

Департамент образования города Москвы
Юго-Западное окружное управление образования
Государственное бюджетное образовательное учреждение
Лицей №1533 (информационных технологий)

Выпускной проект
по специальности «Технологии программирования»
учащегося группы 11.3
Медведева Алексея Вячеславовича
на тему
«Расширение функциональности системы
микширования видеопотоков для работы с видео высокого
разрешения»

Руководитель:	Завриев Н.К.
Консультант:	Завриев Н.К.
Заказчик:	Сумин Д.А.

Оглавление

1.	Введение и актуальность работы.....	3
2.	Постановка задачи.....	4
	Целевая аудитория.....	4
	Аналоги проекта.....	4
	Система видеоконференций ViDiNG.....	5
	Wirecast for Youtube.....	6
	DVSwitch	7
	Постановка задачи.....	11
3.	Решение.....	13
	Структура видеостудии.....	13
	Основные понятия, форматы данных и протоколы.....	14
	Телевидение высокой четкости.....	14
	Интерфейсы FireWire и HDMI.....	16
	Сжатие потока.....	16
	Различия между видеофайлом и потоковым видео.....	17
	Транспортный и прикладный протоколы и их особенности.....	18
	Программная реализация.....	20
	Архитектура и идеология программы	20
	Используемые протоколы.....	21
	Сторонние библиотеки и пакеты.....	22
	Логика работы программы.....	22
4.	Результат работы.....	25
	Пакет Videostudio.....	25
	Программный продукт Videostudio.....	25
	Эргономика.....	26
	Порядок работы программы.....	26
5.	Выводы и возможности дальнейшего развития	30
6.	Список литературы	32
	FFmpeg	32
	Aforge.NET	32
	Информация общего характера.....	32

1. Введение и актуальность работы

Развитие технических систем в последние десятилетия перевернули представления людей о том, какие сферы деятельности могут быть доступным только профессионалам, оснащенным дорогим специализированным оборудованием, а какие - широким массам. Особенно справедливо это для компьютерных систем. С распространением персональных компьютеров обычным людям и небольшим организациям стали доступны такие области деятельности, как издательская, полиграфическая, включая обработку изображений, видеомонтаж. На рынке представлено огромное разнообразие оборудования для видеосъемки и цифровой видеозаписи. Однако процесс организации видеотрансляций с использованием нескольких камер до сих пор чаще всего требует наличия дорогостоящего специального оборудования, а существующие программные пакеты, работающие на ПК, либо отличаются высокой стоимостью, либо имеют серьезные ограничения.

В частности, используемая в видеостудии Лицея система онлайн трансляций и микширования видео работает с видео стандартного разрешения SD, хотя в настоящее время актуальным является стандарт видео высокого разрешения. Сегодня абсолютное большинство пользователей имеют новые мониторы с разрешением HD, большинство камер также поддерживают формат HD. Видео высокого разрешения стало привычным посетителям интернет-сайтов: на Youtube и других сервисах можно найти огромное количество роликов, снятых пользователями в формате HD. Единственным «узким горлышком», ограничивающим ведение трансляций в HD-видео, остается этап видеорежиссуры, или микширования, т.е. выбор источника видеопотока.

В связи с этим возникла необходимость создания применимого на ПК программного комплекса, который бы обеспечивал проведение видеотрансляции и записи ее результатов в актуальных на сегодняшний день форматах: HD и Full HD. Такой комплекс позволил бы перевести трансляции на более высокий уровень качества, привычный современному пользователю.

2. Постановка задачи

Целевая аудитория

Целью данной работы было создание программного комплекса, позволяющего организовать микширование видеопотоков от множества источников (изображения с мониторов компьютеров, камер, подключенных как к компьютеру, на котором работает видеомикшер, так и к удаленным рабочим станциям, работающих в локальной сети организации, и т.д.) и трансляцию формируемой передачи в формате HD. Программный комплекс, работающий на ПК и не требующий дорогостоящего дополнительного оборудования, может быть полезен любым лицам или небольшим организациям, желающий проводить онлайн трансляцию через сеть интернет, например, при проведении каких-либо праздничных или обучающих мероприятий.

В том числе данная работа представляет непосредственный интерес и для созданной в 2007 году видеостудии лица. Ее задачей являются съемки и организация онлайн трансляций различных событий, проходящих в лицее: торжественных мероприятий, защит дипломных работ, концертов и т.д. В настоящее время для переключения между источниками видео, их быстрой режиссуры и дальнейшей передачи в сеть интернет для вещания через сайт лица видеостудия использует программный пульт DVSwitch, который устраивает по архитектуре и набору функций, но имеет устаревшие возможности с точки зрения - поддерживает трансляцию с качеством не выше SD.

Новый программный комплекс должен обеспечивать следующие дополнительные возможности:

- Реализацию захвата и обработки видео в формате HD с имеющихся в видеостудии видеокамер.
- Повышение качества записи и вещания видео до формата HD
- Увеличение количества поддерживаемых программным комплексом устройств захвата видео.

Аналоги проекта

Для организации прямого эфира профессиональные видеостудии и телеканалы используют дорогостоящие специализированные аппаратные системы. На рынке ПО Ближайшими аналогами создаваемого программного комплекса являются:

1. Система видеоконференций ViDiNG

2. Wirecast for youtube
3. DVswitch

Система видеоконференций ViDiNG

ViDiNG (<http://rdtel.ru>)- профессиональная российская система групповой видеоконференцсвязи (ВКС) высокой четкости, поддерживающая технологию серверного микширования видеоизображений. Продукт платный, ориентирован на использование в государственных учреждениях или крупных компаниях.

Главная особенность ViDiNG - микширование видеопотоков происходит на сервере, а не на оконечном оборудовании. В результате обеспечивается более высокое качество картинки и снижаются требования к мощности видеотерминалов. Кроме того, технология дает возможность подключения к ВКС видеотерминалов и серверов видеоконференций разнообразных производителей.

ViDiNG поддерживает широкий спектр протоколов и видеокодеков, предназначенных для достижения высокой степени сжатия видеопотока при сохранении высокого качества, а собственная доработанная версия видеокодека H.264 ViDiNG может обеспечивать организацию сеансов видеоконференцсвязи в качестве HD и Full HD. Система также поддерживает переключение между разными битрейтами в зависимости от ширины канала передачи данных каждого из участников видеоконференции.

Основные возможности ViDiNG:

- Многопользовательские видеоконференции (количество участников ограничено только мощностью серверного оборудования)
- HD/ Full HD
- Режим трансляции
- Показ презентаций и рабочего стола
- Видеотелефония
- Комнаты конференций
- Web-интерфейс с личным кабинетом
- Клиентские gui-приложения
- Запись конференций
- FTP для хранения записанных конференций и дополнительных материалов

ВКС способна подстраиваться под канал передачи данных таким образом, что в случае, если у одного из участников видеоконференции слабый Интернет, качество картинки, передаваемое им, ухудшается. Качество картинки пользователей, у которых нет

проблем с Интернетом, остается хорошим. Битрейт кодека изменяется динамически, в зависимости от ширины канала передачи данных каждого из участников видеоконференции.

Программное исполнение видеоконференц-системы ViDiNG не содержит ограничений на количество активных участников видеоконференций – оно ограничено только мощностью сервера и пропускной способностью канала передачи данных. Оператор может настраивать удобную раскладку окна видеоконференции, включать режим, когда в окне видеоконференции показываются приоритетные, либо активные участники видеоконференции. К конференции могут быть подключены несколько тысяч слушателей без нагрузки на сервер видеоконференцсвязи.

Продукт платный, стоимость конференции, в которой может участвовать до 6 пользователей - – 90 000 рублей. Для полного использования всех возможностей ВКС необходимы специально оснащенные конференц-комнаты. Т.е. продукт неприменим в бытовых условиях и в небольших организациях.

Wirecast for Youtube

Приложение Telestream Wirecast for Youtube ориентировано на проведение интернет-трансляций с нескольких камер, при этом оно обладает рядом дополнительных возможностей. Производитель приложения, компания Telestream, позиционирует программу, как "приложение для продвинутого вебкастинга".

Основные возможности приложения:

- Возможность подключения большого количества камер (производитель не устанавливает ограничения, количество ограничивается вычислительной мощностью ПК)
- Поддержка как DV, так и камер других стандартов, в том числе и веб камер
- Микширование видеопотоков в реальном времени
- Различные эффекты перехода
- Поддержка ChromaKey
- Запуск в эфир видеофайлов различных форматов
- Запись эфира в том же формате, в котором происходит вещание
- Поддержка вещания как по протоколу RTMP (Flash video с кодеком MMS (Windows Media Video), так и по протоколу MMS (Windows Media Video)
- Возможность наложения титров
- Встроенный кодировщик видео
- Поддержка HD Video

Wirecast - одно из наиболее функциональных решений для платформы Windows. Также это единственное решение, поддерживающее Macintosh (есть версии как для Windows, так и для Macintosh).

Программа позволяет вести вещание в нескольких форматах, но не поддерживается вещание с несколькими битрейтами одновременно. Для нормальной работы с HD видео полного разрешения 1920x1080 требуется очень большая вычислительная мощность, как и у любых других решений, кодирующих HD видео в реальном времени. Также при использовании HD увеличивается задержка - время от того как действие происходит в реальности, до того как его увидят зрители в сети. При стандартном разрешении до 720x576 задержка составляет до 2 сек, при использовании HD задержка составит около 5-7 секунд, что неприемлемо в ряде случаев - например, при проведении телемостов.

Стоимость приложения составляет 499\$, поддержка HD video приобретается отдельно и стоит 199\$.

DVSwitch

Пакет DVSwitch создан для построения программной видеостудии, реализующей возможности микширования аудио- и видео потоков в реальном времени, наложения эффектов, вещания видео в Интернет на вещающий сервер, наложения титров на стороне пользователя, взаимодействия с видеоархивом COBA. Пакет имеет модульную архитектуру, предназначен для работы под ОС Linux и состоит из четырех частей:

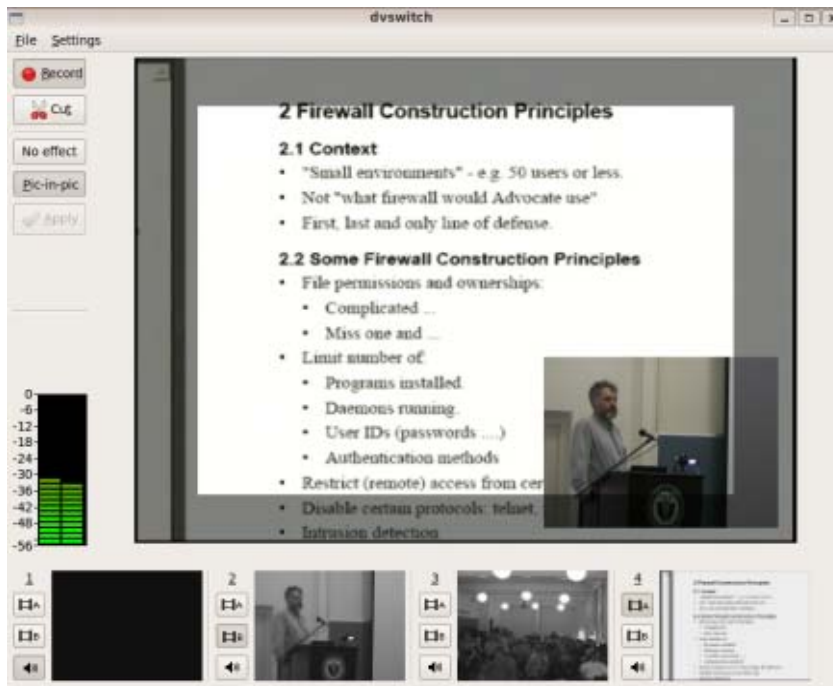
1. DVSwitch – сервер, работающий на компьютере режиссера эфира и отвечающий за микширование потоков;
2. DVSource – приложение, получающее поток с видеокамеры и передающее ее на DVSwitch по сети или в рамках одного компьютера;
3. DVSink – приложение, осуществляющее трансляцию потока по сети или запись в файл;
4. Модуль кодирования (используется внешняя программа - VLC Mediaplayer).

Каждый модуль представляет собой отдельное приложение, запускаемое в консольном режиме. Для выполнения части своих функций сами модули используют сторонние приложения или библиотеки, прежде всего – FFmpeg для Linux.

Функциональные возможности комплекса:

- отображение картинки с подключенных камер (локальных или сетевых);

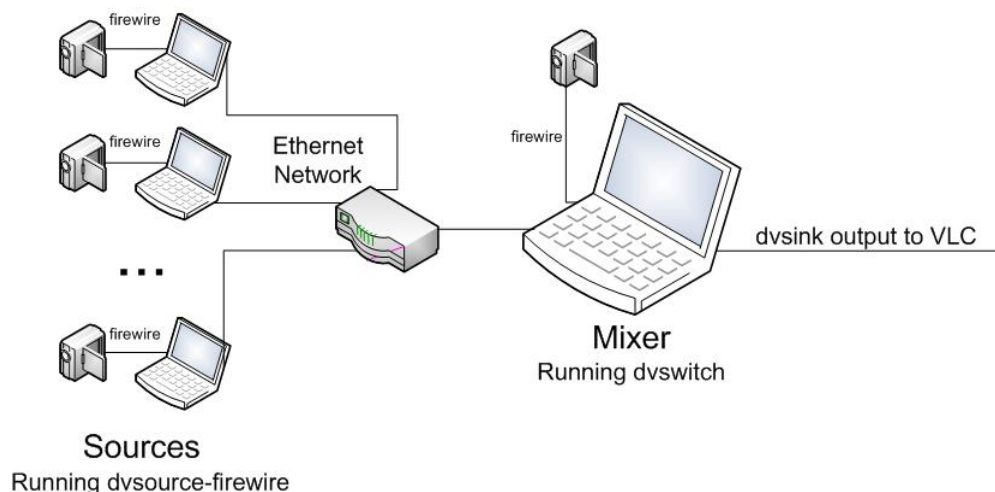
- коммутация аудио и видео потоков;
- эффект «Картинка в картинке»;
- включение, разделение (разбивка) на части и остановка записи микшированного потока.



Интерфейс программы DVSwitch

Данный проект представляет собой полнофункциональную цифровую видеостудию, работающую на обычных персональных компьютерах и ноутбуках. Таким образом, для решения задач видеостудии отпадает необходимость использовать дорогостоящее специализированное/профессиональное оборудование для работы с видеосигналом.

Основным модулем для работы с видео является DVSwitch. Он используется на компьютере режиссера эфира, осуществляет отображение картинки со всех подключенных камер, позволяет выбирать источник видео- и аудиосигнала сигнала для дальнейшей трансляции в режиме реального времени. Источником видеопотока могут быть как камеры, подключенные непосредственно к компьютеру режиссера эфира, так и удаленные, т.е. передающие поток по сети Ethernet. В обоих случаях захват видеосигнала производится модулем DVSource.



Комплекс DVSwitch работает исключительно с форматом DV, который поддерживают практически все существующие видеокамеры. Использование данного формата позволяет значительно сократить время задержки видео потока до микшера режиссера. Весь обмен данными между элементами комплекса происходит с использованием протокола TCP/IP.

При передаче данных с удаленной камеры для снижения нагрузки на канал производится кодирование и сжатие видеопотока в формат H264 путем обращения к стороне программы - медиаплееру VLC.

Для получения формируемого VLC видеопотока для каждой из удаленных камер на компьютере режиссера запускается модуль DVSink. Для передачи на сервер используется протокол RTP. На данный момент модуль позволяет в режиме реального времени отправлять в эфир картинку с одной из подключенных камер, а звук может браться из другого потока или же вообще вводиться через звуковую карту ПК. Приложение работает с форматом DV.

Данный пакет используется в видеостудии лица. На протяжении нескольких лет комплекс DVSwitch показывает себя как достаточно стабильная система, которая при этом достаточно мобильная и может быть запущена практически на любом установленном в лице персональном компьютере. Одной из ключевых причин использования данного комплекса является простота управления им, которая позволяет в течение нескольких минут обучить любого сотрудника основам работы видеорежиссера.

Все компьютеры системы должны работать под ОС Linux. Для устойчивой работы необходимо иметь одинаковый формат картинки со всех камер. Также важно чтобы частота дискретизации звука на всех камерах была выставлена одинаковой (32, 44.1, 48 кГц). DVSwitch автоматически определяет формат, который использует первый источник, а для источников, которые не соответствуют ему будет выдано сообщение об ошибке.

Эксплуатация DVSwitch выявила проблему разрыва связи с источником. В случае отключения одного из источников, будь то нечаянно вырванный кабель, плохой контакт или зависании одного из модулей программы, его невозможно переподключить в горячем режиме, и данную проблему можно решить только путем перезапуска микшера, что недопустимо во время трансляции. В свою очередь, перезапуск микшера чаще всего приводит к остановке всех остальных систем, и требует перезагрузке всех многочисленных компонентов, что приводит к длительному прерыванию трансляции.

Комплекс DVSwitch распространяется с открытым исходным кодом, что формально дает возможность его доработки. Однако для этого необходимо очень хорошо понимать архитектуру системы, схему взаимодействия между модулями данного комплекса, знать языки C и C++. Модификацию кода осложняет то, что комплекс написан с использованием устаревшей среды программирования.

Практика проведения видеотрансляций показала, что звуковой канал чаще всего не связан с видеотрактом: например, при демонстрации концерта смена режиссером камеры-источника не должна влиять на транслируемый звуковой ряд. Поэтому микширование звукового сигнала удобнее делать на специализированном оборудовании, имеющем необходимый функционал (поддержка разного числа каналов, микширование, реверберация, включение аудиотреков для поддержки выступления артистов и т.д.). Чаще всего для работы со звуком используется микшерный пульт, которым управляет режиссер трансляции или, в случае использования большого количества источников звука, звукорежиссер. Смешанный звуковой поток подключается к компьютеру режиссера и указывается как микрофон или линейный вход.

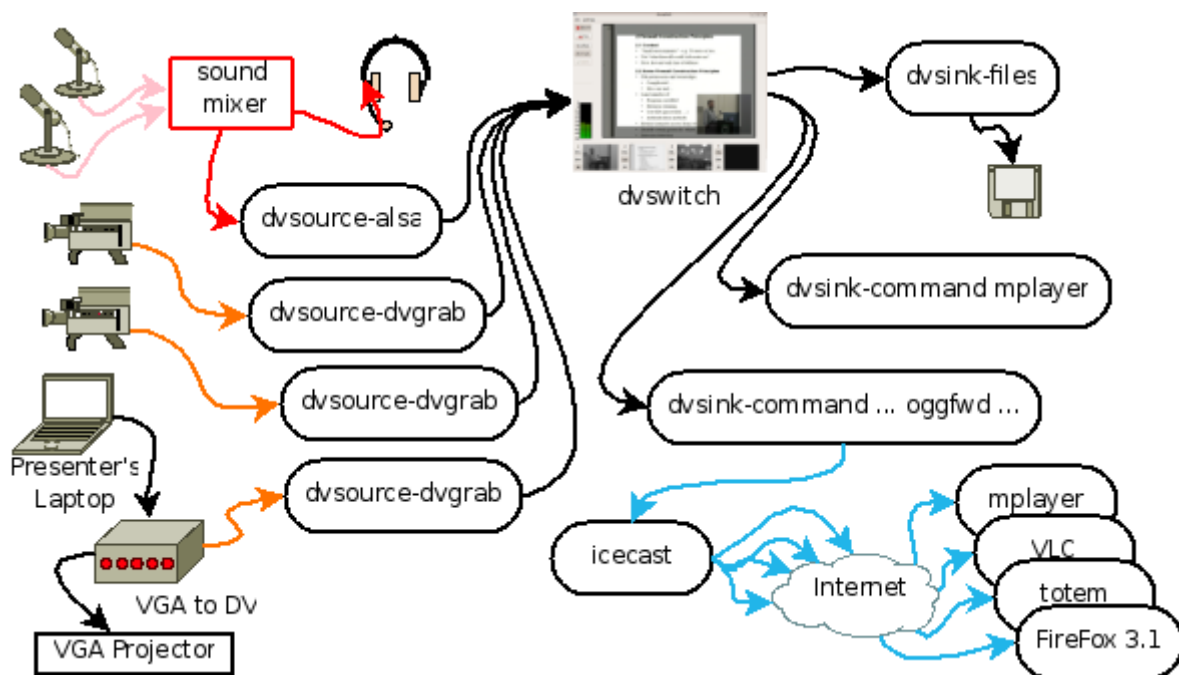


СХЕМА РАБОТЫ ПРОГРАММЫ DVSWITCH

В целом комплекс DVSwitch успешно используется в видеостудии как наиболее полно отвечающий задачам организации видеотрансляций различных мероприятий без использования специализированного профессионального оборудования или дорогих программных комплексов.

Основные недостатки DVSwitch заключаются в следующем

- Комплекс использует подключение видеокамер по FireWire, что в силу пропускной способности шины (до 400 Мбит/с) ограничивает как разрешение камер, так и длину кабеля (в стандарте IEEE 1394 - до 4,5 м без дополнительных усилителей);
- Максимальное поддерживаемое разрешение – 720x576
- При передаче данных между удаленной камерой и режиссерским компьютером используется протокол TCP/IP, что в случае разрыва канала по любой причине приводит к остановке всех компонентов, и необходимости их дальнейшей перезагрузки;
- Требуется перезапуск программы для подключения новых или переподключенных источников видео.

Постановка задачи

Благодаря прогрессу и удешевлению абонентского оборудования сегодня большинству пользователей интернет доступны мониторы и компьютеры, способные обеспечивать воспроизведение видео в формате высокой четкости. Такое изображение выглядит

намного более реалистичным, лучше воспринимается сознанием и вызывает меньшее напряжение зрительной системы человека. В связи с этим возникла задача создания вещательного сервера как элемента распределенной вычислительно-коммуникационной глобальной сети, обеспечивающего проведение видеотрансляций в сети интернет с уровнем качества изображения, соответствующего стандарту HD на базе бытового оборудования.

При разработке такого комплекса были предъявлены следующие требования:

- Способность работать с бытовым оборудованием (подключение камер по интерфейсу IEEE 1394, использование Ethernet для передачи DV-видео с камер в реальном времени, коммутация потоков с камер программным микшером, кодирование в форматы, пригодные для передачи на сервер, передача видео на сервер, раздача видео с сервера клиентам);
- Возможность подключения не менее 4-х источников видео;
- Возможность работы с потоками видео в формате HDV;
- Организация обработки и сжатия потоков видео для корректной работы в реальном времени.
- Добавление возможности работы с потоками видео в популярных форматах.

3. Решение

Структура видеостудии

Видеотрансляция позволяет людям, непосредственно не присутствующим на месте события, увидеть происходящее. Задача видеотрансляции заключается в получении изображения события с различных точек съемки, выборе приоритетных в конкретный момент времени планов, дополнении их различными материалами (видеороликами и т.п.), и передаче потребителю. В общем случае эта задача решается видеостудией, представляющей собой комплекс из четырех компонент:

- Источники сигнала, которыми могут быть:
 - видеокамеры различных типов, использующие различные форматы данных, протоколы и физические интерфейсы;
 - визуальный образ рабочего стола компьютера;
 - видеофайлы различных форматов;
 - видеопотоки, передаваемые по локальной сети или через интернет;
- Коммутатор (или видеомикшер), посредством которого осуществляется выбор источника сигнала для дальнейшей трансляции, преобразование сигнала, в т.ч. с использованием различных эффектов;
- Блок вещания (сервер вещания), отвечающий за дальнейшую трансляцию скоммутированного и обработанного сигнала по эфирным и кабельным каналам связи;
- Блок записи, обеспечивающий сохранение итоговой трансляции или ряда исходных потоков в виде видеофайлов для дальнейшей обработки и использования, в т.ч. внешними пользователями, не имеющими специального оборудования и программного обеспечения.

В случае интернет-трансляции передача информации происходит с помощью компьютерного оборудования и сети интернет. Использование сервера вещания необходимо для обеспечения возможности подключения большого количества пользователей для просмотра видео.

Каждый из перечисленных компонентов видеостудии может быть решен в виде аппаратного устройства или программы. В профессиональных студиях используются преимущественно специализированные аппаратные решения, созданные на базе

специализированных высокопроизводительных процессоров и обладающие высокой производительностью на конкретном наборе задач. Минусом данного оборудования является высокая цена и сложность использования. С развитием персональных компьютеров практически все задачи такого оборудования могут быть решены на обычном персональном компьютере или ноутбуке. Это позволяет создать компактные видеостудии, состоящие из минимального набора компонентов и не требующие профессионального персонала.

Ключевыми параметрами, определяющими функционал и производительность видеостудии, являются протоколы и форматы данных, используемые для передачи данных между ее компонентами. При этом для коммутации каждой пары компонентов используются свой набор стандартов.

Основные понятия, форматы данных и протоколы

Телевидение высокой четкости

Большинство видеокамер, являющихся основным источником видеосигнала, работают в одном из двух форматов – SD или HD.

Стандарт SD (Standard Definition – Стандартное разрешение) относится к категории «традиционных» старых систем видеопросмотра, хранения и передачи сигналов – таких как PAL, SECAM и NTSC, появившихся вместе с цветным телевидением в 1950-е годы. Когда появилось цифровое видео (DV - Digital Video), эти традиционные системы были представлены с помощью пиксельной решетки, преобразующей содержимое аналогового видео в цифровой сигнал. В связи со своей «традиционностью», данные системы ограничены частотой обновления кадров, характерной для PAL и NTSC (25 кадров/с и 29,97 кадров/с соответственно), низкой полосой частот (количество информации пропускаемое за единицу времени) в диапазоне 4 — 6 МГц, а также ограничены количеством строк развертки (625 и 525 соответственно для PAL и NTSC).

Позже появился стандарт (HD или HDTV, сокр. от англ. high definition television), телевидение высокой чёткости - это система телевидения, позволяющая наблюдателю со средней остротой зрения смотреть картинку с расстояния трех высот изображения и не замечать при этом строчную структуру изображения.

Современное цифровое телевидение высокой чёткости основано на рекомендации ITU-R BT.709 Международного союза электросвязи и обеспечивает соотношение сторон экрана 16:9 с разрешением 1920×1080 пикселей. Такое телевизионное изображение, в зависимости от типа развертки, называется 1080i (чересстрочная) или 1080p

(прогрессивная, т.е. последовательная). Российским национальным стандартом, определяющим основные параметры телевидения высокой четкости, является ГОСТ Р 53533-2009.

Корпоративные и национальные стандарты HDTV отличаются от российского. Некоторые американские и европейские телевизионные компании практикуют рекомендованное Европейским вещательным союзом изображение 720p (разрешение 1280×720 с прогрессивной разверткой), которое в России считается телевидением повышенной четкости.

Разновидностями HDTV являются стандарт ATSC (1080i) с чересстрочной разверткой, предусмотренный для сетки вплоть до 1920×1080 пикселей. Часть аппаратуры использует 1440 более «широких» пикселей (1440×1080), которые смотрятся, однако, значительно лучше, чем широкий экран стандарта SD.

Частота обновления изображения в HD варьируется, но для 720p обычно составляет 24p (что очень похоже на кино), 25p, 30p, 50p и 60p для прогрессивного видео (полный кадр), и 25i, 30i, 50i и 60i для 1080i (чересстрочное видео).

Сигнал HDTV несет в себе значительное количество информации, что требует как мощного оборудования для кодирования/ декодирования, так и каналов передачи данных с высокой пропускной способностью. Уровень требований хорошо виден из таблиц 1 и 2.

Таблица 1. Аппаратная нагрузка при работе с HD TV

	720p (HDV-1)	1080i (HDV-2)
Носитель данных	Кассета DV	Кассета DV
Разрешение/ частота кадров	720/25p, 720/50p, 720/30p, 720/60p	1080/50i и 1080/60i
Эффективное число пикселей	1280 x 720	1440 x 1080
Относительная ширина экрана	16 : 9	16 : 9
Сжатие	MPEG2@H-14	MPEG2@H-14
Частота дискретизации	75.25 MHz	55.6875 MHz
Схема цветовой дискретизации	4:2:0	4:2:0
Дискретизация	8 бит	8 бит
Скорость передачи данных	19 Мбит/с	25 Мбит/с

Источник: http://www.studio-videon.ru/Theory/HDTV/HDTV_ukovodstvo.html

Таблица 2. Сравнение форматов HDTV

Форматы	Разрешение	Число кадров/с	Число пикселей в секунду	Скорость передачи для MPEG4/H.264
720/24p или 25p		24/25	23 040 000	4-5 Мбит/с
720/50p	1280 x 720	50	46 080 000	7-9 Мбит/с
1080/50i	1920 x 1080 (2 x 540)	50	51 840 000	8-10 Мбит/с
1080/24p или 25p	1920 x 1080	24/25	51 840 000	7-9 Мбит/с
1080/50p	1920 x 1080	50	103 680 000	14-18 Мбит/с

Источник: <http://www.era-tv.ru/hdtv/slovar.asp>

Даже при использовании сжатия MPEG2, HD-каналы требуют примерно вчетверо больше передающей мощности по сравнению с обычными каналами.

Интерфейсы FireWire и HDMI

Физическое соединение камер с другими устройствами (т.е. соединение компонент в связке «Источник сигнала – Видеомикшер») производится с помощью интерфейсов FireWire или HDMI.

FireWire (IEEE 1394) — последовательная высокоскоростная шина, предназначенная для обмена цифровой информацией между компьютером и другими электронными устройствами. Скорость передачи данных — 100, 200 и 400 Мбит/с в стандарте IEEE 1394/1394a, 800 и 1600 Мбит/с в стандарте IEEE 1394b и 3200 Мбит/с в спецификации S3200.

HDMI (High Definition Multimedia Interface) — интерфейс для мультимедиа высокой чёткости, позволяющий передавать цифровые видеоданные высокого разрешения и многоканальные цифровые аудиосигналы с защитой от копирования. Разъём HDMI обеспечивает соединение нескольких устройств с помощью соответствующих кабелей и имеет пропускную способность в пределах от 4,9 до 10,2 Гбит/с.

Сжатие потока

В связи с большим объемом информации передача видеосигнала высокой четкости на дальние расстояния осуществляется, как правило, в сжатом цифровом виде. Аналогично, для оптимизации дискового пространства видеоданные хранятся также в

сжатом виде. Таким образом, важнейшими функциями сервера вещания и блока записи являются сжатие видео и (в случае сервера вещание) передача видеопотока в одном из принятых стандартов в сеть интернет. Сжатие видео снижает требования к ширине канала передачи (с 1,485 Гбит/с до 8—25 Мбит/с), при этом качество изображения остаётся приемлемым. Для кодирования видеосигнала высокой четкости наиболее часто используются форматы MPEG-2, MPEG-4.

MPEG — название группы стандартов цифрового кодирования видео- и аудио сигналов, организации транспортных потоков видео и аудио информации, передачи сопутствующей информации. Их применение позволяет использовать для передачи контента практически любой цифровой канал достаточной ширины (15—25 Мбит/с для MPEG-2 или 8—12 Мбит/с для MPEG-4 в зависимости от степени сжатия).

В настоящее время большее распространение получили программные кодировщики видео. Для их работы требуется только персональный компьютер достаточной мощности. Практически любой программный кодировщик может использоваться как с программными, так и с аппаратными микшерами. При использовании аппаратного микшера сведенный поток выводится из микшера по какому-либо интерфейсу и заводится в компьютер, где это видео обрабатывается программой кодирования и отправляется в Интернет. Самыми распространенными и продвинутыми программными кодировщиками на данный момент являются такие приложения, как:

- VideoLAN (VLC)
- FFmpeg

По функционалу, работа этих приложений аналогична аппаратным кодировщикам. Максимальное поддерживаемое разрешение и скорость смены кадров напрямую зависит от вычислительной мощности компьютера. Программные кодировщики могут поддерживать несколько кодеков для потоков видео и аудио. Для видео это в основном кодеки h264 или VP6, а для звука - mp3 или AAC. Закодированное видео чаще всего отправляется на сервер посредством протоколов RTP.

Различия между видеофайлом и потоковым видео

Потоковым видео (от англ. Streaming Video) называют технологии сжатия, передачи и буферизации данных, которые позволяют передавать видео в реальном времени через интернет.

Существует два способа передачи потокового видео: последовательный (progressive streaming) и в реальном времени (real-time streaming). При первом способе передачи видео постепенно загружается на локальный диск, и пользователь может просмотреть его по

мере закладки. Таким образом, чтобы просмотреть какую-нибудь часть, необходимо дождаться загрузки нужного момента. Поэтому последовательная передача видео через интернет применяется в основном для коротких клипов и показа полностью сохраненных на сервере видеороликов. Во втором случае компьютер-приемник отображает поток непосредственно по мере его формирования (поступления). Для осуществления передачи видео в реальном времени потребуются специальный сервер потокового вещания (streaming server).

Транспортный и прикладный протоколы и их особенности

Сжатый (закодированный) контент представляет собой поток данных определенного формата. Передача его между компьютерами определяется протоколами различных уровней. Так, протоколы транспортного уровня обеспечивают передачу данных от источника к получателю (tcp/ip, udp), а протоколы прикладного уровня – взаимодействие пользователя с сетью, позволяет приложениям иметь доступ к сетевым службам, таким как обработчик запросов к базам данных, доступ к файлам, пересылке электронной почты (http, ftp).

Среди транспортных среди которых выделяют надежные (гарантирующие доставку каждого пакета данных от отправителя получателю путем уведомления о получении и многократной отправки, хотя бы и ценой задержки) и ненадежные, в которых отправитель не контролирует и не гарантирует доставку пакета получателю.

При сжатии и передаче сигнала возникает задержка, вызванная следующими причинами:

- Затраты времени на обработку данных (чем алгоритм сложнее и плотнее сжимает данные, тем медленнее он работает и большую задержку вызывает);
- Затраты времени на пересылку данных по каналу передачи (время передачи пропорционально объему данных и обратно пропорционально скорости канала, а также количеству служебной информации и требования к надежности передачи);
- Буферизация данных (в ряде алгоритмов сжатие основано на анализе нескольких идущих подряд кадров, что требует ожидания получения всей закодированной последовательности; некоторые транспортные протоколы в случае пропуска пакета на стороне приемника накапливают последующие пакеты, дожидаясь «потерянный», и лишь потом передают все пакеты прикладному протоколу).

В зависимости от используемых алгоритмов и различных параметров, уровень задержки сигнала может составлять 1—10 секунд.

Приложения, формирующие потоки реального времени, требуют своевременной доставки информации и для достижения этой цели могут допустить некоторую потерю пакетов. Например, для аудио-приложения потеря пакета будет означать лишь к пропуску небольшой части звукового ряда, что предпочтительнее, чем ожидание очередного пакета, способного вызвать рассогласование видео- и аудиоряда. Чаще всего в качестве транспортных протоколов используются TCP и UDP.

TCP (англ. Transmission Control Protocol, протокол управления передачей) — один из основных протоколов передачи данных интернета, предназначенный для управления передачей данных в сетях и подсетях TCP/IP. Механизм TCP осуществляет повторный запрос данных в случае потери пакета и устраняет дублирование при получении двух копий одного пакета, гарантируя тем самым, в отличие от UDP, целостность передаваемых данных и уведомление отправителя о результатах передачи.

UDP (англ. User Datagram Protocol — протокол пользовательских датаграмм) использует модель передачи без обеспечения надёжности, упорядочивания или целостности данных. Датаграммы могут прийти не по порядку, дублироваться или вовсе исчезнуть без следа. UDP подразумевает, что проверка ошибок и исправление либо не нужны, либо должны исполняться в приложении. Чувствительные ко времени приложения часто используют UDP, так как предпочтительнее сбросить пакеты, чем ждать задержавшиеся, что может оказаться невозможным в системах реального времени.

Различия между этими протоколами в том, что TCP/IP обеспечивает надёжную доставку, а UDP - нет, поскольку TCP имеет встроенные механизмы контроля доставки и целостности данных. Однако TCP нельзя назвать лучшим решением для передачи мультимедиа, поскольку этот протокол добавляет в пакеты данных большое количество служебной информации. Для TCP главное - безошибочно передать данные, а время доставки вторично. С другой стороны, UDP использует гораздо меньше служебных данных, чем TCP, поэтому он лучше подходит для приложений, работающих с потоковыми данными, где на первый план выходит время доставки информации. Что касается пропусков и искажений пакетов, то решение этой проблемы возлагается на принимающую сторону

На более высоком уровне — на уровне прикладных программ — для передачи видео используется группа протоколов RTP. В частности, при передаче трафика реального времени используется RTP (англ. Real-time Transport Protocol). Протокол RTP переносит в своём заголовке данные, необходимые для восстановления аудиоданных или видеоизображения в приёмном узле, а также данные о типе кодирования информации

(JPEG, MPEG и т. п.). В заголовке данного протокола, в частности, передаются временная метка и номер пакета. Эти параметры позволяют при минимальных задержках определить порядок и момент декодирования каждого пакета, а также интерполировать потерянные пакеты.

RTSP (Real Time Streaming Protocol) - Поточковый протокол реального времени, является прикладным протоколом, предназначенным для использования в системах, работающих с мультимедиа данными, и позволяющий клиенту удалённо управлять потоком данных с сервера, предоставляя возможность выполнения команд, таких как «Старт», «Стоп», а также доступа по времени к файлам, расположенным на сервере.

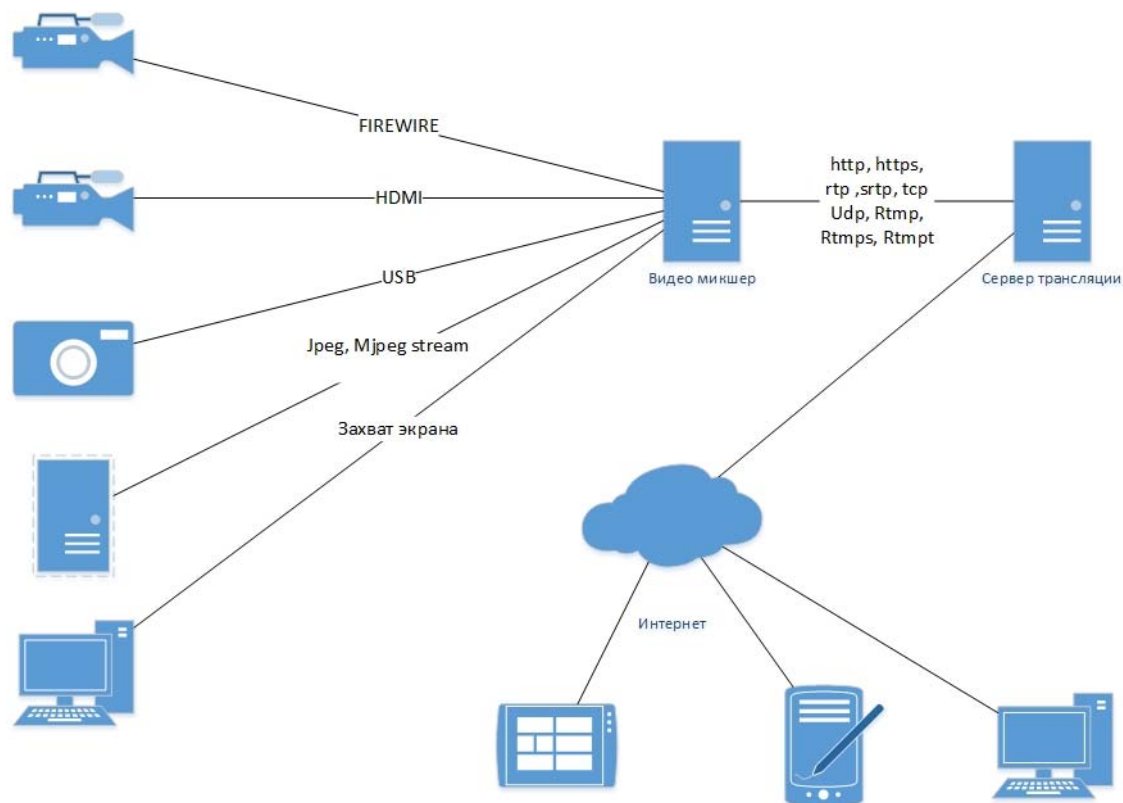
Как правило, протокол TCP не используется в RTP-приложениях, так как надёжность передачи в TCP вызывает временные задержки. Вместо этого большинство реализаций RTP базируется на UDP.

Наиболее привычным нам по работе в интернет протоколом является HTTP (англ. HyperText Transfer Protocol — «протокол передачи гипертекста»). Основой HTTP является технология «клиент-сервер», то есть предполагается существование потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом. Первые версии HTTP были предназначены для работы с гипертекстом, сейчас протокол позволяет передавать широкий спектр данных.

Программная реализация

Архитектура и идеология программы

Архитектура и идеология работы программы Videostudio повторяет структуру типичной видеостудии. Она содержит возможности для получения видеопотока с USB-вебкамер, видеофайлов в форматах MP4, JPEG и MJPEG потоков, а также с любых плат захвата, которые определяются операционной системой, как “Звуковые, игровые и видеоустройства”. Программа позволяет микшировать видеопотоки, создавать эффект картинка в картинке, транслировать изображение в сеть и записи видеопотока в файл.



АРХИТЕКТУРА РАБОТЫ КОМПЛЕКСА

С точки зрения внутренней структуры программа содержит:

- 6 однотипных объектов, отвечающих за настройку видеоисточника и работу с ним;
- Объект, отвечающий за преобразование формата видеопотока, выбор видеопотока для трансляции, его отображение для целей мониторинга, конвертацию видеопотока в заданный пользователем выходной формат, отправку потока на сервер трансляции;
- Сервер трансляции, отвечающий за дальнейшую трансляцию видеопотока в интернет.

Используемые протоколы

Для передачи изображения в рамках одного компьютера между программой и FFmpeg был выбран протокол TCP/IP, так как по нему можно передавать несжатую картинку, а потери внутри локальной машины практически невозможны. TCP обеспечивает сохранение правильной последовательность и целостность видео ряда на этом этапе.

Для передачи потока по локальной сети используется протокол UDP, который обеспечивает лучшие временных характеристики.

При наличии достаточных познаний, программа позволяет пользователю самому выбирать широкий спектр протоколов для дальнейшей передачи данных: https, rtp, srtp, tcp, udp, rtmp и другие.

Для записи всех видеопотоков используется кодек MPEG-2 и контейнер .avi. Разработчик не рекомендует использовать одновременно запись и трансляцию видео потока так как быстродействия большинства персональных компьютеров, а тем более ноутбуков, недостаточно для декодирования 3-4 потоков видео и сжатия еще нескольких.

Сторонние библиотеки и пакеты

При создании программы были использованы следующие библиотеки и пакеты:

AForge.Net – обеспечивает захват и декодирование видеопотока, а также запись потока в файл. Данная библиотека использована для того, чтобы унифицировать в программе обращения к различным источникам видео, имеющим различный характер: поток с видеокамер, видеофайлы, захват изображения с монитора ПК. Эта библиотека выполняет роль интерфейса между программой и DirectShow и FFmpeg.

DirectShow — мультимедийный фреймворк и интерфейс программирования приложений(API), позволяющий Windows-приложениям управлять широким спектром устройств аудио/видео ввода-вывода, включая DV- и веб-камеры, DVD-устройства, ТВ-тюнеры и др. Поддерживает также различные форматы файлов, от WAV и AVI до Windows Media.

FFmpeg – свободно распространяемый пакет программ для конвертации видео, трансляции в интернет и записи видеопотока в файл. Использование FFmpeg необходимо для того, чтобы избежать проблемы лицензионного использования форматов Mpeg (большинство пригодных для C# библиотек лицензионные), а также расширить функциональность программы.

Логика работы программы

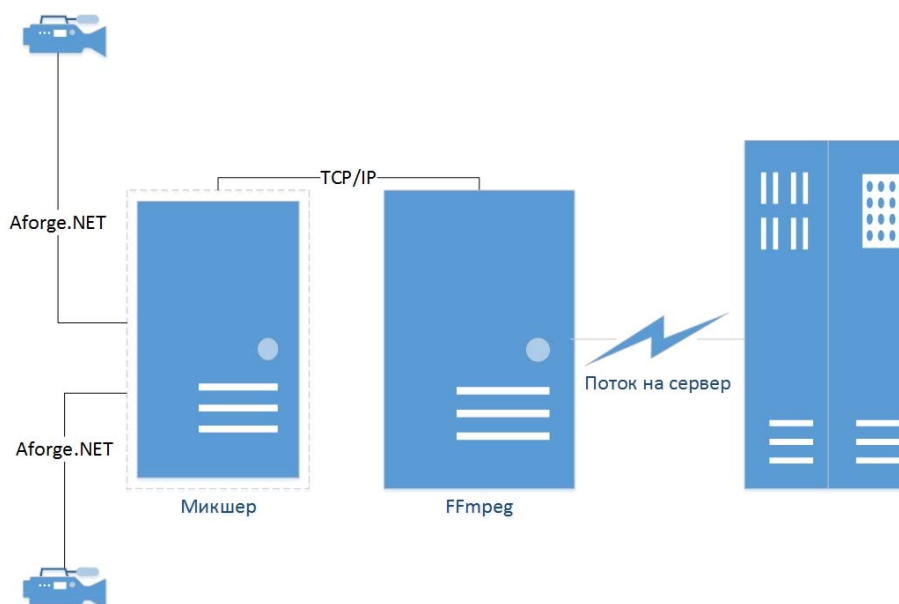
При запуске программы создается шесть независимых друг от друга объектов, которые отвечают за настройку видео источника и работу с ним. Каждый объект отображается в виде окна предпросмотра с кнопками On Air, Sound, Picture-in-Picture, Setup. При нажатии кнопки Setup отрывается окно настройки источника, которое позволяет выбрать устройство, с которого мы хотим получить видео, и задать основные параметры. При выборе захвата с камеры библиотека AForge.NET выполняет сканирование всей операционной системы в поисках доступного видеоприбора, а затем при выборе желаемого устройства возвращает его уникальный индикатор. При выборе других типов источников AForge.NET запрашивает путь к файлу или потоку. При

закрытии окна настроек происходит проверка корректности введенных данных, и после этого происходит подключение к источнику.

После подключения к источнику программа подписывается на событие «появление нового кадра». При получении очередного кадра обновляется окно предпросмотра данного источника, а также сохраняем новый кадр в замен предыдущего.

Примерно тридцать раз в секунду программа обновляет окно отображения микшированного потока, которое проверяет номер устройства, которое хочет видеть режиссер в данный момент, и запрашивает у объекта с данным номером его текущее изображение. Следующим шагом программа определяет включена ли функции «Картинка в картинке», и если да, то также запрашивает картинку и со второго устройства. Затем программа масштабирует основную картинку под предварительно указанное выходное разрешение, и накладывает поверх нее изображение со второго устройства, если это необходимо. После получения результирующего изображения программа отправляет его на показ режиссеру, на запись в файл, и для трансляции в FFmpeg.

При основной настройке программы и отмеченном пункте онлайн трансляции в фоновом режиме запускается пакет FFmpeg с параметрами, указанными при настройке. FFmpeg подключается к программе по протоколу TCP/IP на порт, указанный при настройках. После успешного запуска картинка отправляется в FFmpeg для сжатия и пересылки на сервер трансляции.



При разработке данной программы была обнаружено несколько серьезных проблем. Одной из них была проблема системного таймера. Семейство операционных систем Windows обрабатывает системные прерывания каждые 15,6 мс, а для работы с видео необходима частота 25, 30 кадров в секунду. Значит, для качественной работы с видео в реальном времени необходимо иметь системные прерывания каждые 40, 33 мс, а не 31 и 46 мс.

Следующей проблемой стал высокий трафик в сети при передаче сигнала HDTV. Например, при передаче несжатого потока с разрешением 1280x720 возникает трафик порядка 500мб/с, что недопустимо для использования во многих существующих сетях. Для решения этой проблемы было принято решение передавать поток из программы на пакет FFmpeg, запущенный на том же компьютере, для сжатия и дальнейшей передачи контента по сети. Передача из программы в FFmpeg происходит по протоколу TCP, далее – по любому протоколу, заданному пользователем в окне настройки выходного потока. Используя возможности FFmpeg, пользователь может выбрать широкий спектр протоколов, методов кодировки, форматов изображения и т.д., что значительно добавляет гибкости программе. Например, FFmpeg поддерживает отправку видео по таким протоколам, как http, httpproxy, https, rtp, srtp, tcp, udp,rtmp и другим.

Для реализации программы был выбран язык C#, поскольку он предоставляет удобные средства по работе с изображениями, а также позволяет реализовывать программы для платформы Windows, на которой работают большинство ПК. Для выполнения отдельных задач используется пакет FFmpeg, а также медиаплеер VLC, что обеспечивает совместимость с существующим ПО видеостудии лица.

Использование FFmpeg и VLC позволяет организовать каскадное подключение компьютеров, имеющих пакет Videostudio, и обеспечивать видеотрансляцию с большого числа камер, в том числе удаленных и находящихся в разных помещениях.

4. Результат работы

Пакет Videostudio

В результате работы над дипломным проектом был создан программный продукт – Videostudio 1.0, который выполняет все поставленные задачи, а именно:

- Захватывать видео с любых устройств, распознаваемых ОС Windows как видеоустройства (встроенные камеры, подключенные по USB, FireWier, к платам видеозахвата, HDMI);
- Использовать в рамках одной трансляции источники видеопотока с различными характеристиками (например, камеры с различным разрешением, битрейтом и т.д.), обеспечивая при этом нормализацию выходного видеопотока по параметрам, указанными пользователем в окне настроек;
- Захватывать изображение с рабочего стола режиссера;
- В процессе трансляции отключать устройства-источники и подключать новые без прерывания трансляции;
- Реализовывать эффект «картинка в картинке»;
- Комбинировать несколько видеомикшеров между собой, подключая их на вход друг друга, создавая за счет этого необходимую конфигурацию оборудования;
- Обеспечивать трансляцию видео в сеть интернет по таким протоколам, как http, httpproxy, https, rtp, srtp, tcp, udp,rtmp. Для обеспечения доступа к трансляции большего числа пользователей возможно перенаправление потока на такие распространенные серверы видеовещания как Adobe Flash Media Server, Wowza Mediaserver Pro, Red5 и другие.
- Записывать видео в файл формата Avi.

Программа написана с использованием объектно-ориентированного подхода, что облегчит ее модернизацию в дальнейшем.

Данная программа представляет собой работоспособное приложение и может использоваться для организации видеотрансляций с различным уровнем качества, достижимым с учетом характеристик используемого оборудования.

Программный продукт Videostudio

Videostudio 1.0 представляет собой программный продукт, т.е. программу, которую независимо от ее разработчиков можно использовать в предусмотренных целях на разных компьютерах, удовлетворяющих указанным в описании системным требованиям.

В соответствии с требованиями к программному продукту, он имеет дистрибутив, сформированный на носителе USB-флеш и включающий пакет установки и документацию. Документация описывает порядок установки пакета на компьютер пользователя и порядок использования программы. Установочный пакет включает в себя собственно программу Videostudio и необходимый пакет FFmpeg (распространяется свободно).

Установочный пакет обеспечивает:

- Проверку наличия необходимого места на жестком диске;
- Возможность установки пакета в любое место диска по желанию пользователя;
- Распаковку и установку всех компонентов пакета.

Таким образом, любой человек, получивший в распоряжение дистрибутив программы, сможет самостоятельно установить и полноценно применять ее. Программа в состоянии нормально функционировать не только на компьютере у автора, а в рамках любой подходящей системы. Программа не требует от пользователя каких-либо специальных знаний для установки или работы с ней. Однако рекомендуется изучение возможностей пакета FFmpeg для использования различных режимов трансляции потока.

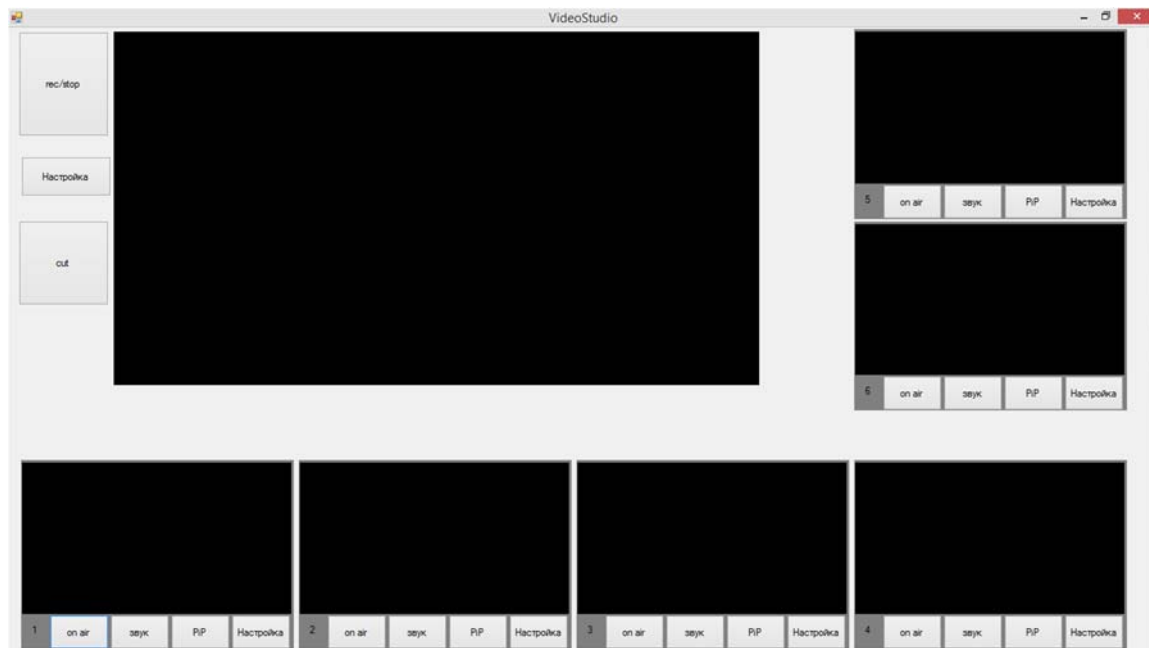
Эргономика

При разработке программы большое внимание уделено эргономике:

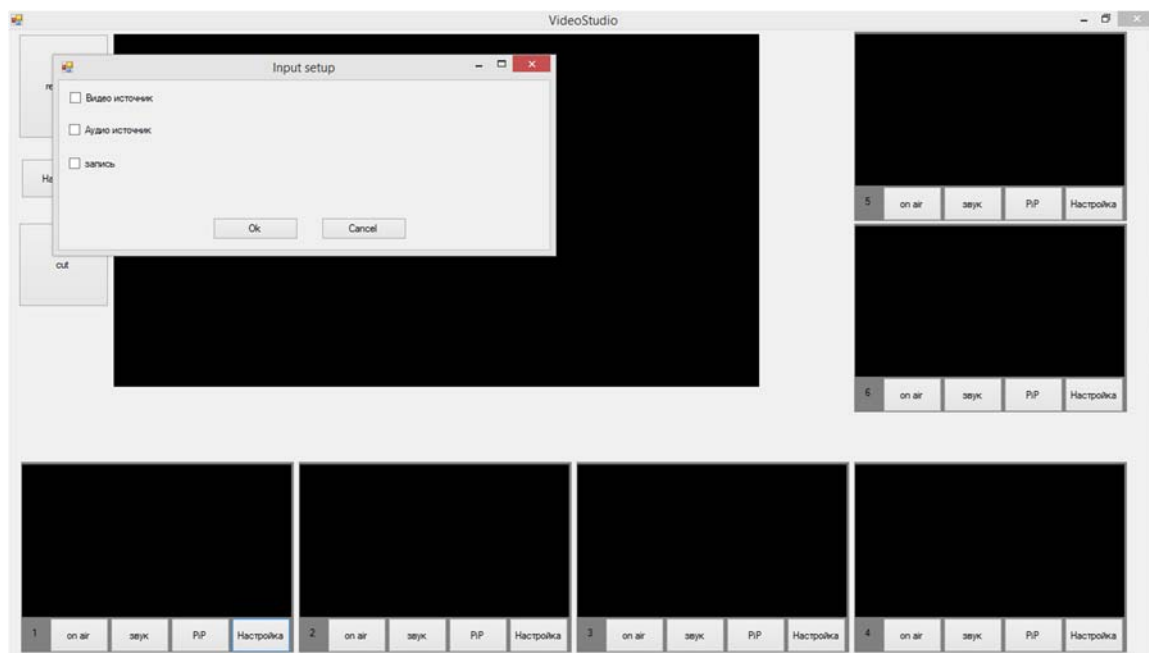
- В зависимости от значения и частоты использования, различные элементы управления выполнены различного размера.
- Кнопки On Air и Pic-in—Pic активного источника выделены цветом, при записи в файл подсвечена также кнопка Start/Stop.
- Предусмотрено переключение между видеовходами как с помощью мыши (путем клика на кнопки соответствующего входа), так и с помощью «горячих клавиш».

Порядок работы программы

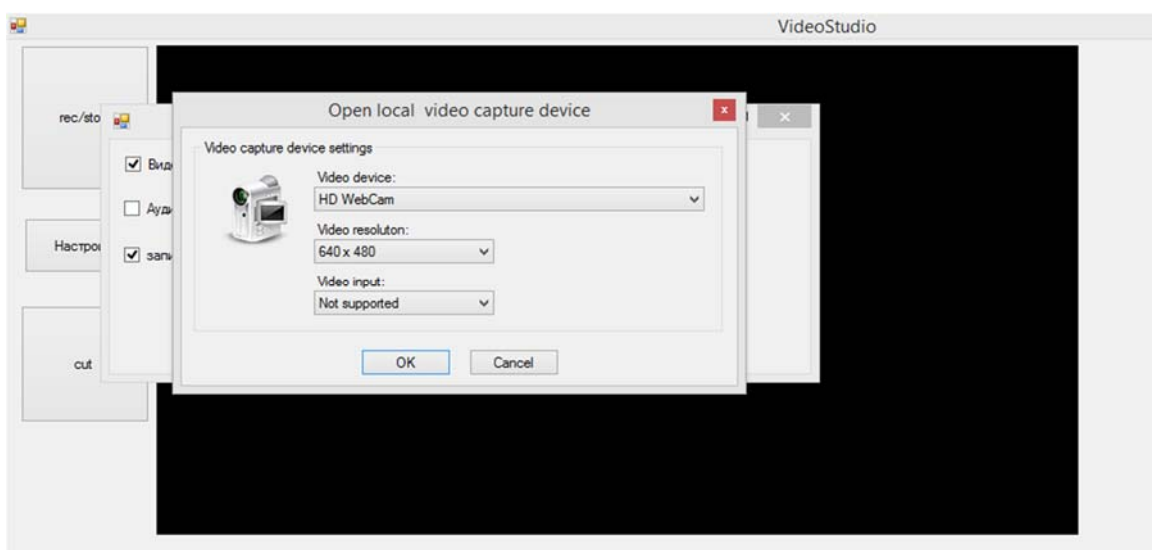
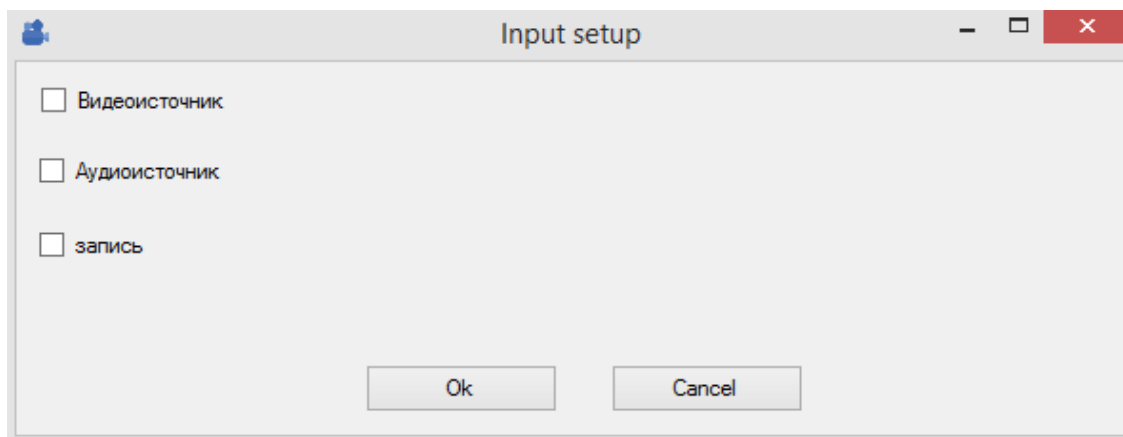
После запуска программы открывается главное окно программы с 6 маленькими одинаковыми блоками, каждый из них настраивается через окно настроек источника. После настройки блока в нем начинает отображаться изображение с источника. Для переключения между источниками можно кликнуть по интересующему источнику, нажать цифру, соответствующую номеру источника на клавиатуре или нажать кнопку On air под превью данного источника.



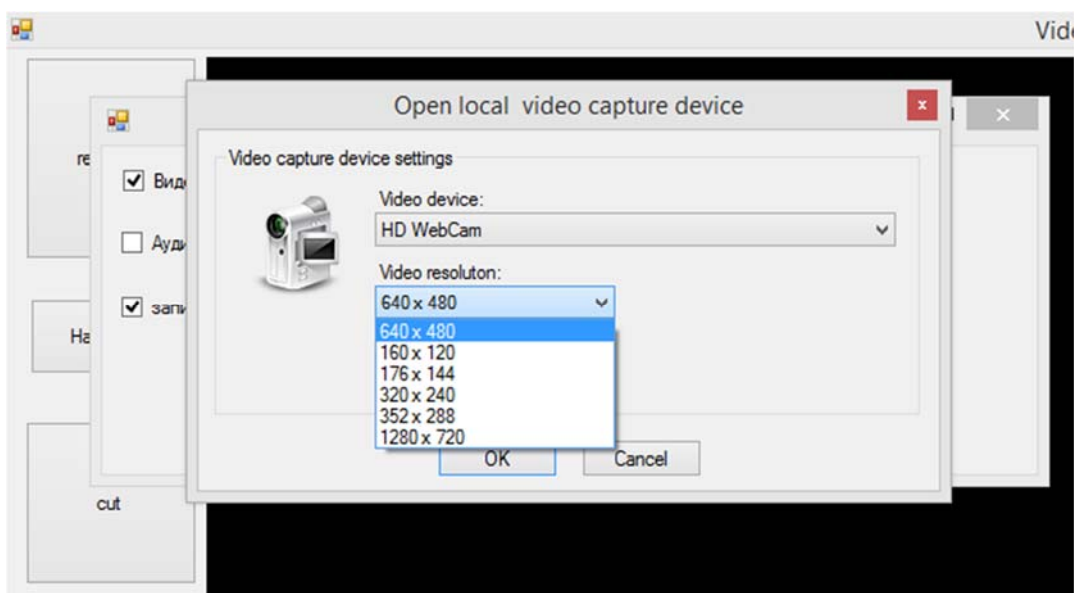
Главное окно программы



Окно настроек источника



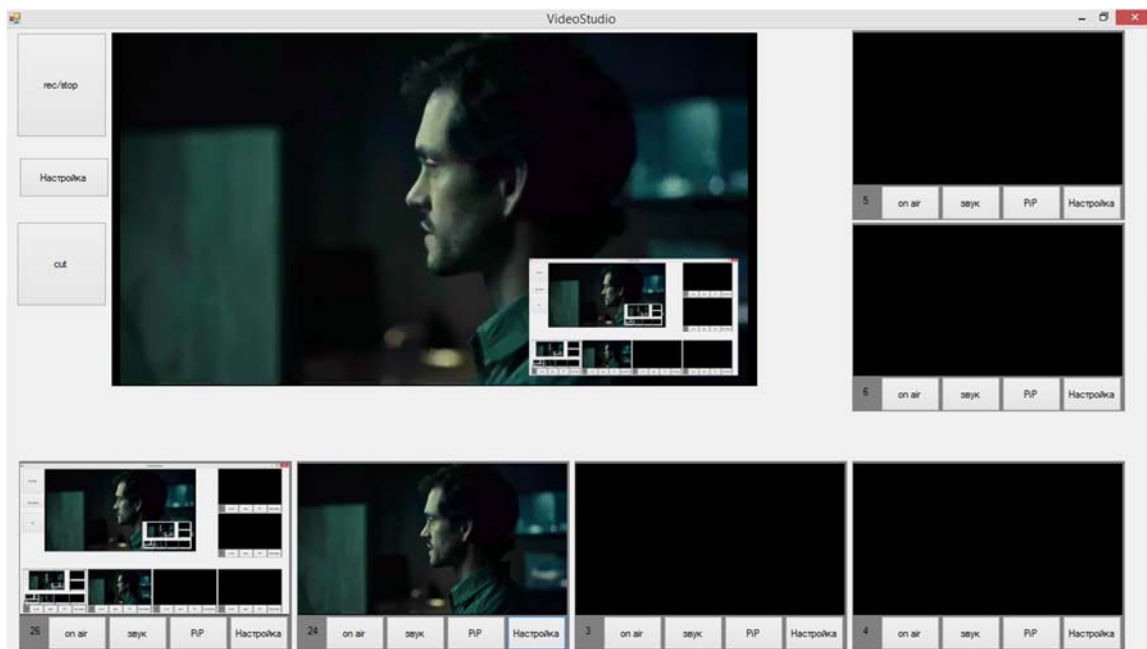
Настройка камеры 1



Настройка камеры 2

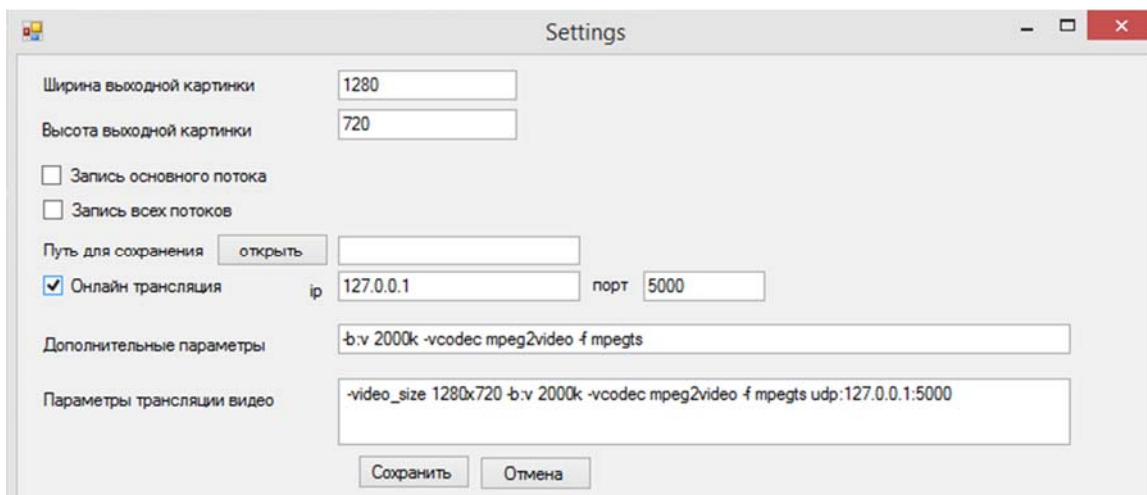
Захват первого монитора ПК

Для создания эффекта «Картинка в картинке» можно нажать клавишу Pic-in-Pic под превью нужного источника, или нажать одну из клавиш клавиатуры (q,w,e,r,t,y соответствуют источникам 1,2,3,4,5,6). Повторный клик или нажатие «горячей клавиши» данного источника отключает функцию «Картинка в картинке».



Демонстрация работы программы (захват с монитора+ проигрывание видеофайла, с применением эффекта картинка в картинке

Окно настроек выходного потока позволяет выбрать основные параметры, а также настроить трансляцию (перед изменением поля “Дополнительные параметры” рекомендуется ознакомиться со справкой для программы FFmpeg). Пользователь может задать любые параметры FFmpeg в специальном поле. При изменении настроек порта необходимо убедиться, что желаемый и следующий за ним порты доступны.



Окно настроек

5. Выводы и возможности дальнейшего развития

В результате работы был разработан и реализован программный продукт - вещательный сервер, обеспечивающий проведение видеотрансляций в сети интернет с уровнем качества изображения, соответствующего стандарту HD на базе бытового оборудования. Данный программный продукт обладает следующими важными свойствами:

- Функционирует под управлением ОС Windows, способен работать на любом современном персональном компьютере или ноутбуке;
- Работает на бытовом оборудовании (подключение камер по интерфейсу IEEE 1394, USB, HDMI, использует Ethernet для передачи HD видео с камер в реальном времени, коммутацию потоков с камер программным микшером, кодирование в форматы, пригодные для передачи на сервер, передачу видео на сервер, раздачу видео с сервера клиентам);
- Позволяет выбирать формат данных для любого из источников видео независимо от других, позволяет отключать, переподключать и менять любой из источников в процессе трансляции;
- Способен работать с шестью источниками видео, включая камеры, захват видео с экрана компьютера, видеопотоки, принимаемые по сети интернет;
- Возможность работы с потоками видео в формате HDTV;
- Обрабатывает и сжимает потоки видео для работы в реальном времени;
- Сформирован в виде полноценного программного продукта, включающего документацию и установочный пакет со всеми необходимыми программными модулями.

Таким образом, пакет может быть использован для организации видеотрансляций, в том числе в видеостудии лица любым пользователем без поддержки автора.

Естественным ограничением применимости программы является большой объем информации в HD-видеопотоке. Поэтому для эффективной работы программы требуется компьютер достаточной мощности.

Созданный пакет решает поставленные задачи и допускает возможности для дальнейшего совершенствования, в т.ч. такие как:

- Возможность встроить использование эффектов при смене источника: описания классов и логика функции смены источника видеопотока предусматривают использование различных алгоритмов эффектов смены кадра (шторка, звезды, затемнение и т.д.);

- Обеспечение многопоточности выходных данных (трансляция нескольких потоков с разным битрейтом).

6. Список литературы

FFmpeg

1. Encoding for Streaming Sites (track.FFmpeg.org)
2. H.264 (track.FFmpeg.org)
3. Streaming guide (track.FFmpeg.org)
4. FFmpeg documentation (FFmpeg.org)
5. Interact with FFmpeg from a .NET program (stackoverflow.com)
6. How to stream raw A/V data to FFmpeg (FFmpeg.gusari.org)
7. ProcessInfo and RedirectStandardOutput (stackoverflow.com)

Aforge.NET

8. Recording using aforges videofilewriter (stackoverflow.com)
9. Aforge video samples (aforgenet.com)
10. Aforge documentation (aforgenet.com)
11. How to record video from laptop camera (aforgenet.com)
12. AForge.Video Namespace (aforgenet.com)
13. How to create video files in C# (from single images) (en.code-bude.net)

Naudio

14. NAudio (naudio.codeplex.com)
15. NAudio multiple Input to multiple Output (stackoverflow.com)
16. NAudio piping WaveIn or WaveProvider to Stream (stackoverflow.com)
17. .NET-диктофон (blogs.msdn.com)

Информация общего характера

18. Захват, обработка и хранение видео с использованием ПК
19. Интернет трансляции (auditory.ru)
20. Цифровая видеостудия (auditory.ru)
21. Comparison of container formats (wikipedia.org)
22. C # UDP Socket client and server (stackoverflow.com)
23. C # Timer.Interval Pattern with 40ms? (stackoverflow.com)
24. UDP: Read data from all network interfaces (stackoverflow.com)