

# ОСОБЕННОСТИ ПРИМЕНЕНИЯ МЕТОДОВ DirectX ДЛЯ ИСПЫТАНИЯ СТАДИЙ СЖАТИЯ СИГНАЛА ЦИФРОВОГО ТЕЛЕВИДЕНИЯ

Поташников А.М., Дьячкова Л.И.

В данной работе рассмотрены особенности использования средств пакета разработки DirectX для создания специализированного программного обеспечения. Были разработаны методы реализации программных приложений с отображением результатов исследований на экране компьютера в реальном режиме времени. Данные методы могут быть применены при испытаниях процессов сжатия телевизионных видеопоследовательностей.

При разработке новых методов сжатия видеопоследовательностей в современных средствах цифрового телевизионного вещания появляется возможность исследования процессов кодирования и декодирования с помощью компьютерной техники.

Использование программного обеспечения для моделирования стадий сжатия цифрового видеосигнала позволяет удобно расположить элементы управления и отображения. Таким образом, появляется возможность, используя минимальное количество действий, моделировать и отобразить требуемую информацию.

Для цифрового телевидения отображение результатов исследования на экране монитора позволяет наглядно оценить результаты воздействия эффектов сжатия видеопоследовательностей с потерей информации. Таким образом, использование компьютерной графики дает возможность оценить результаты работы кодеров и декодеров в реальном режиме времени не только объективно, но и субъективно.

Для возможности работы с видеопоследовательностями требуется решить задачу отображения видеоинформации в реальном режиме времени. Поскольку обработка видеопотока требует больших затрат ресурсов процессора, отображение видеоинформации должно требовать минимальных затрат ресурсов процессора.

Стандартные средства разработки Windows предоставляют возможность отображения с применением WinAPI (application programming interfaces – интерфейсы программирования приложений) для операционных систем семейств Windows и Windows NT корпорации «Майкрософт». В связи с тем, что интерфейсы WinAPI являются универсальными, они удобны в использовании, однако обладают низкой производительностью. Это делает их непригодными для разработки приложений, направленных на работу с отображением видеоинформации.

Для быстрого отображения графики в основном используются два крупных SDK (software developer's kit – комплекты средств разработки) DirectX и OpenGL. OpenGL – это графическая библиотека, а DirectX – мультимедийная [6]. Из этого следует, что OpenGL предназначена

только для обработки и вывода графики на экран монитора, тогда как DirectX предназначена также для работы с видео и аудио устройствами. Поэтому для решения задач отображения видеoinформации наиболее удобным решением является использование средств разработки DirectX.

Поскольку на разных компьютерах используется разное программное и аппаратное обеспечение, то существует возможность, что программа-стенд, работающая на одном компьютере, может не работать на другом. Среди этих проблем наиболее важными являются [5]:

1. Несовместимость с операционной системой (DirectX работает только под операционной системой Windows).
2. Неправильное разрешение экрана (Элементы изображения могут не отобразиться на экране с разрешением меньшим, чем предполагается разработчиком).
3. Неправильная глубина цвета (Может привести к искажению цветов и размера или программа может не работать).

Следовательно, при создании стенда требуется либо оговаривать системные требования, либо использовать универсальный код, что может привести к замедлению работы программы.

При использовании средств DirectX существует возможность создавать приложения в оконном и полноэкранном режимах. Достоинствами оконного режима является возможность быстрого переключения между приложениями. Наличие окон позволяет легко создать стенд, в котором на экране в удобных областях можно отобразить ту или иную информацию. Однако применение оконного режима не дает возможности эксклюзивного режима, то есть режима, в котором все ресурсы компьютера используются одним приложением, что замедляет работу стенда. Кроме того, для измерения телевидения ТВЧ (высокой четкости) требуется использование большого размера экрана, что возможно только в полноэкранном режиме. В полноэкранном режиме можно вручную задать установки монитора (глубину цвета и разрешение)

Скорость работы DirectDraw достигается тем, что этот модуль дает возможность прямого доступа к видеопамяти [2], то DirectDraw предоставляет адрес памяти, в котором находится первый пиксель изображения (поверхности). Адреса остальных пикселей находятся расчетом отступа адреса от начального с учетом глубины цвета. За счет этого рисование или считывание с поверхности состоит из количества действий согласно формуле (1)

$$N = 2 \cdot h \cdot w \quad (1)$$

где  $N$  – количество действий,

$h$  и  $w$  – высота и ширина изображения.

С помощью прямого доступа к памяти возможно быстро осуществить чтение информации с плоскости, произвести над ней требуемые действия и отобразить результирующее

изображение. При этом в большинстве случаев основные ресурсы процессора тратятся на исследования или преобразование, а не на чтение и отображение. Таким образом, появляется возможность обрабатывать большое количество изображений за малый период времени, что особенно важно при испытаниях, связанных с цифровым телевидением.

В случае, если не требуется производить изменений на изображении, что часто требуется при отображении оригинала видеопоследовательностей, DirectDraw предоставляет набор функций для масштабирования и отображения целого изображения или его части [5].

Поскольку в большинстве случаев работа с видеопоследовательностями и их запись производятся на поверхности, которые не отображаются, а записываются в системную память компьютера, данные функции являются особенно полезными при последующем отображении поверхностей на экране монитора. Масштабирование изображений может применяться для отображения эскизов (уменьшенных копий) видеопоследовательностей, а также для увеличения кадров или их частей. Возможная скорость отображения кадров на современном компьютере (при отсутствии дополнительных действий) указана в таблице 1.

Таблица 1. Количество кадров в секунду, которое можно отобразить, используя DirectDraw

№ эксперимента	1	2	3	4	5	6	7	8	9	Средн.
BltFast	5851	9106	8126	7839	9426	10841	10076	10887	10912	8446
BltFast(часть изображения)	7151	9377	9395	10401	7892	10153	7151	9698	6501	8635
Blt	5947	5709	6347	5955	5141	3621	5669	4614	6092	5455

При проведении эксперимента использовались кадры формата cif (352x288). BltFast и Blt – функции DirectDraw для отображении одной поверхности на другой. Blt позволяет осуществлять масштабирование поверхностей и отображение их или их частей на другие поверхности. BltFast (Быстрое Blt) не обладает функцией масштабирования, что позволяет выполнять действия в 1.5 раз быстрее [2].

Несмотря на возможность быстрого создание (рисования) изображений на плоскостях, операции Blt и BltFast выполняются в несколько раз быстрее, поэтому при прорисовке одного изображения (статического) удобно оставить плоскость в памяти компьютера, и выводить ее при каждой перерисовке экрана. Данный метод позволяет сильно ускорить быстроедействие программы, однако при использовании большого количества таких плоскостей программе может не хватить оперативной памяти. При разработке стендов исследований для цифрового телевидения это может стать серьезной проблемой. Для хранения одного кадра формата ВЧ требуется более 8МБ. В связи с тем, что при обработке видеoinформации требуется большое количество памяти как для записи самих кадров видеопоследовательностей, так и для хранения служебной информации, дополнительное использование памяти может привести к использованию жесткого диска для временного хранения, что существенно замедляет работу.

Возможность «просматривать» кадр видеопоследовательности пиксель за пикселем за счет смещения адреса позволяет параллельным сканированием сравнивать два кадра (исходное и измененное изображение), измеряя значения цветов. На основе результатов можно построить разностное изображение или проверить степень искажение, например, по среднеквадратичному отклонению MSE (Mean Square Error)[4].

$$MSE = \frac{1}{h \cdot w} \sum_{n=0}^{h \cdot w} (Value_2 - Value_1)^2 \quad (2)$$

где  $h$  и  $w$  – высота и ширина кадров,

$Value_1$  – Значение цвета пикселя исходного кадра,

$Value_2$  – Значение цвета пикселя измененного кадра.

Используя эти технологии, был разработан стенд для исследования влияния спектральных преобразований, степени влияния их гармоник, а также шага квантования на видеопоследовательность. Часть этого стенда изображена на рисунке 1. На рисунке верхний ряд – это эскизы стадий прямого и обратного целочисленных преобразований. В нижнем ряду изображены увеличенные изображения исходного и преобразованного кадров. Из кадра убраны каждая вторая гармоника преобразования в вертикальном и горизонтальном направлении.

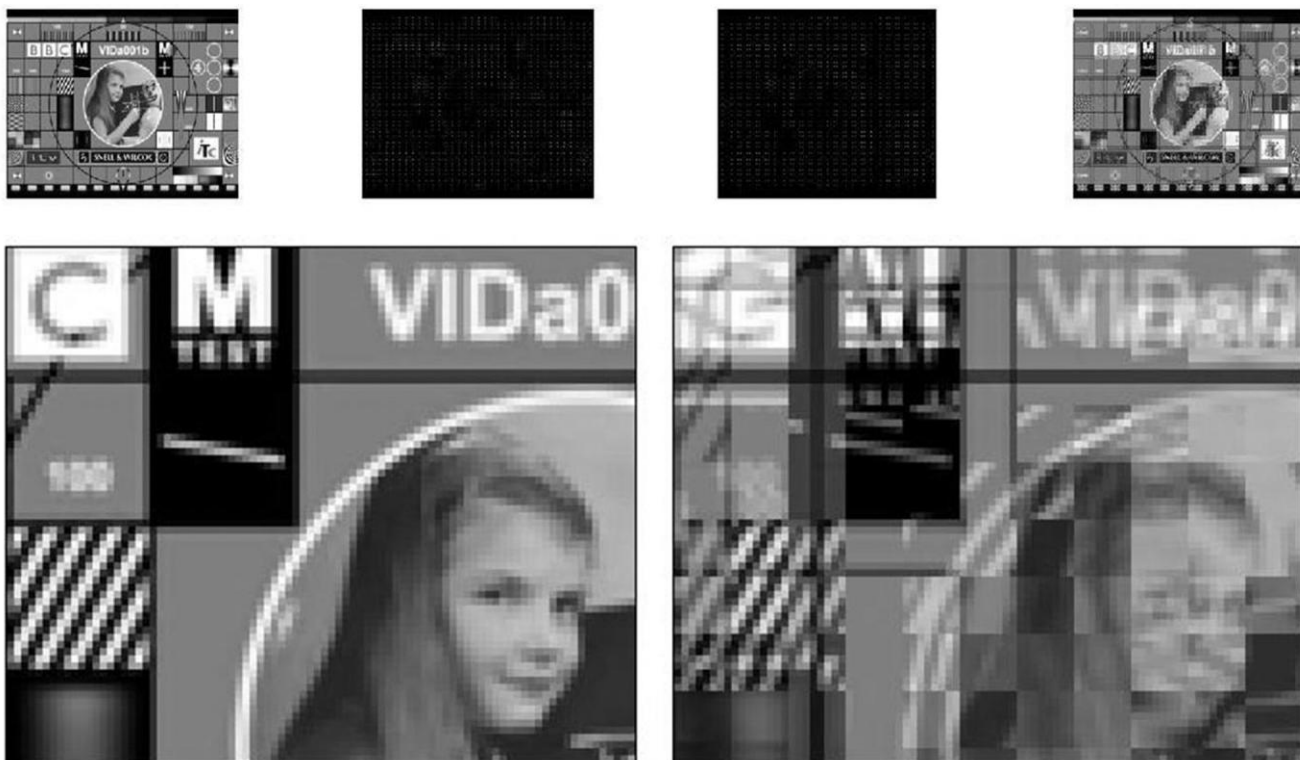


Рисунок 1. Часть телевизионного измерительного стенда

При выделении блока для преобразования определяются адреса в памяти, соответствующие точкам начала каждой строки в блоке инкрементом адреса на область памяти, занимаемой одной строкой. После этого находятся значения остальных отсчетов за счет инкремента каждого адреса на 32бита (глубину цвета). После этого полученный массив

умножается на матрицу преобразования для получения коэффициентов соответствующего преобразования [1].

Для целочисленного преобразования 4x4 применяется следующая матрица:

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}; \quad (3)$$

Дискретное косинусное и дискретное синусное преобразования выполняются для размера 4x4, 8x8, 16x16, 32x32, поэтому возможны только с изображениями, в которых количество строк и элементов в строке кратны тридцати двум. Поэтому стенд работает с последовательностями формата cif (352x288). В отличие от целочисленного преобразования, коэффициенты матриц косинусного и Фурье преобразований рассчитываются, что обеспечивает возможность использовать один код для преобразований любого размера, являющегося степенью двух.

Синусное преобразование не используется в кодерах и декодерах цифрового телевидения, поэтому не описывается в настоящей статье. Косинусное преобразование рассчитывается по формуле [3]:

$$\begin{aligned} M(j) &= 1; 0 \leq j < x; \\ M(i \cdot x + j) &= \sqrt{2} \cos\left(\frac{\pi i}{x} \cdot (j + 0.5)\right), 0 < i < x, 0 \leq j < x, \end{aligned} \quad (4)$$

где  $i$  и  $j$  – положение значения в матрице,

$x$  – линейный размер матрицы,

$M$  – матрица преобразования.

Преобразование Фурье требует использования двух матриц, поскольку реальная и мнимая части рассчитываются отдельно[3]:

$$\begin{aligned} 0 \leq i < x; 0 \leq j < x; \\ arg &= -2 \frac{\pi i \cdot j}{x}; \\ \Re M(i \cdot x + j) &= \cos(arg); \\ \Im M(i \cdot x + j) &= \sin(arg), \end{aligned} \quad (5)$$

где  $i$  и  $j$  – положение значения в матрице ( $i$  и  $j$ ),

$x$  – линейный размер матрицы,

$RM$  и  $IM$  – матрицы реальных и мнимых значений преобразования.

В большинстве случаев при программировании на C++ не используют многомерные массивы. Если необходимо использовать двумерные массивы, то значения записывают в одномерный массив строка за строкой. Поэтому в формулах, приведенных выше  $M(i \cdot x + j)$  означает  $M(i, j)$ . При обработке изображений такая запись удобна вследствие того, что таким же образом построена запись значений цветов пикселей поверхности в C++.

Для наглядного отображения преобразованного изображения отсчеты отображаются на плоскости, на которой величины показаны яркостью соответствующих точек. На рисунке это второе и третье изображение в верхнем ряду.

DirectDraw используется для хранения плоскостей с изображениями, однако промежуточные результаты предпочтительно хранить в системной памяти, не используя средства DirectDraw. Это дает больше возможностей для работы, поскольку дает возможность использовать методы, не связанные с DirectDraw. Кроме того, в этом случае действия выполняются быстрее, поскольку видеопамять работает быстро при чтении информации, но медленно при записи.

Создание восстановленного кадра производится таким же образом, как и выделение блоков. В памяти, выделенной на плоскость, находятся требуемые адреса и отсчеты заполняются значениями из массива значений отсчетов блока, найденного обратным спектральным преобразованием.

Для удобства просмотра мелких деталей изображения на стенде применена возможность масштабирования изображения (на рисунке изображения в нижнем ряду увеличены в четыре раза). Для целей исследования масштабирование не должно искажать или сглаживать изображение. Для этой цели используется функция Vlt. Поскольку она осуществляет простое масштабирование за счет копирования строк, то при кратном масштабировании структура изображения не искажается.

Поскольку функция Vlt работает медленнее, чем VltFast, при создании эскизов кадров, используется сначала копирование с масштабированием на промежуточную плоскость, которая впоследствии без масштабирования выводится на экран.

Поскольку спектральные преобразования ресурсоемки, то при исследовании отдельных кадров, обновление изображений применяется не при каждом обновлении экрана монитора, а при изменениях условий исследования.

Для убирания мельканий от последовательного отображения элементов на экране применяется двойная буферизация, то есть все отображение для вывода на экран создается на промежуточной плоскости, впоследствии выводятся на экран.

К недостаткам DirectDraw можно отнести отсутствие библиотеки элементов управления (кнопок, панелей или текстовых полей). Это приводит к необходимости программировать собственные элементы управления. С другой стороны специализированные элементы управления работают быстрее универсальных и могут быть удобны для конкретных задач. Кроме того, для испытательных стендов интерфейс должен в первую очередь быть удобным, и не требует больших затрат на художественное оформление.

Таким образом, данная технология может быть применена при исследовании различных

эффектов, проявляющихся при сжатии видеопоследовательностей. По сравнению с аппаратными стендами, данная технология имеет преимущество в ее мобильности, легкости перестройки и возможности создания удобного интерфейса. К ее недостатком можно отнести использование достаточно универсальных технологий, что приводит к уменьшению скорости работы измерительного стенда.

### **Список литературы**

1. Н. Кехтарнаваз, Н. Ким. Цифровая обработка сигналов на системном уровне с использованием LabView – Издательский дом «Додэка-XXI», 2007 – 304 с.
2. Фленов М.Е. DirectX и C++. Искусство программирования. – СПб.: БХВ-Петербург, 2006. – 384 с.: ил.
3. Britanak V., Yip P.C., Rao K.P. Discrete cosine and sine transforms – Academic Press, 2006 – 368 pages.
4. Mood, A.; Graybill, F.; Boes, D. (1974). Introduction to the Theory of Statistics (3rd ed.). McGraw-Hill. p. 229.
5. Wendy Jones. Beginning DirectX 9 – Course Technology PTR, 2004. – 352 pages.
6. R.S. Wright, Jr., B. Lipchak, N. Haemel.. OpenGL(R) SuperBible: Comprehensive Tutorial and Reference (4th Edition) – Addison-Wesley, 2007.