

Оптимизация ДКП-3D видеокодека с использованием технологии OpenMAX DL на примере процессора ARM Cortex-A8

В статье рассматриваются вопросы оптимизации программного кода на основе ДКП-3D для широкого спектра устройств – от SIMD-расширений и DSP-процессоров до специализированных аппаратных ускорителей. Это достигается при использовании программного интерфейса OpenMAX DL с профилировкой приложения на примере процессора ARM Cortex-A8.

Видеокодек, ДКП-3D, оптимизация, производительность, профилирование кода, OpenMAX, ARM, Cortex-A8, NEON, OMAP3530

1. Введение

Общая проблематика данной статьи обусловлена исследованием разработчика фирмы «On2 Technologies» Tero Rintaluoma «Optimizing H.264 decoder for Cortex-A8 with ARM NEON Openmax DL implementation»[6], в котором была проведена оптимизация декодера H.264 видео путём использования векторных инструкций NEON через низкоуровневый интерфейс OpenMAX часть DL. В связи с тем, что в документации ARM присутствует запись о том, что среди модулей OpenMAX DL функции кодирования видео не поддерживаются и это не планируется в будущем[4], в [6] это не позволило провести ту же работу относительно кодера H.264. В данной работе было решено провести подобную оптимизацию для DCT-3D видеокодека, принимая во внимание тот факт, что этот кодек, в отличие от H.264, можно перенести на векторные вычисления практически полностью. В качестве макета в статье используется платформа OMAP3530[3], работающая на процессоре общего назначения ARM Cortex-A8 с тактовой частотой 720 МГц и ЦОС-процессоре TMS320C64x+ с частотой 520 МГц. На целевой системе развёрнута система Linux GeeXboX.

2.1. Структура 3D-DCT кодека, его особенности. Известно, что современные видеокодеки, реализуют оптимальное согласование источников видеоинформации с каналами связи путем компрессии/декомпрессии видеоинформации в соответствии с принципом устранения статистической избыточности (декорреляции). Большая часть кодеков для этого используют трудно формализуемые алгоритмы вычисления межкадровой разности и компенсации движения, которые являются достаточно громоздкими и плохо поддаются оптимизации. В контексте необходимости декорреляции, видеосигнал как многомерная функция имеет оптимальное разложение по всем своим аргументам [1]. Другими словами, для устранения временной корреляции нет существенных формальных оснований использовать средства, отличающиеся от применяемых для пространственной декорреляции. С точки зрения процедуры оптимизации мы получаем унифицированное основное преобразование, отлично оптимизированное и подходящее для аппаратной реализации, одновременно избавляясь от сложных алгоритмов компенсации и поиска векторов движения. Общая структура ДКП-3D кодека представлена на рисунке 1.

2.2. Почему OpenMAX? Изначально стоял выбор между традиционной «ручной» оптимизацией, используя SIMD-инструкции NEON, по аналогии с реализацией векторных вычислений на архитектуре x86 через MMX/SSE/.../SSE4/AVX. При переводе на SIMD-инструкции сложных алгоритмов этот путь является более простым, так как в этом случае

предоставляется большая гибкость в разработке, и нет необходимости перестраивать структуру кодека.

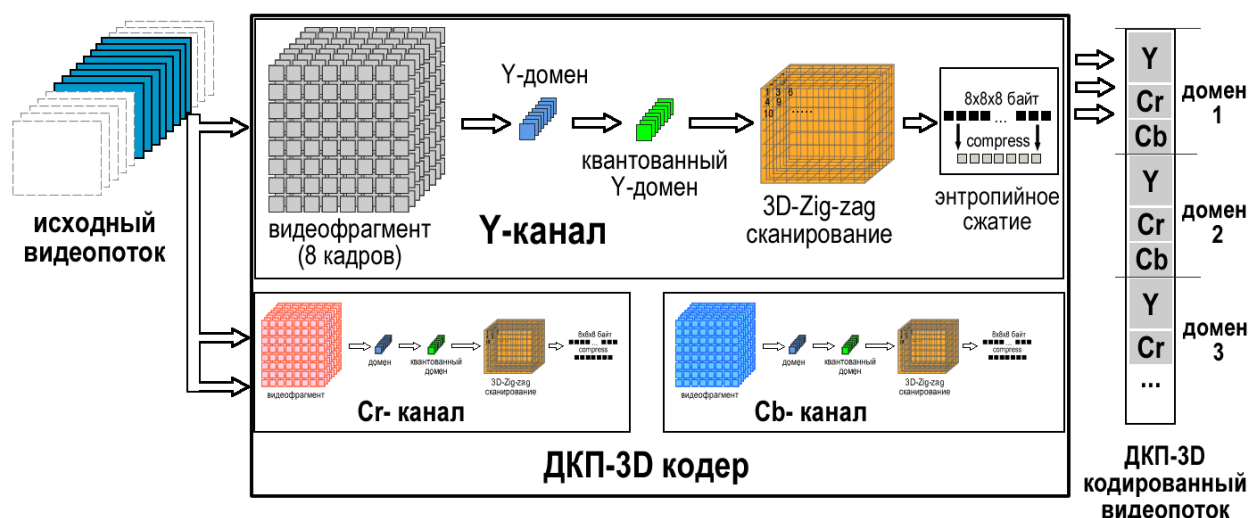


Рис. 1. Общая структура ДКП-3D видеокодека

В случае использования OpenMAX требуется работать с теми примитивами, которые представлены в конкретной реализации, что зачастую заставляет перерабатывать код, использующий нестандартную упаковку битов изображения. В данном случае, в связи с достаточно низким уровнем сложности кода и полной автономности различных частей базовой версии кодека, возможны оба варианта. В такой ситуации наиболее удобна и быстра двухэтапная схема:

- 1) С помощью OpenMAX-примитивов реализуются наиболее нагруженные участки кода (до 80% общей вычислительной мощности). Эту версию необходимо сохранить.
- 2) Дальнейшая оптимизация ведётся уже с использованием инструкций, скрываемых за OpenMAX-представлением.

«Чистая» OpenMAX-версия интересна тем, что возможна её работа в любом другом аппаратном окружении, производимое в действие путём простой перекомпиляции с использованием библиотек реализации стандарта нового целевого устройства. Разница лишь в том, что сами инструкции, в зависимости от возможностей платформы и имеющейся реализации, могут выполняться на SIMD-расширении, ЦОС-процессоре или же любом другом аппаратном ускорителе.

2.3. Базовая технология NEON. Технология NEON – 128-битное SIMD-расширение (1 инструкция/«много» данных) для процессоров ARM семейства Cortex-A8, спроектированное для проведения гибкой и мощной оптимизации. 10-стадийный конвейер обработки мультимедийных данных NEON позволяет серьёзно улучшить производительность в мультимедийных приложениях[2]:

- Возможность 60-150% выигрыша производительности в последних видеокодеках;
- В отдельных простых алгоритмах ЦОС, можно достичь 4-8 кратного ускорения;
- Уменьшение загрузки центрального процессора в пользу специализированных расширений, позволяет снизить общее энергопотребление.

Поскольку речь идёт в первую очередь о встраиваемых устройствах, данная технология является одной из главных приоритетных направлений оптимизации.

3.1. Профилирование 3D-DCT кода. Перед тем, как начать оптимизацию важно получить максимально точные данные о производительности, причём желательно иметь статистику с целевого устройства, для того, чтобы учитывать реальную латентность памяти. В данной статье мы использовали модуль ядра Linux, профилировщик OProfile[5] для оценки производительности. Модуль OProfile использует встроенные в процессор

аппаратные возможности мониторинга производительности и собирает сведения о ядре и исполняемых модулях, в частности, статистику использования памяти, число обращений к кэшу второго уровня и полученных аппаратных прерываний. Данный профилировщик поддерживает работу с ARM Cortex-A8 (профиль «armv7»).

В результате профилирования кода, без оптимизирующих инструкций, были получены результаты, представленные на рисунке 2:

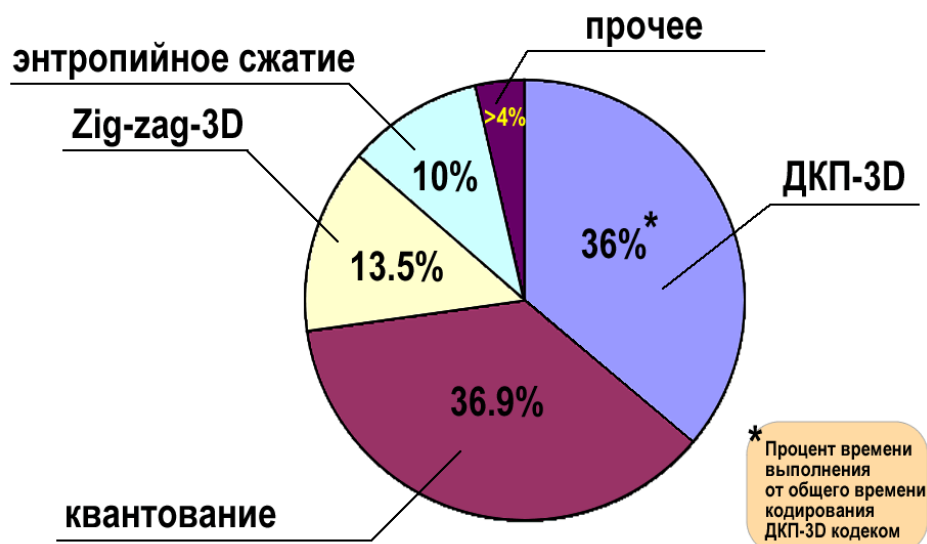


Рис. 2. Профилирование кода ДКП-3D кодека с помощью Oprofile для 480p видеофрагмента

Достаточно равномерное распределение времени работы функций кодека, в особенности ДКП-3D и квантование говорит о такой же равномерной степени оптимизации, что хорошо подходит для начала ввода примитивов.

3.2. Внесение OpenMAX DL-примитивов. Для ДКП-3D, как весьма близкого к JPEG, кодека наиболее используемыми являются функции, относящиеся к домену работы с изображениями:

- кодирование/декодирование JPEG, прямое и обратное ДКП 8x8, квантование;
- Объединённые функции ДКП & квантования, кодирования Хаффмана;
- Обработка изображений – цветовые преобразования, упаковка контейнеров;
- Удаление блочности, фильтры, копирование блоков, вращение, масштабирование.

Тем не менее, некоторые части домена работы с видео также возможно использовать при модификации и доработке кодека:

- 8x8/16X16 Сложение/вычитание;
- 8x8 (O)ДКП + (де)квантование + (обратное)сканирование;
- Декодирование длин серий (по стандарту MPEG-4).

Среди функций OpenMAX выполняющих наиболее значимые преобразования существует 2 версии, одна из которых отвечает за преобразование потока со смешанной структурой цветовых компонент (YCbCrYCb...), а вторая – за версию с плоским (planar) порядком размещения (YYYY...CrCr...CbCb...). В тестовом приложении использовалась плоский вариант размещения.

Инициализация была обеспечена OpenMAX-примитивами. Это обеспечило выполнение строгих требований по выравниванию данных и общей организации буфера. Кроме общих функций, таких как ДКП над вектором 8x8 можно получить дополнительный выигрыш в скорости, используя оптимизированные комбинированные функции, выполняющие сразу 2-3 преобразования. К таким функциям, например,

относится DCTQuantFwd_Multiple_S16. На рисунке 3 приведены примитивы, использованные в оптимизации:

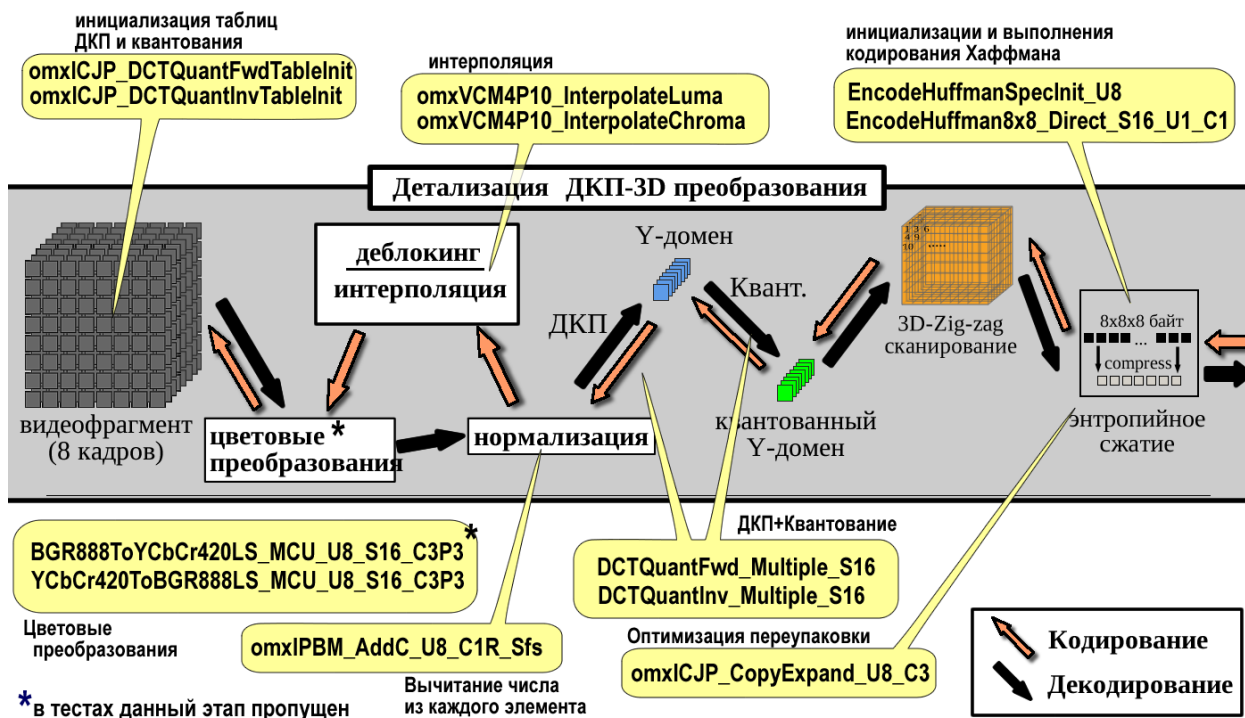


Рис. 3. Использование OpenMAX DL примитивов в тракте ДКП-3D видеокодека

После компиляции проекта и запуска на целевом устройстве с профайлером были получены результаты, представленные на рисунке 4. Слева изображена диаграмма нового распределения вычислительной нагрузки в кодеке. Справа – результаты тестового задания по кодированию небольшого видеофрагмента формата 480р с изменением сканирующего ДКП-окна и замером времени кодирования для OpenMAX DL-версии и обычной версии на языке Си.

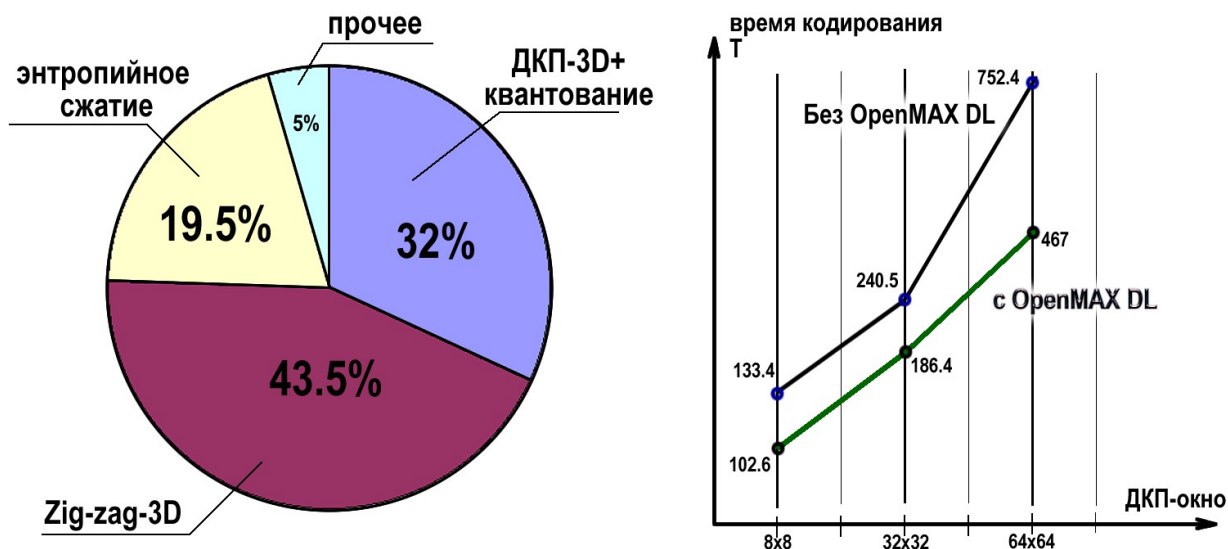


Рис. 4. Результаты профилировки и тестирования OpenMAX-версии

Декодирование с помощью OpenMAX позволило уже в первой версии в среднем на 28% повысить скорость выполнения. Среди дальнейших возможностей оптимизации

можно отметить более полное использование DL-примитивов, связанных с перемещением данных между буферами и использование функций домена VC (Video) – в частности, для уменьшения нагрузки от текущей наиболее трудоёмкой части кодека — Зиг-заг сканирования. Для этого могут подойти функции `omxVCM4P2_DecodeVLCZigzag_Inter` и `omxVCM4P2_DecodeVLCZigzag_IntraAC[DC]VLC`. Кроме этого также необходимо снижение общего количества транзакций и дальнейшая NEON-оптимизация через ассемблерные инструкции.

4. Выводы. Декодирование видео высокого качества по-прежнему относится к типу задач, которые являются сложными для современного мобильного устройства. 3D-DCT кодек, исследуемый в данной статье, позволяет снизить высокую планку производительности, которую установили кодеки стандарта MPEG-4. Благодаря использованию низкоуровневого интерфейса OpenMAX DL удалось достичь дополнительного прироста производительности в среднем 28 % при кодировании/декодировании видеоматериала.

Библиографический список

1. С. В. Очкур, Ш. С. Фахми. Универсальный видеокодек реального времени на основе трехмерного дискретного косинусного преобразования. 64-я научно-техническая конференция. Апрель 2009 г. Труды конференции. Стр. 41-42.
2. NEON. <http://www.arm.com/products/processors/technologies/neon.php> (дата обращения: 04.05.11).
3. OMAP3530/25 Applications Processor // DaVinci Digital Media Processor. [Электронный ресурс]. URL: <http://focus.ti.com/lit/ds/symlink/omap3530.pdf> (дата обращения: 02.05.11).
4. OpenMAX Cortex-A8 Release Note. OX002-GENC-008165 // <http://www.arm.com>. с. 9.
5. Oprofile. <http://oprofile.sourceforge.net/> (дата обращения: 04.05.11).
6. Rintaluoma T. Optimizing h.264 decoder for cortex-a8 with arm neon openmax dl implementation // IQ Magazine: Smart Approach to Designing with the ARM Architecture – 2009 - Volume 8, Number 2 – с. 32-36.

Optimization of DCT-3D videocodec for Cortex-A8 with ARM NEON OpenMAX DL

Ochkur S.V.

Saint Petersburg ETU, the Faculty of Radio Engineering and Telecommunications/Television and Videotechnics

This article discusses the questions of DCT-3D program videocodec optimization for wide variety of devices – from SIMD instruction set and DSP processors to specialized hardware accelerators. This is achieved through the program interface OpenMAX DL use with the application profiling on the ARM Cortex-A8 example.

Videocodec, DCT-3D, optimization, API, NEON, OpenMAX, development layer, profiling, Cortex-A8,