClip (var)
```

Пример

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
a = AVISource("d:\capture.00.avi")
b = AVISource("d:\capture.01.avi")
c = AVISource("d:\capture.02.avi")
sound_track=AVSource("d:\audio.wav")
AudioDub(a+b+c, sound_track)
```

В переменные a,b,c записываются параметры трех видео роликов. Которые потом будут показаны последовательно друг за другом с общей звуковой дорожкой которую мы тоже предварительно загрузили.

Фильтры

В AviSynth можно использовать фильтры написанные уже раньше. Для этого их надо подключить специальной функцией:

```
LoadPlugin ("filename"[,...])
```

Фильтры в AviSynth имеют вид:

<имя>.avs

Фильтры

GRAPHICS & MEDIA LAB
VIDEO GROUP

AviSynth позволяет подключить фильтры написанные в VirtualDub. Единственное ограничение – необходимый формат видео RGB32. Но если будет другой формат можно воспользоваться встроенными в AviSynth функциями для перевода в нужный формат.

Использование фильтров VirtualDub

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
LoadVirtualDubPlugin
("filename","filtername", preroll)
```

Подключение фильтра VirtualDub:

1. Первый параметр <имя>.vdf плагин VirtualDub
2. Второй параметр имя файла с конвертированного по AviSynth (<имя>.avs)
3. Preroll показывает сколько кадров необходимо держать в буфере (например, для деинтерлейсинга)

Пример

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
Import("d:\vdub_filters.avs")
AviSource("d:\filename.avi")
ConvertToRGB32() # Там где надо
VD_SmartBob(1, 0, 10, 1)
ConvertBackToYUY2() # Там где надо
```

VD_SmartBob(1, 0, 10, 1) – функция осуществляющая подгрузку плагина с соответствующими параметрами для данного фильтра.

Загрузка фильтра деинтерлейсинга

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
function  VD_SmartBob(clip  ''clip'',  bool
   ''show_motion'', int ''threshold'', bool
   ''motion_map_denoising'')
{
   LoadVirtualdubPlugin("d:\bob.vdf","_VD_Sm
   artBob", 1)
   Return
   clip.SeparateFields._VD_SmartBob(clp.GetP
   arity?1:0,default(show_motion,false)?1:0,
   default(threshold,10),
   default(motion_map_denoising,true)?1:0)
}
```

Итоги

GRAPHICS & MEDIA LAB
VIDEO GROUP

AviSynth является весьма гибким средством применительно к любому приложению работающему с видео. Возможность работы с исходниками позволяет получить при правильном подходе весьма ощутимые результаты.

Основное преимущество – возможность СУЩЕСТВЕННО сэкономить время при массовых операциях с фильмами.

Содержание:

GRAPHICS & MEDIA LAB
VIDEO GROUP

- MMX технология
- Программа VirtualDub
- Программа AviSynth
- Программа Mathcad

7

Mathcad

Удобнейшее средство визуализации данных.
Средство предварительной проработки фильтров.

Введение в Mathcad
Достоинства mathcad’a

GRAPHICS & MEDIA LAB
VIDEO GROUP

Почему стоит использовать Mathcad:

- Промежуток времени для получения первых результатов работы алгоритма значительно меньше по сравнению с разработкой в какой-либо среде
- Каждое изменение текста программы динамически влияет на результат
- Поиск ошибок осуществляется быстрее, чем в исходном тексте программы на к-л. языке программирования
- Реализовав основную часть алгоритма в mathcad’e, время написания реальной программы уменьшается на порядок
- **Множество** реализованных, готовых к использованию функций

Введение в Mathcad (2)
Достоинства mathcad’a

GRAPHICS & MEDIA LAB
VIDEO GROUP

Почему стоит использовать Mathcad:

- Исходный код программы выводится в графическом режиме, и потому выглядит нагляднее, чем в текстовом редакторе
- Как правило, для реализации к-л. задачи в mathcad’е требуется написать меньше исходного теста, чем например в C++
- Реализовав основную часть алгоритма в mathcad’е, время написания реальной программы уменьшается на порядок
- Отличная помощь: все описано кратко и понятно
- Простота использования
- Индексация в массиве начинается с нуля

Пример функции чтения Сбоку - изображение с NEDI

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
LoadC2(A, i, j, M) := | m ← M − 1
                      | str ← 0
                      | for d ∈ 0..m
                      |     i0 ← i + d
                      |     j0 ← j − m + d
                      |     for k ∈ 0..m
                      |         C[str, 0] ← A[i0−k, j0+k−2]
                      |         C[str, 1] ← A[i0−k−2, j0+k]
                      |         C[str, 2] ← A[i0−k+2, j0+k]
                      |         C[str, 3] ← A[i0−k, j0+k+2]
                      |         str ← str + 1
                      | C
```

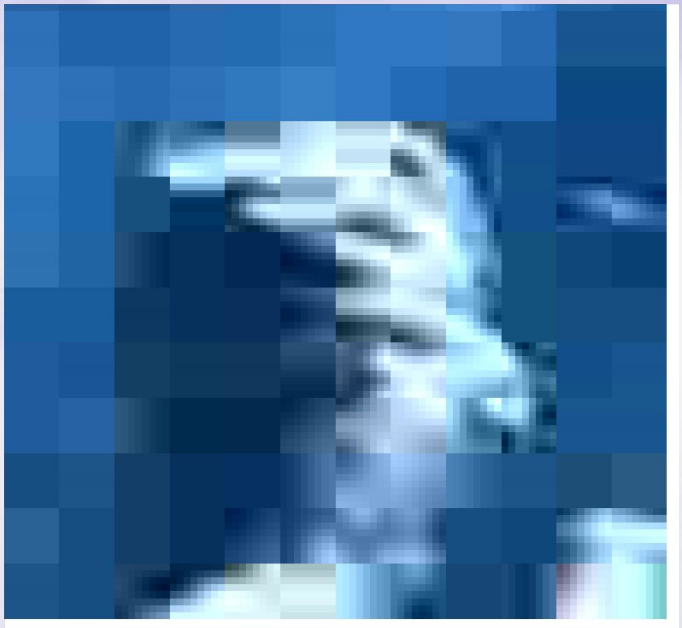
R2, G2, B2

Разработка фильтра
Чтение изображения

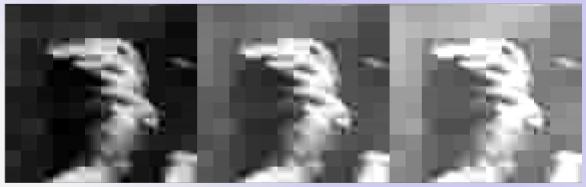
GRAPHICS & MEDIA LAB
VIDEO GROUP

Визуализация матрицы – Ctrl + T

S := "F:\Doklad\battle_100_59_blocked.bmp " P := READRGB(S)



S

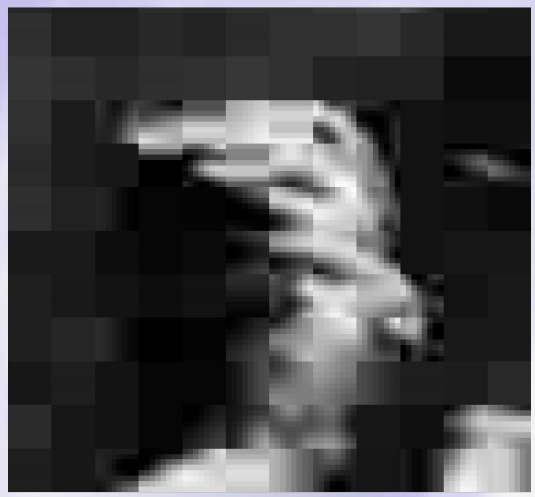


$$\text{DeTriplex}(P)^{P} := \left| \begin{array}{l} w \leftarrow \frac{\text{cols}(P)}{3} \\ R_0 \leftarrow \text{submatrix}(P, 0, \text{rows}(P) - 1, 0, w - 1) \\ R_1 \leftarrow \text{submatrix}(P, 0, \text{rows}(P) - 1, w, 2 \cdot w - 1) \\ R_2 \leftarrow \text{submatrix}(P, 0, \text{rows}(P) - 1, 2 \cdot w, 3 \cdot w - 1) \\ R \end{array} \right.$$

Разработка фильтра
Показ компонент изображения

GRAPHICS & MEDIA LAB
VIDEO GROUP

RGB:= DeTriplex(P)



RGB_0

RGB_1

RGB_2

Разработка фильтра
Перевод изображения в YUV

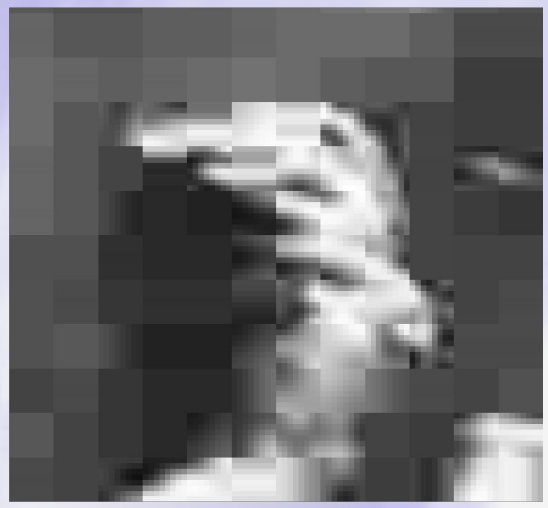
GRAPHICS & MEDIA LAB
VIDEO GROUP

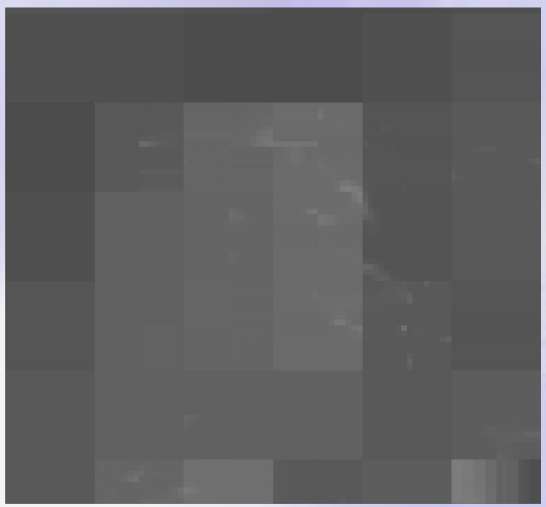
$$
\begin{aligned}
&\mathrm{RGB\_to\_YUV}(RGB) := \\
&\quad R \leftarrow RGB_0 \\
&\quad G \leftarrow RGB_1 \\
&\quad B \leftarrow RGB_2 \\
&\quad YUV_0 \leftarrow \mathrm{Clip}(0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B) \\
&\quad YUV_1 \leftarrow \mathrm{Clip}(-0.147 \cdot R - 0.289 \cdot G + 0.436 \cdot B + 128) \\
&\quad YUV_2 \leftarrow \mathrm{Clip}(0.615 \cdot R - 0.515 \cdot G - 0.1 \cdot B + 128) \\
&\quad YUV
\end{aligned}
$$

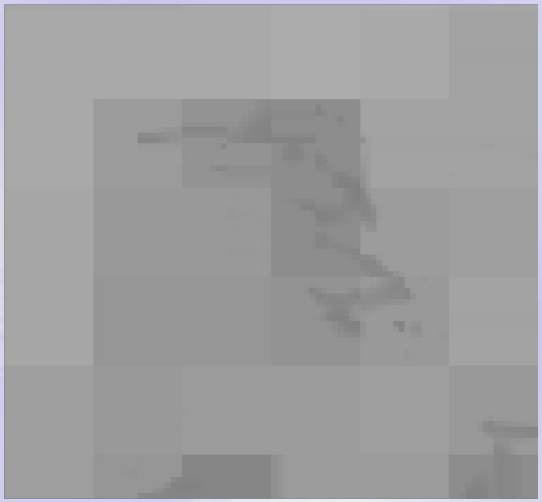
Разработка фильтра
Показ результата

GRAPHICS & MEDIA LAB
VIDEO GROUP

$YUV := RGB\_to\_YUV(RGB)$







YUV_0 YUV_2 YUV_1