О ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ ВИДЕОАППАРАТУРЫ ДЛЯ РАБОТЫ С МУЛЬТИМЕДИА ДАННЫМИ, КАК С АБСТРАКТНЫМ ПОТОКОМ

Бобрышев Д.Н., группа ЭлС-03
Руководитель проф. каф. ЭТ Чичикало Н.И.

Аннотация. В статье рассмотрены варианты связи видеоаппаратуры с компьютером.

Актуальность Захват видео кадра с видео потока — одна из проблем, необходимых для анализа поставленной задачи Актуальность темы в том что можно выделять отдельный кадр с видео потока информации. И полученный кадр сохранять в виде картинки на компьютер. В качестве источника видео потока не обязательно иметь видео файл, можно воспользоваться любой цифровой техникой: начиная с Цифровой камеры, цифрового фотоаппарата, Веб-камеры, мобильного телефона со встроенной или съемной камерой, и видеокамеры. Необходимо лишь подключить устройство к вычислительной машине и запустить программу.

Анализ существующих измерительных систем. На сегодняшней день распространены цифровые видео камеры или фото камеры. Но различных программ мало. Известны программы для записи и для воспроизведения видео потока. Есть также программы для преобразования форматов видеопотоков. Но довольно мало программ по захвату кадра.

DirectShow — это архитектура для воспроизведения, перехвата и обработки потоков мультимедиа. С помощью этого API можно:

– проигрывать мультимедийные файлы различного формата, такие как MPEG (Motion Picture Experts Group), AVI (Audio-Video Interleaved), MP3 (MPEG Audio Layer-3), DVD и конечно WAV;

– перехватывать видео-поток с различного рода TV-карт, видеокамер и т.п.;

– создавать нестандартные обработчики мультимедиа-потоков и свои собственные форматы файлов (что, впрочем, вряд ли понадобится простым смертным);

– обращаться непосредственно к видео и аудио потокам, чтобы выводить их на Surface DirectDraw (что для нас как раз интересно).

Звучит заманчиво. Это может понадобиться, например, если представить себе, что Вы запрограммировали трехмерный мир, с анимированными спрайтами, трехмерными объектами и т.п. И в отсвете пользователь нового шедевра, видит воспроизводимый видео клип. DirectDraw — DirectShow интегрирован с DirectX так, что использует DirectDraw и DirectSound для вывода изображения и звука, и, при наличии аппаратного ускорения, автоматически им воспользуется.

Для работы с DirectShow необходимо:

– иметь некоторое представление о технологии COM (Component Object Model) — хотя быть знатоком этой технологии вовсе не обязательно — просто достаточно знать, что для получения COM-интерфейса нужно вызвать QueryInterface;

– скачать заголовочные файлы DirectShow API, переведенные на Delphi в рамках проекта JEDI — (http://www.delphi-jedi.org/DelphiGraphics/DirectX/DX5Media.zip) и либо поместить их в каталог Delphi\Lib либо добавить путь к каталогу, в котором они находятся в установках Delphi Library Path. DirectShow скорее всего уже установлен на компьютере — он входит в стандартную поставку Windows 9x, Windows NT 4 (Service Pack 3 и выше), Windows 2000 (если Вы программируете для UNIX или DOS — то это известно)

Основы DirectShow

Цель: поиск путей захвата изображения с видео потока при помощи компьютера и языка программирования Delphi 7.0.

Изложение основного материала

В концепции DirectShow мультимедийные данные — это поток, который проходит через несколько обрабатывающих блоков. Блоки, обрабатывающие поток данных, передают данные по цепочке друг другу, таким образом можно представить себе несколько "устройств", каждое из которых выполняет какую-то обработку данных и передает их соседнему "устройству". Эти "устройства" или "блоки обработки" данных называют фильтрами. Цепочка, по которой передаются данные, содержит несколько фильтров, связанных определенным образом.

В DirectShow имеются готовые фильтры, из которых, словно из детских кубиков, программист может выстроить ту или иную цепочку обработку данных, кроме того, конечно, можно создать свои, нестандартные фильтры.

Для создания такой "цепочки обработки" (которая, кстати, официально называются Filter Graph — "граф фильтров" или, в несколько вольном переводе — "схема соединения фильтров"), так вот для создания схемы соединения фильтров, предназначен самый базовый и лежащий в основе всех основ компонент DirectShow, под названием Filter Graph Manager — Менеджер Графа Фильтров.

Например, программа, показывающая видео из AVI-файла может построить граф фильтров (рис. 1.): пакованные видео данные фильтру AVI Decompressor, который их распаковывает и передает фильтру Default DirectSound Device, выводящему звук. AVI Decompressor передает распакованные данные фильтру Video Renderer, который выводит кадры видео на экран.

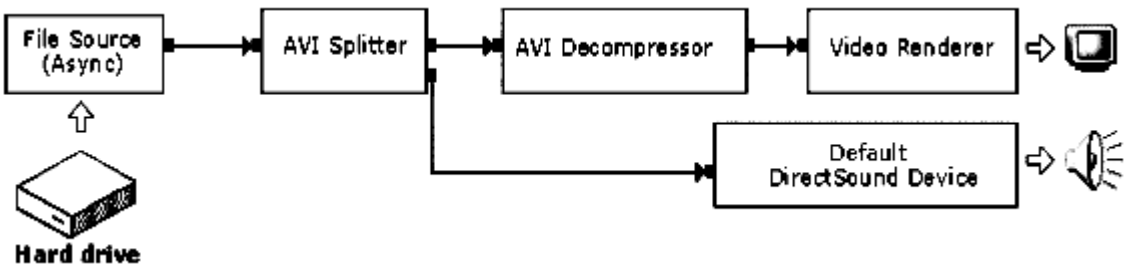

Рисунок 1 — Граф фильтров

Фильтры делятся на три типа:

Фильтры-источники (Source filters) — эти фильтры просто получают данные из какого-то источника, с диска (как фильтр File Source (Async) на рисунке), с CD или DVD дисковода или с TV- карты или карты, к которой подключена цифровая видеокамера.

Фильтры-преобразователи (Transform filters) — эти фильтры как видно из названия преобразуют поток данных, проходящий через них каким-либо образом, например — разделяет поток данных на кадры, производят декомпрессию и т.п. На нашем рисунке к таким фильтрам относятся AVI Splitter и AVI Decompressor. Фильтры вывода (Renderer filters) — фильтры, которые получают полностью обработанные данные и выводят их на монитор, звуковую карту, пишут на диск или выводят на еще какое-нибудь устройство. Из фильтров-кубиков можно выстраивать граф. Делается это с помощью интерфейса IGraphBuilder. Создать объект типа IGraphBuilder можно так:

```
CoCreateInstance (CLSID_FilterGraph, nil, CLSCTX_INPROC_SERVER, IID_IGraphBuilder, MyGraphBuilder);
```

Здесь переменная MyGraphBuilder имеет тип IGraphBuilder; идентификатор класса CLSID_FilterGraph и IID_IGraphBuilder объявлены в файле DShow.pas, поэтому их необходимо добавить; uses DShow.pas

Итак, интерфейс IGraphBuilder получен. Можно построить граф фильтров, такой, какой нам нужно. Впрочем, IGraphBuilder достаточно интеллектуален, он может сам, автоматически, построить граф, в зависимости от того какие файлы мы собираемся воспроизводить. IGraphBuilder имеет метод RenderFile, который получает имя файла в качестве параметра и, в зависимости от типа файла (которое определяется по расширению и по специальным сигнатурам в файле), сканирует реестр, в поисках необходимой для построения графа информации, а

также создает необходимые фильтры и строит граф, предназначенный для воспроизведения файлов этого типа (WAV, AVI, MP3, MPG и т.д.). После построения графа DirectShow готов к воспроизведению. Для управления потоком данных через граф обработки предназначен интерфейс IMediaControl — он имеет методы Run, Pause и Stop (названия говорят сами за себя)

Примеры конкретной реализации. Рассмотрим пример:

```
uses
... DShow, ActiveX,ComObj;
var
MyGraphBuilder : IGraphBuilder;
MyMediaControl : IMediaControl;
begin
CoInitialize(nil);
{получаем интерфейс IGraphBuilder}
CoCreateInstance(CLSID_FilterGraph,nil,CLSCTX_INPROC_SERVER,IID_IGraphBuilder,
MyGraphBuilder);
{вызываем RenderFile — граф фильтров строится автоматически}
MyGraphBuilder.RenderFile(cool.avi',nil);
{получаем интерфейс ImediaControl}
MyGraphBuilder.QueryInterface(IID_IMediaControl,MyMediaControl);
{Примечание — MyMediaControl — переменная типа IMediaControl}
{проигрываем видео}
MyMediaControl.Run;
{ждем пока пользователь не нажмет ОК (видео воспроизводится в отдельном (thread) потоке)}
ShowMessage('Нажмите ОК');
CoUninitialize;
end;
```

Если не скопировать эту часть кода в Delphi и не запустить его то avi-файл проигрывается в отдельном окошке, которое не принадлежит нашему приложению. Для управления окошком, в котором воспроизводится видео предназначен специальный интерфейс IVideoWindow. Получить этот интерфейс можно из экземпляра IGraphBuilder, вызвав QueryInterface и передав в качестве идентификатора интерфейса константу IID_IvideoWindow.

Интерфейс IVideoWindow содержит методы для управления заголовком, стилем, местоположением и размерами окошка в котором проигрывается видео. Поставим задачу переделать наш пример так, чтобы видео выводилось не в отдельном окошке, а, на компоненте TPanel, расположенном в нашей форме. Добавим на форму компонент TPanel, пусть он называется Panel1.

```
uses
... DShow, ActiveX,ComObj;
procedure TForm1.Button1Click(Sender: TObject);
var
MyGraphBuilder : IGraphBuilder;
MyMediaControl : IMediaControl;
VideoWindow : IVideoWindow;
begin
CoInitialize(nil);
{получаем интерфейс IGraphBuilder}
CoCreateInstance(CLSID_FilterGraph,nil,CLSCTX_INPROC_SERVER,IID_IGraphBuilder,
MyGraphBuilder);
{вызываем RenderFile — граф фильтров строится автоматически}
MyGraphBuilder.RenderFile('C:\Program
Files\Borland\Delphi5\Demos\Coolstuf\cool.avi',nil);
{получаем интерфейс ImediaControl}
MyGraphBuilder.QueryInterface(IID_IMediaControl,MyMediaControl);
{Примечание — MyMediaControl — переменная типа IMediaControl}
{получаем интерфейс IVideoWindow}
MyGraphBuilder.QueryInterface(IID_IVideoWindow,VideoWindow);
{Примечание — VideoWindow — переменная типа IVideoWindow}
{располагаем окошко с видео на панель}
VideoWindow.Set_Owner(Self.Panel1.Handle);
VideoWindow.Set_WindowStyle(WS_CHILD OR WS_CLIPSIBLINGS);
VideoWindow.SetWindowPosition(0,0,Panel1.ClientRect.Right,Panel1.ClientRect.Bottom);
{проигрываем видео}
MyMediaControl.Run;
ShowMessage('Нажмите OK');
CoUninitialize;
end;
```

Очевидно, что это просто

DirectShow и DirectX

Для того, чтобы использовать DirectShow совместно с DirectX нужно разобраться с понятием Multimedia Streaming (потоки мультимедиа).

Multimedia Streaming — это архитектура, используемая в DirectShow для облегчения жизни программиста. Эта архитектура позволяет работать с мультимедиа данными, как с абстрактным потоком, не вдаваясь в подробности форматов хранения мультимедиа-файлов или специфику устройств-источников мультимедиа. Используя эту архитектуру, программист концентрируется не на расшифровке и преобразовании данных, а на управлении потоком данных, кадрами видео или аудио семплами.

На рис. 2 изображена иерархия объектов Multimedia Streaming

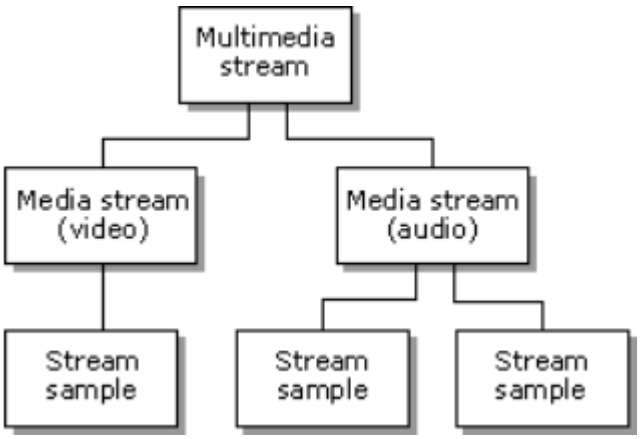

Рисунок 2 — Иерархия объектов Multimedia Streaming

На вершине иерархии находится базовый объект Multimedia Stream, который является контейнером для объектов Media Stream. Объект Multimedia Stream может содержать один или несколько объектов Media Stream. В то время как каждый объект типа Media Stream предназначен для работы с данными какого-то одного типа (видео, аудио и т.п.) — Multimedia Stream — просто содержит методы для обращения к содержащимся в нем объектам Media Stream и не зависит от типа данных.

Объект типа Stream Sample, доступ к которому можно получить из Media Stream — позволяет управлять непосредственно элементами мультимедийного потока (для видео каждый Sample — это кадр видеоизображения, для аудио он может содержать несколько семплов звука).

Далее создадим необходимые объекты, чтобы вывести видео на Surface DirectX. Для этого нам надо обращаться к кадрам видео изображения (т.е. к объекту типа Stream Sample). Цепочка объектов, которую предстоит создать выглядит так:

```
IAMMultiMediaStream
  + IDirectDrawMediaStream
      + IDirectDrawStreamSample
```

Сравним это с нашим рисунком. Как видим, на вершине находится объект типа MultiMediaStream, который будет содержать MediaStream конкретного, нужного нам типа (IDirectDrawMediaStream). С помощью него мы получим доступ к конкретным видео кадрам через интерфейс IDirectDrawStreamSample.

Итак, сейчас мы создадим объект типа IAMMultiMediaStream. Этот интерфейс унаследован от IMultimediaStream и содержит, кроме прочего, функцию OpenFile, которая автоматически строит граф фильтров для воспроизведения файла.

```
CoCreateInstance(CLSID_AMMultiMediaStream, nil, CLSCTX_INPROC_SERVER, IID_IAMMultiMediaStream, AMStream);
```

Здесь переменная AMStream имеет тип IAMMultiMediaStream. Мы создали контейнер для мультимедийных потоков. Сверяемся с рисунком — мы на верхнем уровне иерархии. У нас есть объект типа IMultimediaStream — теперь в этот контейнер нужно проинициализировать и добавить один или несколько мультимедиа потоков, нужного нам типа. Сначала инициализация:

```
AMStream.Initialize(STREAMTYPE_READ, AMMSF_NOGRAPHTHREAD, nil);
```

При инициализации указываем, что будут создаваться мультимедиа потоки для чтения, передав значение STREAMTYPE_READ (другие варианты STREAMTYPE_WRITE, STREAMTYPE_TRANSFORM).

Создадим теперь мультимедиа потоки для видео и звука:

```
AMStream.AddMediaStream(DDraw, MSPID_PrimaryVideo, 0, NewMediaStremVideo);
AMStream.AddMediaStream(nil, MSPID_PrimaryAudio, AMMSF_ADDDEFAULTRENDERER, NewMediaStremAudio);
```

Вызываем метод OpenFile — файл загружается, и автоматически строится граф фильтров:

```
AMStream.OpenFile('cool.avi', 0);
```

Осталось направить видео поток мультимедиа поток на Surface. Вот процедура, которая делает это:

```
procedure TForm1.RenderStreamToSurface(Surface : IDirectDrawSurface; MMStream : IMultiMediaStream);
var
PrimaryVidStream : IMediaStream;
DDStream : IDirectDrawMediaStream;
Sample : IDirectDrawStreamSample;
RECT : TRect;
ddsd : TDDSURFACEDESC;
Z : DWORD;
begin
MMStream.GetMediaStream(MSPID_PrimaryVideo, PrimaryVidStream);
PrimaryVidStream.QueryInterface(IID_IDirectDrawMediaStream, DDStream);
ddsd.dwSize := sizeof(ddsd);
DDStream.GetFormat(ddsd, Palitra, ddsd, Z);
rect.top:=(480–ddsd.dwHeight)div 2; rect.left:=(640–ddsd.dwWidth) div 2;
rect.bottom := rect.top+ddsd.dwHeight;
rect.right := rect.left+ddsd.dwWidth;
DDStream.CreateSample(Surface, Rect, 0, Sample);
MMStream.SetState(STREAMSTATE_RUN);
end;
```

В этой процедуре сначала получаем интерфейс типа IDirectDrawMediaStream (соответствующий второму уровню иерархии на нашем рисунке), потом из него получаем объект типа IDirectDrawStreamSample (переменная Sample), который соответствует третьему уровню иерархии нашего рисунка. Теперь остается вызывать в цикле (в демо-программе это делается по таймеру — здесь для простоты опускаем):

```
hr:=Sample.Update(0 , 0, nil, 0);
if hr = $40003 {MS_S_ENDOFSTREAM} then
    MMStream.Seek(0);
```

Метод IDirectDrawStreamSample.Update выводит очередной кадр на Surface. При достижении конца потока он вернет ошибку с кодом $40003 (MS_S_ENDOFSTREAM), я в этом случае просто перематываю поток к началу, методом Seek.

Вывод:

1. Имея видео файл, можно снимать кадры и хранить в базах данных, или же, имея устройства для восприятия видео потока, можно напрямую захватывать часть видео потока как картинки. Недостатки: имея кадры, мы имеем два вида информации:

– сам человек или объект виден

– в поле зрения виден задний фон.

2. При работе с изображением, например, в медицинских или измерительных целях, отличить цвет основной цели и фона нужно в ручную. Программа будет принимать всю картинку за единое целое.

3. Исключение будет лишь при работе с микроскопом, где можно выставить границы, в которых будет лежать наш химический состав и в этом случае можно определить компоненты состава.

4. Такой метод будет использовать созданные базы данных.