GRAPHICS & MEDIA LAB
VIDEO GROUP

Декодирование видео на современных видеокартах

Арсаев Марат

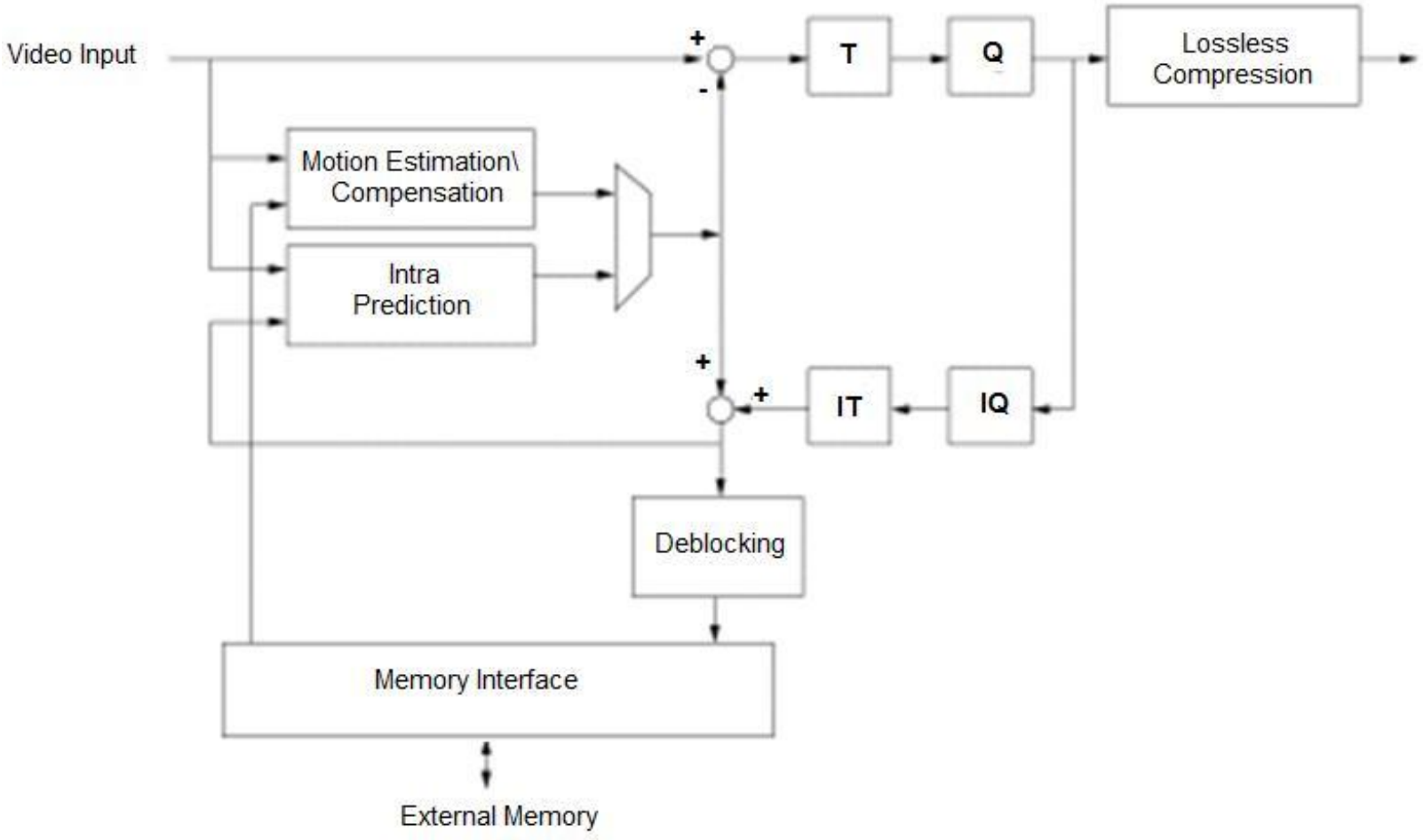
Video Group
CS MSU Graphics & Media Lab

19.05.2010

Содержание

- **Введение**
- Интерфейсы обработки видео
- Аппаратная поддержка
- Декодирование на CUDA
- Выводы

Кодирование видео

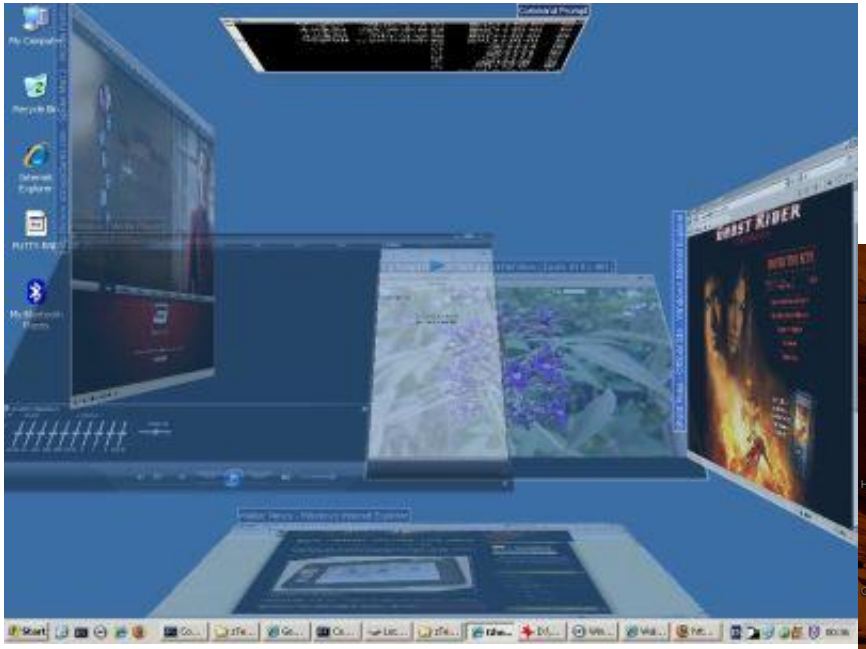


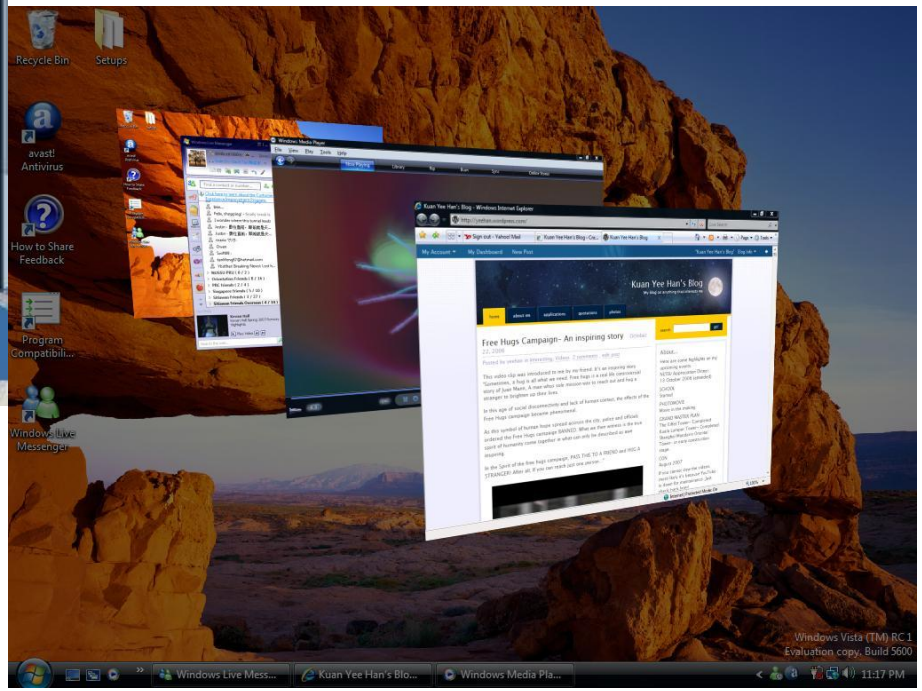
Содержание

- Введение
- Интерфейсы обработки видео
  - **Microsoft DirectX Video Acceleration**
  - Аналоги для Linux
- Аппаратная поддержка
- Декодирование на CUDA
- Выводы

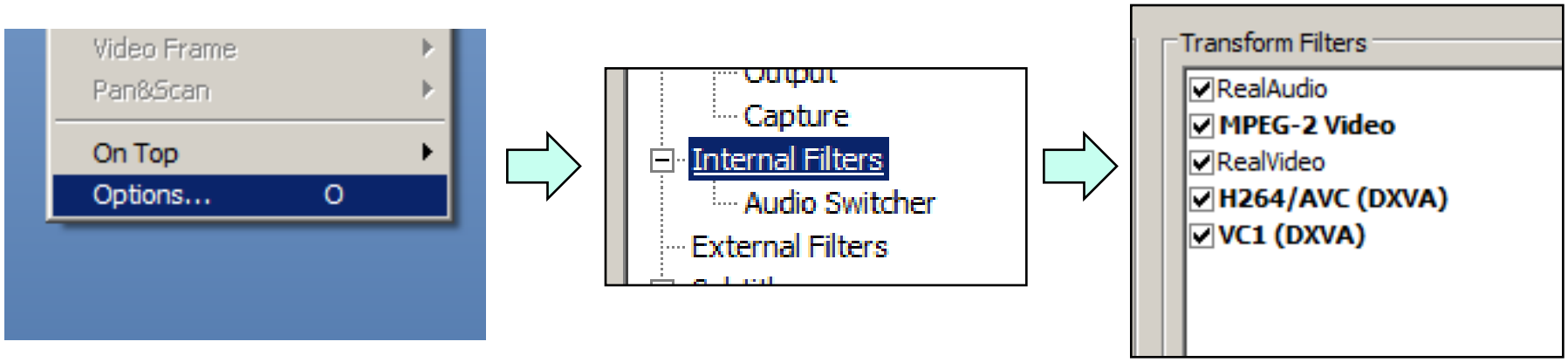
Microsoft DirectX Video Acceleration

GRAPHICS & MEDIA LAB
VIDEO GROUP

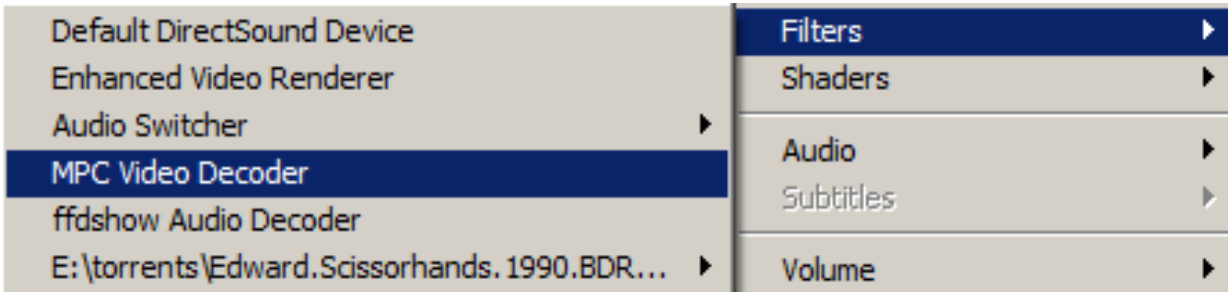


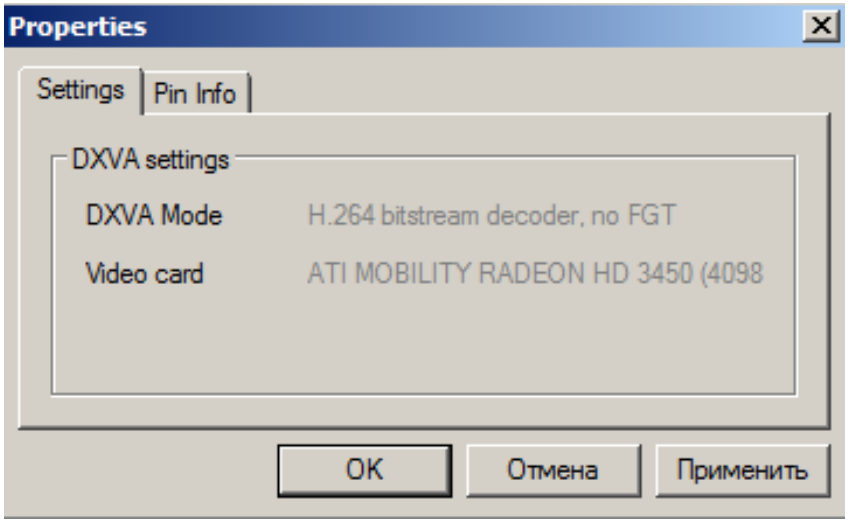


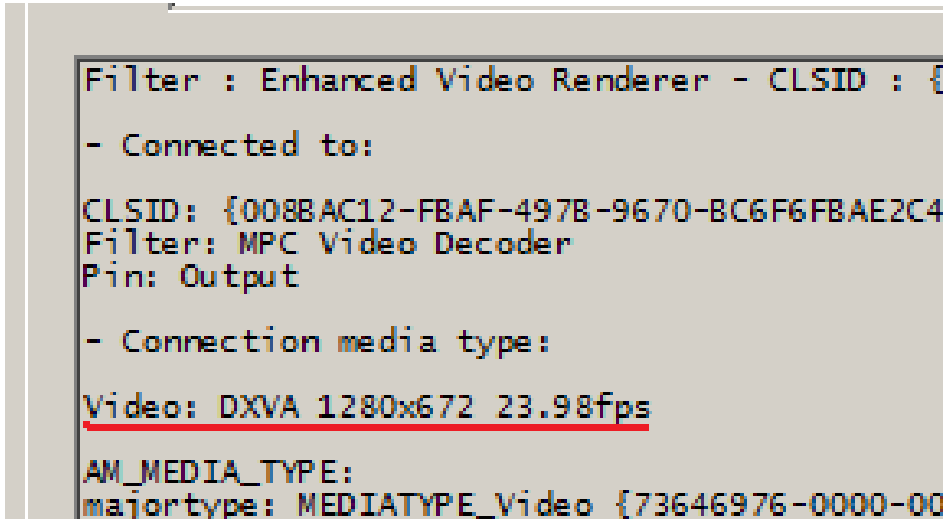
Microsoft DirectX Video Acceleration: MPC-HC



Microsoft DirectX Video Acceleration: MPC-HC







Microsoft DirectX Video Acceleration

GRAPHICS & MEDIA LAB
VIDEO GROUP

Состоит из Device Driver Interface:

- Motion Compensation DDI (decoder DDI)
- ProcAmp DDI (post-processing DDI)
- Deinterlacing DDI
- COPP DDI (security DDI)

Вышла в декабре 2002 года.

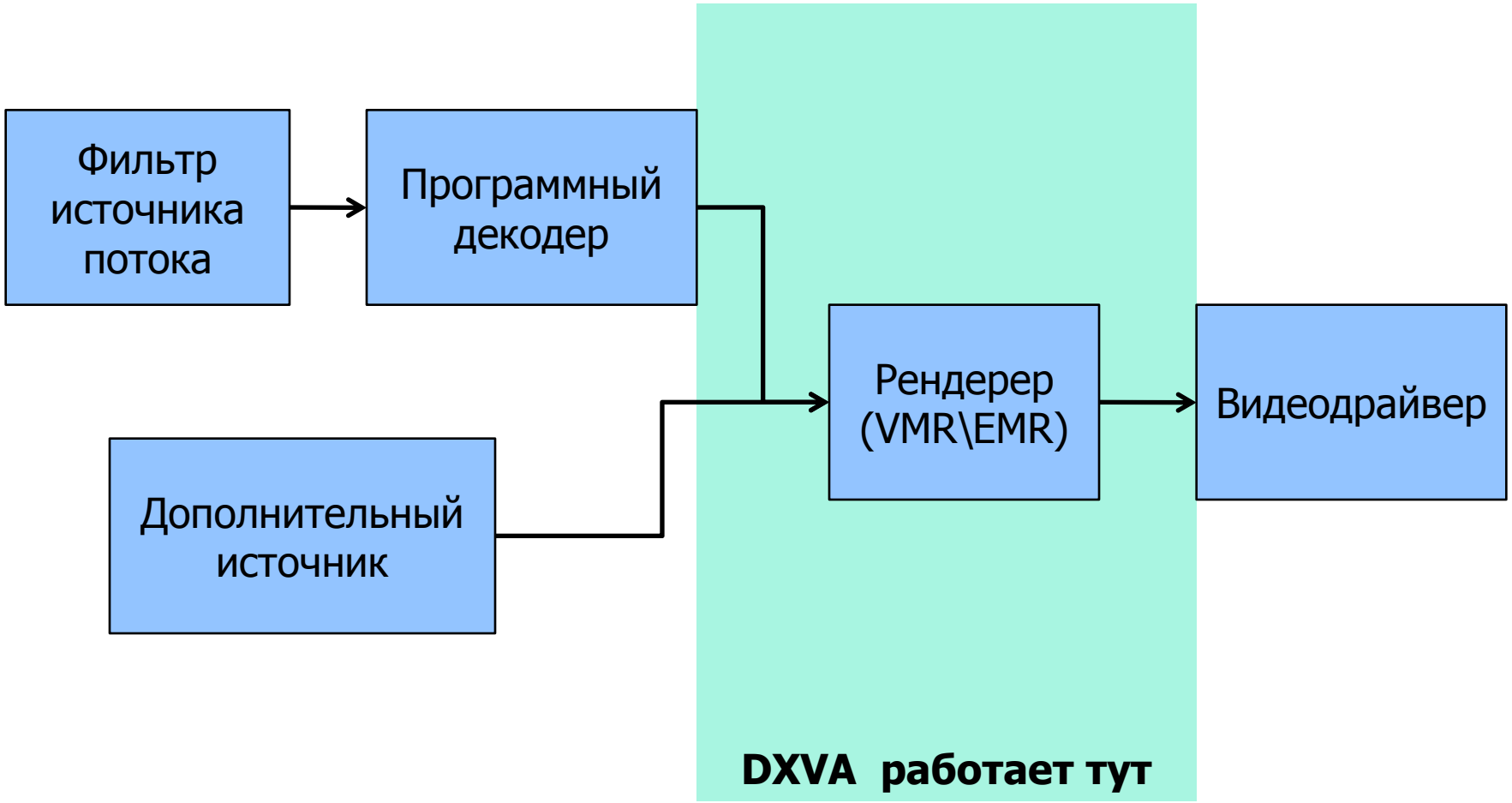
Microsoft DirectX Video Acceleration

GRAPHICS & MEDIA LAB
VIDEO GROUP

Для того чтобы DXVA заработал, необходимо:

- Программный декодер с необходимыми DXVA вызовами
- Системный DXVA-драйвер (входит в DirectX)
- Видео драйвер, поддерживающий запрашиваемые вызовы
- Видеокарту с аппаратной поддержкой требуемых функций

Поток данных DirectShow



Decoder DDI

Поддерживаемые стандарты: MPEG-1, MPEG-2, MPEG-4, H.263, H.264, VC-1

Будут ускорены аппаратно только те части декодирования, которые мы сконфигурируем

Deinterlace DDI

Доступные алгоритмы деинтерлейсинга:

- Bob (line doubling)
- Simple Switching Adaptive
- Motion Vector Steered
- Advanced 3D Adaptive

Deinterlace DDI

None

Bob

ATI Vector Adaptive

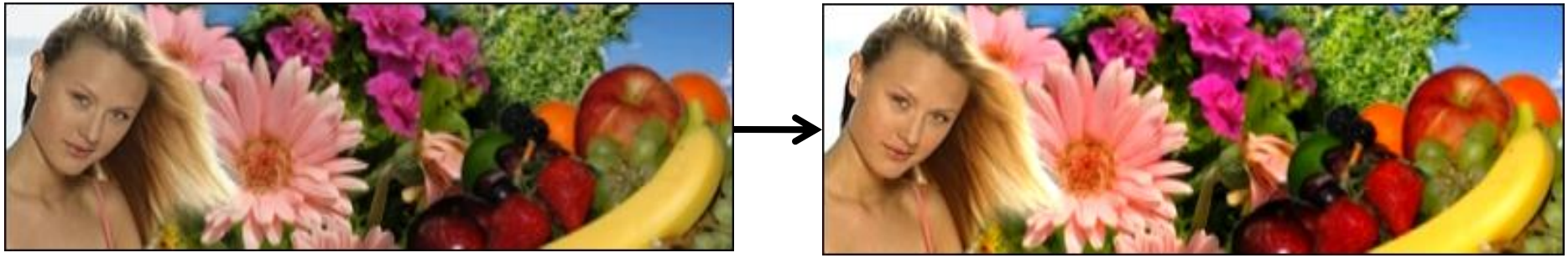
Deinterlace DDI

Доступные алгоритмы преобразования частоты кадров:

- Frame Repeat/Drop
- Linear Temporal Interpolation
- Motion Vector Steered

ProcAmp DDI

- Предоставляет возможность улучшить визуальное качество
- Предоставляет возможность автоматической регулировки параметров видеокартой



Использование Microsoft DXVA

Работа с DXVA идет через фильтры DirectShow и состоит из следующих шагов:

- Запрос возможностей устройства (от определенного типа видео)
- Подготовка рендерера
- Выделение буферов в видеопамяти
- Выполнение

Microsoft DXVA 2.0

- Доступен, начиная с DirectX 10 (2006 год)
- Новый интерфейс – Capture DDI
- Позволяет обходить ограничения на DirectShow pipeline
- Использует Enhanced Video Renderer (EVR)
- Дополнительный стандарт – DXVA-HD

Microsoft DXVA: резюме

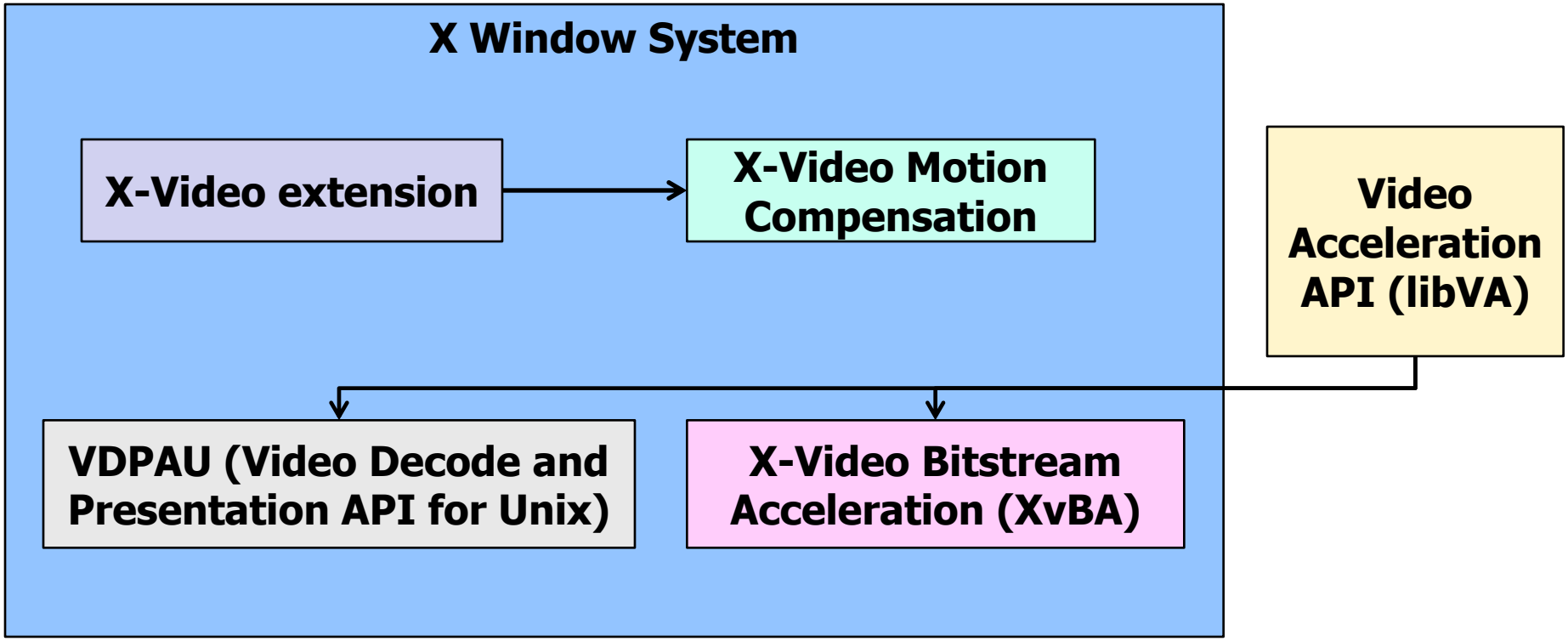
GRAPHICS & MEDIA LAB
VIDEO GROUP

+ Хорошо описанный интерфейс с множеством форматов
+ Активная поддержка со стороны производителей железа и ПО (Adobe Flash, CoreAVC, ffdshow, Media Player Classic, и т.д.)
- Довольно сложное программирование
- Привязан к Microsoft DirectX

Содержание

- Введение
- Интерфейсы обработки видео
  - Microsoft DirectX Video Acceleration
  - **Аналоги для Linux**
- Аппаратная поддержка
- Декодирование на CUDA
- Выводы

Иерархия расширений



X-Video extension

Расширение отвечает за проигрывание и вывод видео в системе X Window (с 1991 года)

- Поддержка framebuffer object
- Возможность вывода на удаленный X Window сервер
- Возможность аппаратного линейного скейлинга, конвертирования форматов

X-Video Motion Compensation

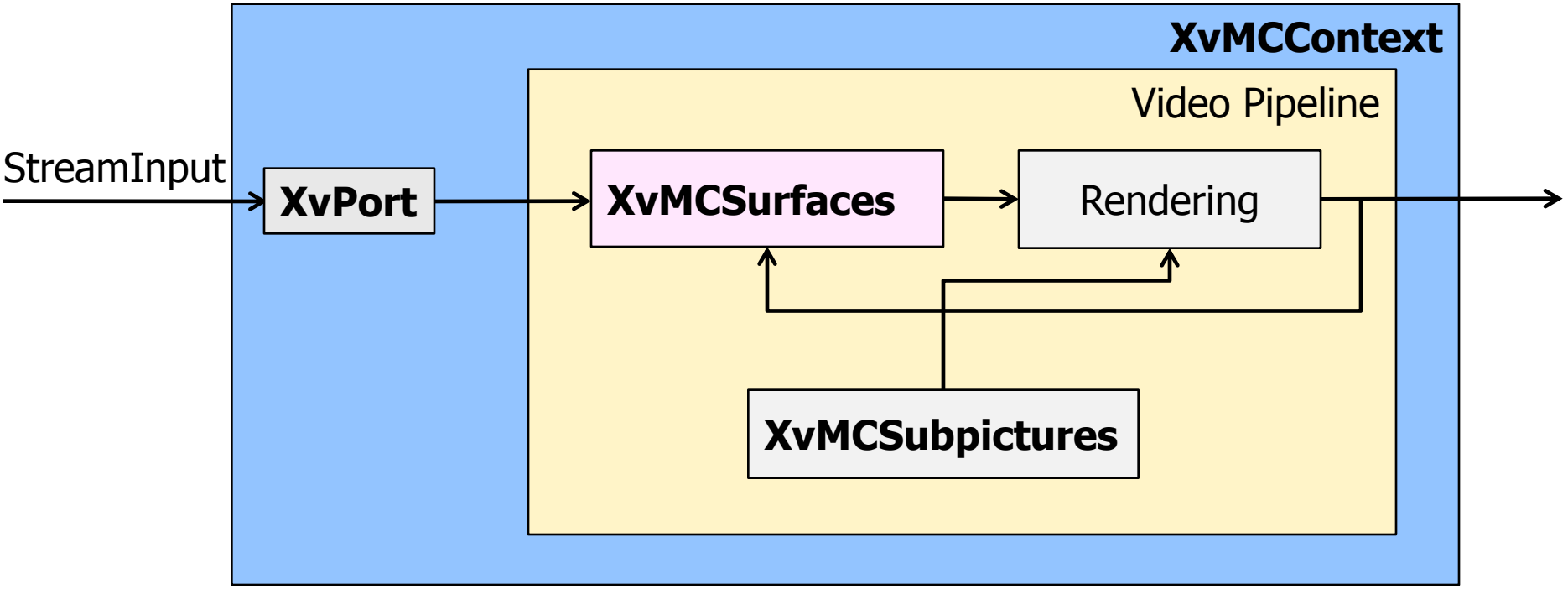
GRAPHICS & MEDIA LAB
VIDEO GROUP

Частичное добавление аппаратного декодирования для MPEG-1, MPEG-2, MPEG-4:

- Motion compensation
- iDCT
- Variable Length Encoding (VLD)

Поддержка множества видеокарт – NVIDIA, VIA, Intel, S3 с 2002 года. ATI сделали поддержку совсем недавно

X-Video Motion Compensation



Video Acceleration API

GRAPHICS & MEDIA LAB
VIDEO GROUP

Создана для замещения XvMC с использованием последних возможностей GPU

Возможности:

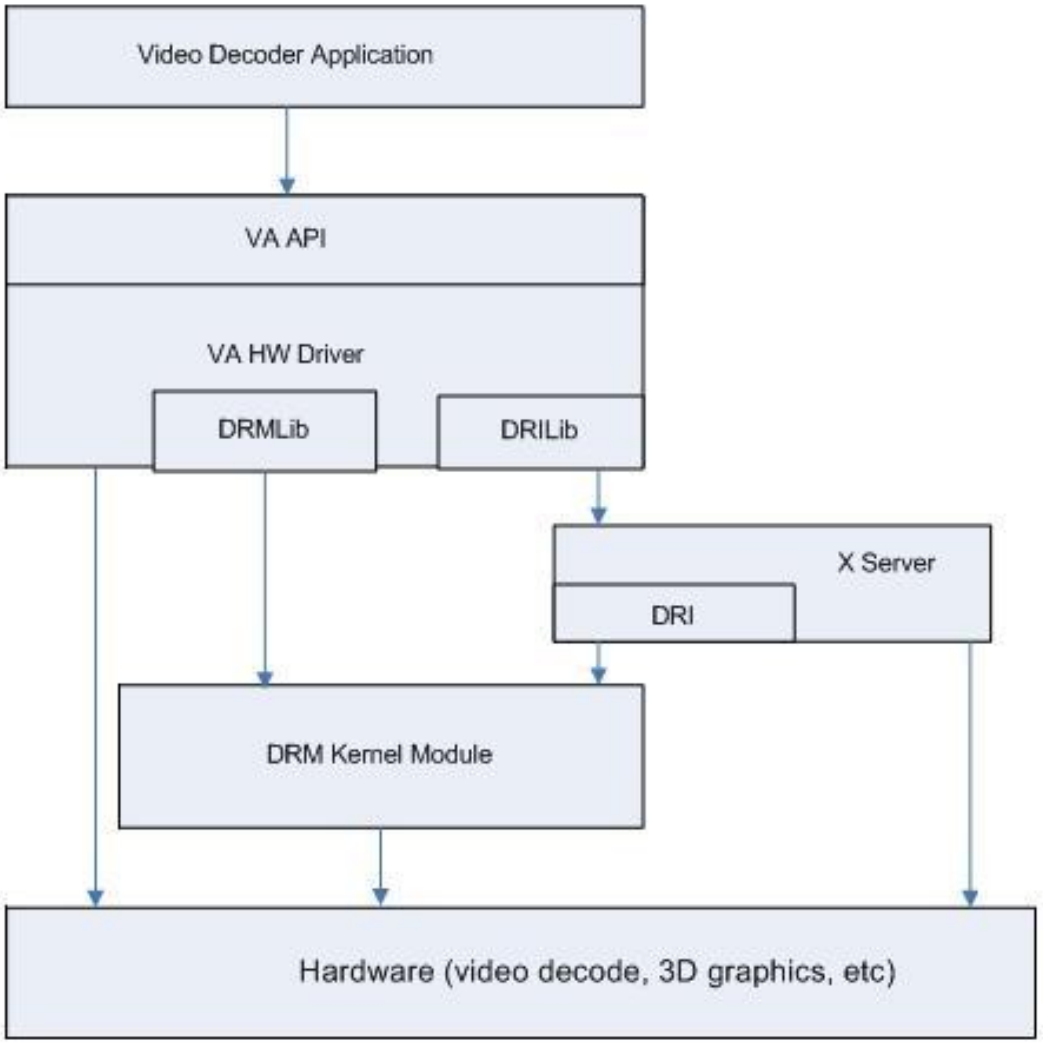
- Motion compensation
- iDCT (+ Modified iDCT)
- In-loop Deblocking filter
- Intra-frame prediction
- Обратное квантование
- Variable Length Decoding
- Пространственно-временной деинтерлейсинг
- Context-adaptive binary arithmetic coding / Context-adaptive variable-length coding

Для видео в формате MPEG-2, H.264, VC-1

Video Acceleration API

Поддерживается обмен данными с X Window сервером через Direct Rendering Infrastructure

Благодаря удобному интерфейсу и расширенным возможностям получил развитие от NVIDIA и ATI



Video Acceleration API: Compiz

Video Decode and Presentation API for Unix

- Открытая библиотека (libvdpau) - back-end VAAPI от Nvidia (2007 год)
- Распространяется и в виде библиотеки, отдельной от драйверов
- Поддерживается некоторыми видеокартами S3
- Три поколения с наращиваемым функционалом

Video Decode and Presentation API for Unix

GRAPHICS & MEDIA LAB
VIDEO GROUP

Реализует MC, iDCT, VLD, Deblocking, Deinterlacing для стандартов

- MPEG-1
- MPEG-2
- MPEG-4
- H.264
- VC-1

Во второе поколение в стандарт добавлены подавление шума и шарпен

X-Video Bitstream Acceleration

- Back-end для VAAPI от ATI (2009 год)
- Поддержка MPEG-2, H.264, VC-1
- Поддержка для ограниченного набора GPU (начиная с 4xxx)
- Поддержка базовых операций:
  - Motion compensation
  - iDCT
  - VLD

Сравнение декодеров

	Supported formats	Supported operation
DXVA	MPEG, H.264, VC-1	basic, FRC, DI, CE, IQ, encryption
Xv	-	rescaling, format conversion
XvMC	MPEG, H.264	basic, DI
VA API	MPEG, H.264, VC-1	basic, DI, FRC, IQ
VDPAU	MPEG, H.264, VC-1	basic, DI, denoising, sharpen
XvBA	H.264, VC-1	basic, DI

basic – Bit stream decoding
FRC – Frame Rate Conversion
CE – Color Enhancement
DI – deinterlacing
IQ – inverted quantization

Содержание

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Введение
- Интерфейсы обработки видео
- Аппаратная поддержка
  - **Nvidia PureVideo**
  - ATI AVIVO & Unified Video Decoder
- Декодирование на CUDA
- Выводы

Nvidia PureVideo

- Помогает в декодировании и пост-обработке видео
- Доступна с GeForce 6xxx
- Технология используется практически во всех программных DVD/HD-плеерах
- Продается отдельный DirectShow декодер – Nvidia PureVideo Decoder

Nvidia PureVideo 1st gen

Опубликован для видеокарт серии 6xxx (2004 год):

- использовался VMR9
- улучшено визуальное качество деинтерлейсинга и рескейлинга
- использовался в части MPEG-1/MPEG-2 decoding pipeline
- начиная с 6600 (2005 год) – полная акселерация MPEG-1/MPEG-2 decoding pipeline, ограниченная поддержка VC-1 и H.264

Nvidia PureVideo 2nd gen HD

Начиная с видеокарт серии 8xxx (2007):

- переделанный H.264 pipeline полностью на GPU
- большая часть декодера VC-1 на GPU
- технология позволяла среднему компьютеру проигрывать HD-DVD и Blu-ray фильмы без задержек

Nvidia PureVideo 3rd gen HD

На некоторых картах 8-ой и 9-ой серий (2008):

- полностью поддеживаемый декодинг VC-1
- минорные изменения в декодировании MPEG-2
- соответствует поддержке Video Decode and Presentation API for Unix feature set B

Nvidia PureVideo 4th gen HD

На картах 3xx и некоторых 2xx (2009):

- полная поддержка декодировании MPEG-4 ASP
- высококачественный рескейлер
- снятие ограничений для H.264
- соответствует поддержке Video Decode and Presentation API for Unix feature set C

Содержание

- Введение
- Интерфейсы обработки видео
- Аппаратная поддержка
  - Nvidia PureVideo
  - **ATI AVIVO & Unified Video Decoder**
- Декодирование на CUDA
- Выводы

ATI AVIVO

Начиная с видеокарт X-серии (2005 год) поддерживаются:

- автоматическое регулирование цветности, денойзинг, шарпен
- частичное декодирование форматов H.264, VC-1, WMV9, MPEG-2 (кроме декодирования битового потока)
- адаптивный деинтерлейсинг, скейлинг

ATI Unified Video Decoder/ Unified Video Decoder+

В первом поколении, доступном с 2xxx серии (2007 год):

- полная поддержка VC-1, AVC/H.264
- поддержка MPEG-2 в виде шейдеров
- обширный пост-процессинг: denoising, deinterlacing, scaling/resizing, in-loop deblocking

ATI Unified Video Decoder 2.0

Во втором поколении, доступном с 4xxx серии (2008 год):

- полная поддержка декодирования потоков VC-1, AVC/H.264, MPEG-2
- декодирование двойного потока
- Picture-in-picture
- полностью BD-Live совместима

ATI UVD vs. NV PureVideo

- Ситуация такова, что пользователь не проиграет
- Современные графические процессоры управляются с потоком любой сложности
- Загрузка CPU – минимальна
- Качество картинки – на усмотрение пользователя

ATI UVD vs. NV PureVideo

ATI AVIVO

http://www.hardwareheaven.com/reviews.php?reviewid=552

ATI UVD vs. NV PureVideo

NVIDIA PureVideo

Содержание

- Введение
- Интерфейсы обработки видео
- Аппаратная поддержка
- **Декодирование на CUDA**
- Выводы

Библиотека NVCUVID

- Предоставляет возможность управлять процессором декодирования
- Поддерживает MPEG-1, MPEG-2, H.264
- Декодированный кадр сразу сохраняется в CUDA Device Memory
- Позволяет отображать видео (D3D или OGL) или выгружать кадры из видеопамяти

Библиотека NVCUVID

Последовательность действий:

- Разобрать исходное видео (с помощью API)
- Декодировать очередной кадр (с помощью API)
- Произвести изменения кадра (пост-обработка)
- Вывести кадр на экран (с помощью 3D API)

Создание декодера

Пользователь заполняет структуру CUVIDDECODECREATEINFO для вызова cuvidCreateDecoder(), с информацией входного кадра:

- тип кодека
- размеры кадра
- цветовой формат

Также пользователь указывает параметры выходных данных:

- размеры кадра
- цветовой формат
- количество кадров

Создание декодера

```
#include <cuvid/cuviddec.h>
#include <cuvid/nvcuvid.h>
…
Cuvideodecoder                  oDecoder ;
CUVIDDECODECREATEINFO           oVideoDecodeCreateInfo_;
memset(&oVideoDecodeCreateInfo_, 0, sizeof(CUVIDDECODECREATEINFO));
oVideoDecodeCreateInfo_.CodecType= cudaVideoCodec_H264;
oVideoDecodeCreateInfo_.ulWidth  = 1280;
oVideoDecodeCreateInfo_.ulHeight = 720;
oVideoDecodeCreateInfo_.ChromaFormat = cudaVideoChromaFormat_420;
oVideoDecodeCreateInfo_.OutputFormat = cudaVideoSurfaceFormat_NV12;
oVideoDecodeCreateInfo_.DeinterlaceMode = cudaVideoDeinterlaceMode_Adaptive;
…
cuvidCreateDecoder(&oDecoder, &oVideoDecodeCreateInfo_);
```

Декодирование кадра

Для вызова функции декодирования cuvidDecodePicture() пользователь заполняет структуру CUVIDPICPARAMS, содержащую :

- информацию об интерлейсинге кадра
- указатели на данные кадра
- размеры кадра

Декодер записывает кадры в очередь, откуда их можно достать по номеру

Декодирование кадра
Подготовка парсера

```
CUVIDPARSERPARAMS      oVideoParserParameters;
Cuvideoparser          hParser_;
memset(&oVideoParserParameters, 0, sizeof(CUVIDPARSERPARAMS));
oVideoParserParameters.CodecType  = cudaVideoCodec_H264;
oVideoParserParameters.ulMaxDisplayDelay = 1;
oVideoParserParameters.pUserData  = &oParserData_;
oVideoParserParameters.pfnDecodePicture  = HandlePictureDecode;
cuvidCreateVideoParser(&hParser_, &oVideoParserParameters);
```

Декодирование кадра
Вызов декодера

```
HandlePictureDecode(void * pUserData, CUVIDPICPARAMS * pPicParams)
{
    cuvidDecodePicture(oDecoder, pPicParams);

    return true;
}
```

Пост-обработка и отображение

- Подготавливаем kernel
- Отображение в память вызовом cuvidMapVideoFrame()
- Вызов kernela для обработки кадра вызовом cudaPostProcessFrame()
- Отображение при помощи 3D API

Пост-обработка и отображение

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
#include "cudaModuleMgr.h"
CUmoduleManager  * g_pCudaModule;
CUfunction          g_mykernel;
CUStream            g_KernelSID;
g_pCudaModule = new CUmoduleManager("mykernel.ptx", exec_path,
        nKernels, nGlobalMem, nTexRef);
g_pCudaModule->GetCudaFunction("my_main", &g_mykernel);
…
GLuint gl_pbo = createPixelBufferObject(imageParams);
cuGLRegisterBufferObject(gl_pbo);
glutDisplayFunc(display);
…
void display(){
        copyDecodedFrameToTexture();
        render(); }
```

Пост-обработка и отображение

GRAPHICS & MEDIA LAB
VIDEO GROUP

```
void copyDecodedFrameToTexture()
{
    CUdeviceptr    pPBOData  = 0;
    unsigned int   nPBOPitch = 0;
    Cudeviceptr    pDecodedFrame = 0;
    unsigned int   nDecodedPitch = 0;
    CUVIDPARSERDISPINFO  oDisplayInfo = queryImage();
    CUVIDPROCPARAMS oVideoProcessingParameters = fillParameters();
    cuvidMapVideoFrame(oDecoder, oDisplayInfo.picture_index,
          &pDecodedFrame , &nDecodedPitch , &oVideoProcessingParameters);
    cuGLMapBufferObject( &pPBOData, &nPBOPitch,   gl_pbo);

...
```

Пост-обработка и отображение

```
    ...
    cudaPostProcessFrame(&pDecodedFrame, nDecodedPitch, &pPBOData,
            nPBOPitch, g_pCudaModule->getModule(), g_mykernel, g_KernelSID);
    SetKernelParameters(g_mykernel);
    cuLaunchGridAsync( g_mykernel, grid.x, grid.y, g_KernelSID);
    cuGLUnmapBufferObject(gl_pbo);
    cuvidUnmapVideoFrame(oDecoder_, &pDecodedFrame);
}
void render(){
        glBindBufferARB (GL_PIXEL_UNPACK_BUFFER_ARB, gl_pbo);
        glBindTexture   (GL_TEXTURE_2D, t_tex);
        glTexSubImage2D (… , 0);
        DrawImage();
}
```

NVCUVID
Резюме

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Гибкая система пост-обработки
- Все производится внутри видеопамяти
- Кросс-платформенность

Содержание

- Введение
- Интерфейсы обработки видео
- Аппаратная поддержка
- Декодирование на CUDA
- **Выводы**

Выводы

- Каковы бы ни были условия входящего видеопотока – производители GPU будут готовы
- Готовые декодеры удовлетворяют большинству требований пользователя, для определенных нужд – можно запрограммировать свой

Список литературы

1. http://www.freedesktop.org/wiki/Software/vaapi VA API
2. http://msdn.microsoft.com/en-us/library/ms798379.aspx DirectX Video Acceleration
3. http://msdn.microsoft.com/en-us/library/aa965263(VS.85).aspx DirectX Video Acceleration 2.0
4. http://forum.doom9.org/ Doom9's Forum
5. http://cbaoth.dk/~cbaoth/nvcuvid.pdf CUDA Video Decoder API
6. http://nvworld.ru/utilities/dxvacheck/ DXVA Checker
7. http://www.hardwareheaven.com/reviews.php?reviewid=552&pageid=14 UVD vs PureVideo

Лаборатория компьютерной графики и мультимедиа

GRAPHICS & MEDIA LAB
VIDEO GROUP

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