

Faculdade de Engenharia da Universidade do Porto



FEUP

Transmissão Wireless de vídeo DVD/HD

Ricardo Miguel Almeida Salgado

Tese submetida no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major de Telecomunicações

Orientador: Eurico Carrapatoso (Professor Auxiliar)

Julho de 2008

A Dissertação intitulada

“Transmissão Wireless de Vídeo DVD/HD”

foi aprovada em provas realizadas 18/Julho/2008

o júri

Presidente Professor Doutor Mário Jorge Moreira Leitão
Professor Associado da Faculdade de Engenharia da Universidade do Porto



Professora Doutora Susana Isabel Barreto de Miranda Sargento
Professora Auxiliar Convidada da Universidade de Aveiro



Professor Doutor Eurico Elias Morais Carrapatoso
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto



O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

Autor - Ricardo Miguel Almeida Salgado



Resumo

Esta proposta surgiu da necessidade de simplificação da logística necessária para a instalação de um sistema com capacidade de aquisição e transmissão, via *wireless*, de vídeo de alta definição. A necessidade de se capturar e transmitir eventos desportivos em que as capturas são efectuadas em infra-estruturas que não estão preparadas para estes eventos, tem neste sistema uma opção viável e simples. Essa transmissão deverá ser efectuada até uma distância de 60 metros com um sinal de qualidade DVD (*Digital Versatable Disc*) ou do formato HD (*High Definition*).

O desenvolvimento de novas tecnologias de transmissão sem fios permite que este sistema funcione, conseguindo satisfazer todos os requisitos propostos. A solução proposta indica a utilização da tecnologia 802.11 e fornece uma aplicação com várias funcionalidades capazes de demonstrar a viabilidade da solução. A aplicação foi desenvolvida em Direct-Show e permite a visualização ou o armazenamento, no computador, do sinal recebido em diferentes formatos, e com diferentes qualidades de imagem. Permite ainda a recodificação de ficheiros multimédia previamente armazenados.

Os resultados obtidos permitem comprovar a solução proposta acerca da transmissão *wireless* de vídeo de alta definição.

Abstract

This project resulted from the need to simplify the logistics necessary for the installation of a system capable of acquiring and transmitting through wireless, high-definition video. The need to capture and transmit sports events where catches are made in infrastructure that is not prepared for these events and this system is a viable option and simple. This transfer will be made up to a distance of 60 meters with a mark of quality DVD (*Digital Disc Versatable*) or the HD format (*High Definition*).

The development of new technologies for wireless transmission allows this system to work, achieving all the requirements proposed. The proposed solution indicates the technology 802.11 and provides an application with many features able to demonstrate the viability of the solution. The application was developed in DirectShow and allows the viewing or storage in the computer, of signals received in different formats and with different qualities of image. It also allows the recoding of multimedia files stored previously.

The results can prove the proposed solution on the transmission of wireless high-definition video.

Agradecimentos

Queria agradecer ao Prof. Eurico Carrapatoso (Orientador) e ao Eng. António Barny (Proponente) por me elucidarem sobre os objectivos e características da tese, e ainda por me incentivarem e guiarem no desenvolvimento da mesma.

Queria agradecer ao José Pedro Horta que me ajudou na introdução ao DirectShow.

Gostaria de agradecer aos meus colegas que me apoiaram e ajudaram a desbloquear sempre que me era necessário fornecendo perspectiva diferente e complementar para os meus problemas, e em especial ao colega João Meixedo que ainda me ajudou na correcção ortográfica da tese.

Gostaria também agradecer ao meu tio, Ramiro Almeida, pela disponibilização de hardware para testes.

Em último, mas não o menor agradecimento, queria agradecer à minha família que me proporcionou tudo o que foi necessário para a minha educação, a qual encerra um grande capítulo com esta tese.

Ricardo Salgado

*“O que faz um homem de ciência não é a posse de conhecimento,
mas a busca obstinada e audaciosamente crítica da verdade.”*

Karl Popper

Conteúdo

1	Introdução	1
1.1	Contexto e Motivação	1
1.2	Objectivos	2
1.3	Estrutura	2
2	Alternativas Tecnológicas	5
2.1	Aquisição de dados	5
2.2	Transmissão de dados	6
2.2.1	UWB	7
2.2.2	IEEE 802.11	10
2.2.3	Resumo	11
2.3	<i>Codecs</i>	12
2.3.1	<i>codecs</i> sem perdas	13
2.3.2	<i>codecs</i> com perdas	14
2.3.3	Resumo	15
2.4	Recepção de dados	15
2.5	Protocolos de comunicação	16
2.6	Tecnologias de Programação	17
2.6.1	QuickTime	17
2.6.2	GStreamer	17
2.6.3	DirectShow	18
2.6.4	Resumo	20
3	Arquitectura, requisitos e especificação	21
3.1	Arquitectura	21
3.2	Requisitos	22
3.3	Usabilidade	24
3.4	Compatibilidade	24
3.5	Desempenho e fiabilidade	25
4	Implementação	27
4.1	Aplicação de interacção com o utilizador	27
4.1.1	<i>Open File</i>	32
4.1.2	<i>Open Stream</i>	35
4.1.3	Resumo	38
4.2	Transmissão <i>wireless</i>	39
4.2.1	Débito necessário na transmissão	39
4.2.2	Teste às tecnologias 802.11	40

4.2.3	Resumo	41
5	Conclusões e Trabalho Futuro	43
	Referências	46

Lista de Figuras

2.1	Gráfico que demonstra potência em função da frequência	8
2.2	UWB Receiver Transmitter	9
2.3	Comparação de Tecnologias	12
2.4	Evolução das normas dos <i>codecs</i>	13
2.5	Exemplo de um Gráfico	19
2.6	Interface de Desenvolvimento	20
3.1	Arquitectura do sistema	22
3.2	Diagrama de casos de utilização	23
3.3	Diagrama de casos de utilização da escolha Open File	23
3.4	Diagrama de casos de utilização da escolha Open Stream	24
4.1	Gráfico que importa um ficheiro mp3	30
4.2	Gráfico que importa e decodifica um ficheiro mp3	30
4.3	Gráfico de reprodução de um ficheiro MPEG	31
4.4	Aplicação de interacção com o utilizador	31
4.5	Menu de abertura de um ficheiro	32
4.6	Gráfico de visualização de um ficheiro AVI	33
4.7	Gráfico da recodificação de MKV para MPEG	34
4.8	Gráfico da visualização de um ficheiro de vídeo	35
4.9	Menu de abertura de um dispositivo de captura	36
4.10	Gráfico da visualização do dispositivo de captura	36
4.11	Gráfico de captura do dispositivo de captura	38
4.12	Gráfico da utilização da largura de banda	40

Lista de Tabelas

2.1	Diferentes Tecnologias de imagem	6
2.2	Diferentes Tecnologias de Transmissão sem Fios	8
2.3	Tabela Comparativa entre Tecnologias de Transmissão sem Fios	9
2.4	Comparação Débito Distância entre diferentes Tecnologias	10
2.5	Consumos de Componentes de Tecnologias	11
4.1	Débito necessário para transmissão de vídeo	39
4.2	Tabela da diminuição do débito com a distância	40

Capítulo 1

Introdução

1.1 Contexto e Motivação

Este trabalho teve como motivação cobrir a lacuna existente na transmissão de vídeo em eventos ocasionais, nos quais a inexistência de fios será uma mais-valia.

O formato de vídeo HD, do inglês *High Definition*, caracteriza o vídeo de alta definição que será transmitido neste projecto, tendo como resoluções mais comuns as 1280X720 *pixels*(720p) e 1920X1080 *pixels*(1080i ou 1080p).

No mercado já se podem obter filmes e conteúdos multimédia distribuídos em alta definição, quer seja através de HD DVD e de *BlueRay* ou através da transmissão de televisão digital fornecida por operadores de cabo ou satélite. Para se poder tirar partido destas características de vídeo é necessário equipamentos com capacidade *HD Ready* ou *Full Hd*, conseguindo-se obter com a primeira uma resolução de 720p e na segunda a resolução de 1080i.

Este projecto tem como finalidade a transmissão de vídeo em eventos desportivos, concretamente em provas de natação, nas quais se pretende transmitir dados provenientes de duas câmaras, sem recorrer a qualquer tipo de cablagens, devido à complexidade inerente e à falta de infra-estruturas apropriadas nos locais de captura de imagem. Outra possível utilização deste estudo seriam as câmaras utilizadas em programas televisivos com emissão em directo, nos quais se tiraria partido deste projecto, uma vez que seria desnecessária a utilização de uma grande quantidade de cabos e ainda do pessoal necessário para o seu manuseamento.

A tecnologia *wireless* baseia-se em protocolos de transmissão de dados sem fios. Esta tecnologia tem como meio de transmissão a propagação de ondas electromagnéticas no ar. Tal como no caso de uma tecnologia cablada, também esta tem limites de distância, que podem ser atenuados pelo uso de repetidores permitindo uma mobilidade maior mantendo as características de transmissão de dados fornecidas pelo protocolo. Esta tecnologia

permite uma enorme mobilidade de dispositivos como os telemóveis e até à comunicação transatlântica utilizando satélites artificiais. No caso dos telemóveis são usados o GSM (*Global System for Mobile communications*) [1], GPRS (*General Packet Radio Service*) [2] e UMTS (*Universal Mobile Telecommunications System*) [3] para criar as ligações entre os telemóveis e a central para possibilitar as comunicações entre os utilizadores da rede e acessos à Internet. Já os computadores utilizam o protocolo IEEE 802.11 que opera a uma frequência de 2,4GHz para criar redes de dados que permitem partilha de dados e ligações à Internet.

1.2 Objectivos

O objectivo principal desta tese centra-se na proposta de uma solução para a transmissão de dados entre dispositivos de captura de vídeo e um dispositivo de recepção, um computador. A transmissão deverá ser efectuada com base num protocolo de uma tecnologia sem fios. Um dos requisitos principais é transmitir a uma distancia máxima de 60 metros. Pretende-se obter um sinal de vídeo com uma elevada resolução e por conseguinte uma elevada qualidade de imagem, equiparada à qualidade obtida através da reprodução de um DVD (*Digital Video Disc*) ou de uma emissão em formato HD (*High Definition*). Será ainda estudada a melhor forma de transmissão dos dados provenientes da captura de vídeo, verificando-se quais as vantagens, ou desvantagens associadas à codificação do sinal.

Será também criada uma aplicação, no âmbito deste estudo, de interacção com o utilizador com a finalidade de comprovar a validade da solução. Essa interface terá como funcionalidades a visualização da imagem proveniente do dispositivo de captura e a gravação da mesma.

No local de recepção, deve ser possível a visualização do sinal recebido em tempo real, o armazenamento em suporte digital, nomeadamente um disco rígido ou outro qualquer dispositivo físico, em vários formatos com ou sem compressão. Deve ainda ser possível o seu encaminhamento para outras interfaces, como por exemplo a ligação a uma interface de visualização exterior, como um DataShow ou um ecrã de grandes dimensões adequado para eventos desportivos, permitindo aos espectadores uma outra perspectiva do acontecimento.

Neste estudo abordar-se-ão as normas das tecnologias envolvidas e as tendências esperadas para os próximos anos.

1.3 Estrutura

Para além da introdução, esta dissertação encontra-se organizada em mais 4 capítulos, cada um formado por várias secções.

No capítulo 2 são descritas as alternativas tecnológicas relevantes a este projecto, caracterizando a aquisição, transmissão e recepção de dados, sendo referidos em detalhes

os codecs existentes, os protocolos de comunicação e enunciando as tecnologias de programação da aplicação.

No capítulo 3 é demonstrada a arquitectura, requisitos e especificações do estudo. São enumerados os casos de utilização fornecidos ao utilizador da aplicação e caracterizado o desempenho do sistema.

No capítulo 4 encontra-se a implementação da solução ao objectivo da tese. Essa implementação inclui a aplicação criada e dados obtidos em testes.

No capítulo 5 encontram-se as conclusões e trabalho futuro acerca desta matéria.

Capítulo 2

Alternativas Tecnológicas

Neste capítulo serão focadas todas as tecnologias importantes para a concretização deste projecto, mais especificamente, as tecnologias de aquisição, transmissão e tratamento de dados. Além disso será estudado todo o software necessário à interligação das tecnologias anteriores.

2.1 Aquisição de dados

Nesta secção será abordado os tipos de dados que poderão ser adquiridos, indicando algumas características e caracterizando algumas especificações.

A aquisição de dados, neste caso áudio e vídeo, será efectuada por um dispositivo de captura de vídeo e áudio, que será neste caso uma câmara. Este estudo prevê a captura de imagens numa piscina para posterior avaliação dos atletas ou a transmissão em tempo real de um evento.

Existem no mercado vários dispositivos de captura de vídeo que são apropriados a este projecto, dispositivos esses que variam em qualidade e em preço. Tendo em conta as diferentes características de vídeo existentes, o hardware e métodos de transmissão de dados, é necessário ajustar a qualidade de vídeo tendo em consideração esses factores. Existem diferentes resoluções de vídeo e cada uma tem características diferentes e débitos diferentes, como se pode verificar na tabela 2.1.

No mercado existem vários dispositivos de captura que são capazes de capturar a imagem e armazená-la com características de alta definição, e algumas são capazes de transmitir em tempo real a captura por via digital ou analógica. O sistema deverá ser capaz de capturar e transmitir em tempo real imagens com resoluções de 720p e 1080i/p sem qualquer tipo de compressão. Esta funcionalidade necessita de um alto débito binário que poderá não ser suportado pela tecnologia escolhida, podendo-se optar pela utilização de uma saída comprimida tendo em conta os *codecs* existentes e discutidos na secção 2.3.

Sistema (Sem Compressão)	Resolução	Frame-Rate (fps)	Débito (MB/s)
525 NTSC	720 x 486	29.97	27
625 PAL	720 x 576	25	26
720p HDTV	1280 x 720	29.97	140
1080i 1080p HDTV	1920 x 1080	24	127
1080i 1080p HDTV	1920 x 1080	25	132
1080i 1080p HDTV	1920 x 1080	29.97	158

Tabela 2.1: Diferentes Tecnologias de imagem

As diferenças entre as resoluções 1080i e 1080p centram-se na forma de representação da imagem. O *i* significa *Interlaced Scanning* no qual o ecrã representa todas as linhas ímpares numa primeira passagem e as pares em seguida para criar a totalidade da imagem. O *p* significa *Progressive Scanning*, em que a criação da imagem é mais suave e na qual as linhas horizontais são percorridas progressivamente. A resolução 720p segue exactamente os mesmos princípios descritos para o 1080p.

No caso de se optar pela transmissão da saída analógica será necessária a utilização de um conversor A/D (*Analógico/Digital*), que converterá o sinal analógico emitido num sinal digital em binário, com uma resolução que depende do número de níveis de quantificação. Para se conseguir transmitir um sinal de grande débito binário será necessário optar-se por um conversor A/D com um número elevado de níveis, para se conseguir transmitir o vídeo de alta definição. O número de níveis é dado pelo número de bits de resolução do conversor. Esse número (Q), de níveis de quantificação, é obtido pela seguinte fórmula:

$$Q = \frac{\delta}{2^M} \quad (2.1)$$

Na qual o δ é diferença entre os picos do sinal e o M é número de bits, e além disso, segundo o critério de Nyquist, o conversor deverá ter uma frequência de amostragem duas vezes superior à frequência máxima do sinal analógico.

Ao optar-se pela transmissão do sinal digital, as imagens poderão ser transmitidas comprimidas ou sem compressão tendo em conta a resolução escolhida e a tecnologia de transferência de dados existente. As imagens serão transmitidas pelo mesmo terminal físico que transmitirá o sinal proveniente do conversor A/D ao se optar pela utilização da saída analógica.

2.2 Transmissão de dados

Nesta secção serão abordadas algumas tecnologias existentes de transmissão de dados sem fios, tendo em consideração os parâmetros exigidos pelo projecto, que restringiram o uso a tecnologias de alto débito binário como o UWB (*Ultra-Wideband*), as tecnologias

baseadas no protocolo IEEE 802.11 (*Institute of Electrical and Electronics Engineers*) e descartando à partida algumas tecnologias definidas internacionalmente como por exemplo o Bluetooth, GSM, GPRS ou UMTS.

2.2.1 UWB

A tecnologia UWB é uma tecnologia *wireless* de transmissão de dados por rádio frequência. Esta tecnologia baseia-se nas normas *WiMedia* e caracteriza-se como sendo uma tecnologia de alto débito, baixo consumo e baixo alcance tornando-a indicada para formar redes domésticas, interligando dispositivos de áudio, vídeo e outros de grande débito.

Os sistemas UWB são diferentes dos sistemas convencionais porque não se baseiam num sistema de portadoras e são constituídos por um ou mais impulsos de ondas rádio, capazes de transmitir e receber impulsos muito curtos, de duração inferior a 1 nano segundo.

Devido à curtíssima duração dos impulsos e à potência de transmissão se encontrar espalhada ao longo de um grande espectro de frequências, este sistema é um sistema de baixa densidade espectral de potência.

A densidade espectral de potência (*PSD - Power Spectrum Density*) define-se pela seguinte fórmula matemática:

$$PSD = \frac{P}{B} \quad (2.2)$$

na qual P é a potência transmitida em Watts e B a largura de banda utilizada (Hz).

Normalmente os sistemas usuais de transferência de dados utilizam uma banda de transmissão estreita enquanto o sistema UWB usa uma banda de transmissão muito mais alargada, e como a potência média de transmissão é distribuída por essa banda, permite que o UWB obtenha uma baixa densidade espectral de potência, tornando esta tecnologia muito apetecível porque minimiza as interferências entre esta tecnologia e os sistemas de banda estreita. Na figura 2.1 [4] pode-se verificar esta afirmação.



Figura 2.1: Gráfico que demonstra potência em função da frequência

Na tabela 2.2 encontra-se a comparação entre diferentes tecnologias de transmissão sem fios indicando a potência de transmissão, a banda utilizada e o PSD classificando cada uma das tecnologias, com base no seguinte artigo [4]:

Sistema	Potência de Transmissão	Largura de Banda	PSD ($\frac{W}{MHz}$)	Classificação
Rádio	50 Kw	75 KHz	666600	Banda Estreita
Televisão	100 Kw	6 MHz	16700	Banda Estreita
Tecnologia 2G	10 mW	8.33 KHz	1.2	Banda Estreita
802.11a	1 W	20 MHz	0.05	Banda Larga
UWB	1 mW	7.5 GHz	0.013	Banda Ultra Larga

Tabela 2.2: Diferentes Tecnologias de Transmissão sem Fios

A tecnologia UWB está neste momento em desenvolvimento, não existindo uma norma fixa, estando neste momento várias empresas a desenvolver esta tecnologia, cada uma obtendo diferentes funcionalidades e características. Estes desenvolvimentos estão restringidos com alguns parâmetros fixos impostos pelo FCC (*Federal Communications Commission*) nos Estados Unidos e pela ITU-R (*International Communication Union- Radiocommunication Center*), como por exemplo a gama de frequências em que esta tecnologia pode operar, que terá que ser dos 3.1 GHz até aos 10.6 GHz em recintos fechados, e a potência limitada a -41 dBm/MHz .

O sinal UWB foi inicialmente modulado recorrendo a técnicas de modulação tradicionais, como a modulação por posição de pulso PPM e a modulação por amplitude de pulso PAM, tendo diferentes resultados em termos de débito e densidade espectral de potência como se pode ver em [5]. Mais tarde surgiram novas técnicas de modulação como modulação por formato de pulso PSM binário ou quaternário que obtém, dependendo das circunstâncias, melhor desempenho que o PAM e o PPM [6] e ainda a modulação por divisão de bandas PBM.

A transmissão entre o emissor e o receptor pode ser executada em sincronização perfeita, na presença de ruído branco gaussiano, sob a interferência de multi-percursos e em sincronização imperfeita. Todos estes modos de transmissão obtêm resultados diferentes, alguns mais favoráveis a determinadas situações e outros a situações complementares. Na figura 2.2 apresenta-se um esquema possível para um sistema UWB básico.

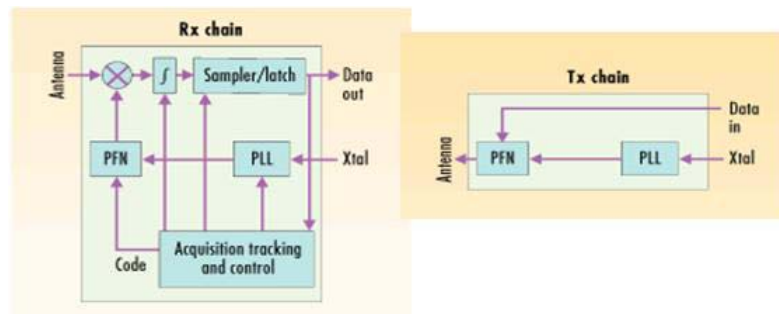


Figura 2.2: UWB Receiver Transmitter

Esta tecnologia tem uma excelente capacidade espacial, que pode ser medida através da divisão da taxa máxima de dados (bit/s) pela área (m^2) em que o sistema pode transmitir. A capacidade espectral é similar a outras tecnologias existentes e é calculada através da divisão da taxa máxima de dados pela largura de banda (Hz). Na tabela 2.3 encontra-se uma comparação entre as diferentes tecnologias, em relação a estes dois parâmetros.

Sistema	Débito de Transmissão (Mb/s)	Cap. Espacial ($Kbps/m^2$)	Cap. Espectral ($Kbps/Hz$)
UWB	100	318.3	0.013
IEEE 802.11a	54	6.9	2.7
Bluetooth	1	3.2	0.012
IEEE 802.11b	11	0.350	0.1317

Tabela 2.3: Tabela Comparativa entre Tecnologias de Transmissão sem Fios

Em termos de débito esta tecnologia pode-se caracterizar como a tecnologia similar a USB, mas *wireless*, porque consegue obter débitos na ordem dos 480 Mbps em curtas distâncias. Com o aumento da distância esta tecnologia perde capacidade de transferência, e essa perda é muito significativa, pois a 10 metros de distância a capacidade de transferência já desceu para cerca de 100 Mbps. Podemos ver na tabela 2.4 uma comparação entre tecnologias.

O consumo de energia desta tecnologia pode-se considerar como ultra *Low-Power*, como foi dito anteriormente e pode ser confirmado pela comunidade internacional, tendo como objectivo final um consumo na ordem dos 0,1W. Em 2.5 encontra-se uma tabela comparativa dos consumos de diversos componentes de diversas tecnologias.

Esta tecnologia é, apesar de tudo, uma tecnologia pouco complexa e com um baixo custo associado. Os sistemas UWB podem ser construídos numa base quase totalmente

Débito Binário (Mb/s)	Normas
480	UWB, USB 2.0
200	UWB (4 metros)
110	UWB (10 metros)
90	Ethernet 100
54	802.11a
20	802.11g
11	802.11b
10	Ethernet 10
1	Bluetooth

Tabela 2.4: Comparação Débito Distância entre diferentes Tecnologias

digital e com poucos componentes de rádio frequência. Com a intenção de aumentar o débito de transferência de dados, estes sistemas têm incrementado a complexidade mas com a produção em massa o preço deste sistema será muito reduzido.

2.2.2 IEEE 802.11

Wi-fi é uma tecnologia de transmissão de dados sem fios baseada em protocolos IEEE 802.11 que tem várias variantes. O 802.11b e g são os mais usuais e os que neste momento estão a ser usados em maior quantidade em todo o planeta. O *Wi-Fi* é uma tecnologia que neste momento está a ser utilizada nas redes locais para garantir mobilidade, sendo utilizada por computadores e também telemóveis/PDA's, caracterizando-se por ser uma tecnologia de débitos binários razoáveis e consumo relativamente reduzido.

O *Wi-Fi* é um sistema que se baseia em portadoras e tem como frequência de transmissão os 2,4 GHz, com um débito binário de 54 Mbit/s no caso do 802.11g o protocolo mais utilizado. Existem certos fabricantes que obtêm um débito de 108 Mbit/s neste protocolo com a introdução de novas tecnologias nos seus equipamentos. O último protocolo aprovado foi o 802.11n que tem uma técnica de modulação diferente dos outros protocolos usando múltiplas antenas para transmitir e receber dados de forma a melhorar os débitos, atingindo um débito binário máximo de 248 Mbit/s prevendo-se que esse valor possa ainda ser superado substancialmente. Apesar de ainda não ter sido aprovada uma versão final, o desenvolvimento da tecnologia tem sido considerável, tendo sido aprovado o *Draft 4.0* [7, 8] em Maio do corrente ano.

Em relação a aspectos de consumo, esta tecnologia terá aproximadamente o mesmo consumo que a referenciada acima para o protocolo 802.11a e os sistemas que usem esta tecnologia terão um custo energético muito superior ao necessário para um sistema que use a tecnologia UWB.

O alcance (aproximadamente 70 metros) e principalmente o débito foram as principais melhorias neste protocolo em relação aos protocolos anteriores.

Aplicação/ Tecnologia	Consumo(mW)
802.11a	1500-2000
Telemóvel	200
Conversor A/D de 12-bit Câmara digital	150
UWB	100
Display TFT a cores de Telemóvel	75

Tabela 2.5: Consumos de Componentes de Tecnologias

Neste preciso momento existem empresas em fase de melhoria das especificações deste protocolo que ainda se encontra em fase de melhoramentos, apesar de já ter sido aprovado. Essas melhorias são principalmente influenciadas pela conjugação desta tecnologia com a tecnologia MIMO (*Multiple Input Multiple Output*). A tecnologia MIMO baseia-se no uso de antenas múltiplas tanto no emissor como no receptor. Com múltiplas antenas explora-se a dimensão espacial para incrementar as performances da ligação. Este sistema baseia-se na divisão de um sinal de alto débito em vários de menor débito, que são transmitidos e na sua recepção agrupado. Existem ainda várias características desta tecnologia que influenciam a transmissão, mas o uso desta tecnologia conjugada com o protocolo IEEE802.11n levou ao aumento do débito binário e ainda da distância. Com a utilização de uma variante da tecnologia MIMO, denominada de 4X4 MIMO, este aumento ainda se torna mais notável, podendo elevar o débito de pico até 600 Mbps.

Neste protocolo existem ainda factores que limitam o débito binário útil, para transmissão de dados, e que estão em estudo com o objectivo de melhorar o rácio débito fornecido, débito útil. Neste artigo [9] o autor introduz alguns melhoramentos que podem ser executados para diminuir o *overhead* através de novas estratégias de MAC que, como ele indica, melhoram muito o débito útil.

Neste momento podemos dizer que esta tecnologia, apesar de não ser completamente oficializada, já se encontra no mercado podendo-se afirmar:

”At 300 feet, 802.11g performance plummets to 1 Mbps. 802.11n networks operate at up to 70 Mbps-70 times faster than 802.11g.” (fonte Broadcom Inc. [10])

2.2.3 Resumo

As tecnologias de transmissão de dados referidas podem ser utilizadas neste projecto, tendo ainda que se ter como factores de decisão algumas restrições específicas de cada tecnologia e as características necessárias para a execução do projecto. Em termos de débitos binários a comparação entre estas tecnologias encontra-se ilustrada na figura 2.3, na qual se demonstra as diferenças entre os valores de débito de pico ao longo dos anos da UWB e do IEEE802.11n, indicando ainda o valor existente para o Wireless USB.

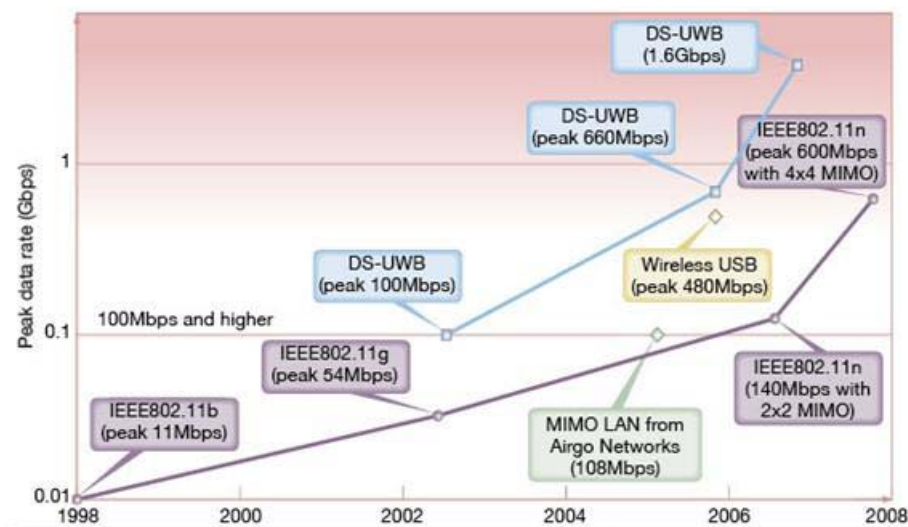


Figura 2.3: Comparação de Tecnologias

2.3 Codecs

Neste projecto será necessário estudar as codificações de vídeo e áudio existentes que melhor se aplicariam no sistema. Essencialmente existem dois tipos de *codecs* (*Coding Decoding*), os *codecs* sem perdas e os *codecs* com perdas. Os *codecs* com perdas exploram certas características do sistema visual humano para reduzir o débito binário e o espaço de armazenamento necessário para os dados. Dentro destes existem um número incontável de tipos de *codecs*, cada um com diferentes características que se adaptam melhor a diferentes tipos de circunstâncias. Segundo Mohammed Ghanbari, autor do livro [11], os princípios de compressão podem ser resumidos a 3 princípios de redução de redundância:

1. Redução de redundância espacial - que se baseia na redução da redundância ao longo dos *pixels* dentro da imagem empregando alguns compressores de dados.
2. Redução de redundância temporal - reduz-se tendo em conta as semelhanças entre imagens sucessivas codificando apenas as diferenças entre elas.
3. Métodos estatísticos - métodos baseados na codificação de *Huffman* e na codificação aritmética, usando técnicas de codificação de tamanho variável para, tendo em conta as probabilidades de ocorrências de símbolos dentro dos dados comprimidos, diminuir a taxa de bits.

Existem diferenças entre estes dois tipos de *codecs*, mas a principal característica baseia-se naturalmente no débito necessário para cada tipo, em que os *codecs* sem perdas compactam 2 a 3 vezes, enquanto os *codecs* com perdas podem compactar dezenas de vezes o tamanho original. Essa compressão implicará uma maior complexidade na codificação e decodificação dos dados, e segundo Borivoje Westwater [12], esse aumento apesar de não ser directamente proporcional levará a um aumento da quantidade de processamento

necessária e a um conseqüente aumento do atraso entre a captura e a reprodução em tempo real do sinal de vídeo.

Se os dados não fossem codificados teríamos requisitos de débito transmitido e espaço de armazenamento proibitivos, tendo como exemplo um vídeo (320 x 240 *pixels*) que para ser visualizado num televisor RGB a 30 fps (*Frames per Second*) necessitaria de aproximadamente 7 milhões de bytes por segundo. Durante a dissertação irá ser determinado qual destes tipos se enquadrará melhor nas especificações requeridas efectuando alguns testes. Na figura 2.4 pode-se observar a evolução das normas de codificação de vídeo dos comités ITU-T (*ITU- Telecommunication Standardization Sector*) e ISO/IEC (*International Organization for Standardization / International Electrotechnical Commission*).

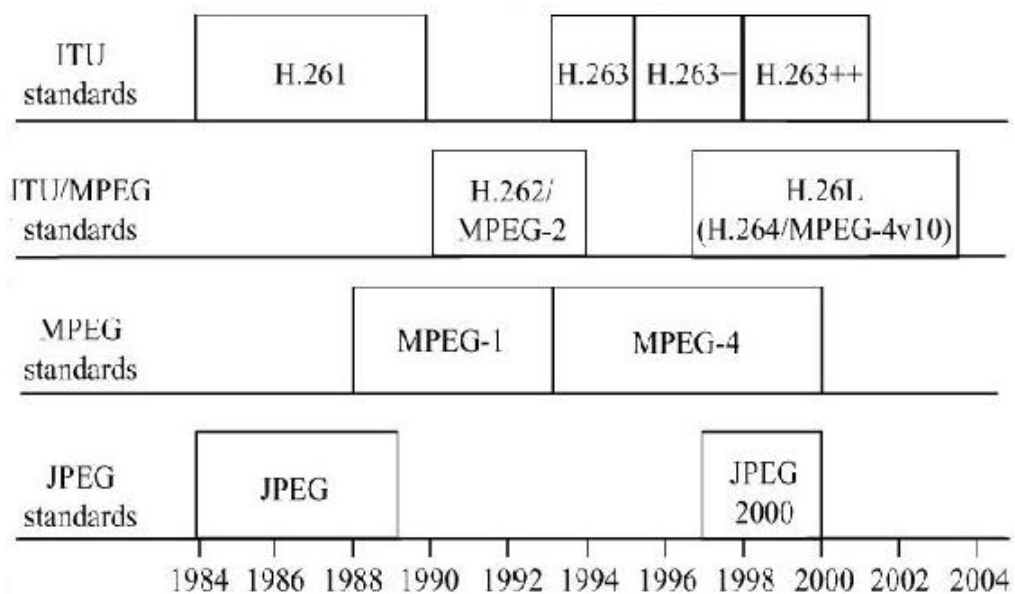


Figura 2.4: Evolução das normas dos *codecs*

Isto não implica que não se possa utilizar normas mais antigas, apenas indica evoluções em alguns e inovações em outras áreas.

2.3.1 *codecs* sem perdas

A compressão sem perdas utiliza algoritmos de compressão que permitem obter os dados originais sem perda de informação. Este tipo de compressões não é apenas usada em vídeo e som, mas em qualquer tipo de dados.

Os *codecs* de vídeo existem em menor número que os *codecs* de áudio devido à grande quantidade de dados envolvida e tende-se a optar por *codecs* com perdas para facilitar a mobilidade dos dados.

De entre os *codecs* de vídeo existentes destacam-se os seguintes: o MSU Lossless Video Codec que é o *codec* que obtém a melhor taxa de compressão dentro deste tipo de *codecs*, o Huffuyf que é um *codec* com uma codificação rápida, podendo-se equiparar o tempo de

codificação necessária para a codificação com os tempos necessários para os *codecs* com perdas. O Lagarith, que não é tão rápido como o anterior, tem um suporte mais vasto de gamas de cores e utiliza um melhor método de compressão. Existem ainda o FFV1, o CorePNG e ainda outros.

Os *codecs* de áudio são inúmeros devido à grande importância dada pelos utilizadores à possibilidade de preservar toda a sua biblioteca musical em formatos que, em caso de acidentes, permita reaver a sua biblioteca com a qualidade inicial, e ainda à grande quantidade de áudio difundida na Internet, quer por transmissões *online* de rádios ou apenas como forma de interacção entre o utilizador e o conteúdo que está a visualizar.

O grande número de *codecs* não será totalmente ilustrada, mas apenas algumas referências serão feitas. O FLAC (*Free Lossless Audio Codec*) destaca-se como um *codec* muito compatível e robusto e tendo ainda características muito similares a todos os outros, o Apple Lossless desenvolvido pela Apple, o WMA Lossless (*Windows Media Audio 9 Lossless*) desenvolvido pela Microsoft, o OptimFROG que consegue a maior taxa de compressão de todos os *codecs*, existindo ainda o Monkey's Audio, o LPAC, o ALAC e ainda muitos outros.

2.3.2 *codecs* com perdas

A compressão com perdas utiliza algoritmos de compressão que conseguem uma taxa de compressão muito alta à custa da deterioração da qualidade. Este tipo de compressão é geralmente utilizada no áudio, vídeo e imagens nas quais a perda de qualidade pode não ser notada pelo ser humano porque o cérebro tende a corrigir as falhas em relação ao original, tendo como base a experiência passada.

Existem imensos *codecs* de vídeo disponíveis para utilização, muitos com características semelhantes e poucos pontos individualizadores que possam levar a um uso preferencial, sendo assim muito difícil identificar qual o melhor para este projecto. Serve de exemplo o MJPEG (*Motion Joint Photographic Experts Group*) no qual o vídeo é uma sequência de imagens JPEG, o Pixlet criado pela Apple com o intuito de diminuir o débito necessário para se poder visualizar HD (*High Definition*), o WMV (*Windows Media Video*) criado para competir com o Real Video baseado no *codec* H.263 antecessor do H.264. O H.264 foi criado tendo em consideração três aplicações, *IP-based streaming*, o download de *Video Streams* previamente codificados e aplicações de vídeo-conferência. No artigo [13] S. Wenger conclui que estas novas funcionalidades combinadas com o novo *draft* das especificações do protocolo RTP (*Rela-Time Transport Protocol*) melhoram consideravelmente o desempenho da transferência de vídeo de alta qualidade por IP (*Internet Protocol*). Este protocolo é uma norma da compressão de vídeo que definiu as bases de todas as evoluções de *codecs* deste tipo de compressão e que ainda está em desenvolvimento. O H.264 foi a base de evolução do MPEG-4 ASP (*Moving Picture Experts Group - Advanced Simple Profile*) e do MPEG-4 AVC (*Moving Picture Experts Group - Advanced Video Coding*). Baseando-se nestas evoluções surgiram muitos *codecs* diferentes, dos quais se podem

destacar, no caso do MPEG-4 ASP, o Xvid um software livre que compete com o comercial DivX (*Digital Video Express*). No caso do MPEG-4 AVC o Core AVC foi desenvolvido por uma empresa sendo hoje em dia um dos mais rápidos codificadores existentes.

Não existem muitos *codecs* de áudio e o aparecimento do MP3 (*MPEG-1 Audio Layer 3*), que é o *codec* mais utilizado em todo mundo, contribuiu para isso. Mesmo assim, ainda se usam outros *codecs* como o WMA (*Windows Media Audio*), o Vorbis que é um *codec* livre e que geralmente está associado ao pacote Ogg, sendo estes três vocacionados para músicas, existindo também ainda outros *codecs* diferentes que são mais vocacionados para o discurso.

2.3.3 Resumo

Os *codecs* são responsáveis pela diminuição do débito binário necessário para a transmissão de dados entre o dispositivo de captura e o receptor, e os *codecs* com perdas obtêm uma compactação muito superior à compactação obtida pelos *codecs* sem perdas, tendo como factor negativo a pior qualidade final nos dados.

Tendo em consideração que o objectivo é transmitir vídeo através de uma rede sem fios, e concordando com os autores [14] que afirmam que a transmissão de dados tem que ser feita tendo em consideração a largura de banda existente e, se tal for necessário, terá que se diminuir o débito através da codificação e considerando que se pretende obter vídeo com qualidade de alta definição, terá que se ponderar as combinações possíveis entre tecnologias de transmissão de dados e a utilização ou não de codificações existentes.

No caso de se utilizar codificações, segundo [15], existe ainda o problema inerente à perda de dados da transmissão que poderá levar ao bloqueio da descodificação dependendo do tipo de codificação escolhida. Portanto na escolha da codificação a utilizar terá que se ter em conta este parâmetro escolhendo *codecs* que apliquem técnicas de sincronização ou interpolação espacial ou temporal.

2.4 Recepção de dados

O terminal de recepção de dados será influenciado pela escolha da tecnologia de transmissão. Considerando que a recepção e o envio de dados utilizam a mesma tecnologia, o terminal terá que ter capacidade de processamento para efectuar o tratamento de dados, similar à capacidade de processamento necessária quando se efectua uma ligação directa por USB ou FireWire no terminal onde se irá efectuar o armazenamento, tratamento/edição e visualização dos dados recebidos. Geralmente o dispositivo de captura fornece software para edição dos dados, mas existem *softwares* profissionais disponíveis se tal for necessário.

2.5 Protocolos de comunicação

Após se escolher a tecnologia de transmissão mais adequada ao projecto e o *codec* que se irá utilizar na codificação dos dados, terá que se escolher o protocolo de transmissão de dados sobre IP. Os protocolos existentes são o TCP (*Transmission Control Protocol*) e o UDP (*User Datagram Protocol*).

O TCP é um protocolo do nível da camada de transporte do modelo OSI e é o mais utilizado pela maioria das aplicações existentes tendo como características a confiabilidade, a orientação à conexão, a execução de ligações ponto a ponto, a entrega ordenada e o controlo de fluxo. O desempenho dos mecanismos de controlo de fluxo e de congestionamento do TCP são influenciados do o valor $Bandwidth * Delay$, valor esse influenciado pelo RTT (*Round Trip Time*) e pela largura de banda da ligação determinando o máximo volume de dados que se podem transmitir. Estes mecanismos atrasam a entrega dos dados e ainda diminuem a quantidade máxima de dados que se pode transmitir. Resumindo é um protocolo que garante que os dados enviados chegam ao destino sem perdas, no qual é estabelecida uma ligação entre o emissor e o receptor e transmitidos os dados. Caso exista alguma perda ou erro o protocolo tem informação dos dados perdidos e retransmite-os e no final termina a ligação entre o emissor e o receptor.

O protocolo UDP não garante a confiabilidade do protocolo TCP, os dados podem chegar desordenados, duplicados ou até nem chegarem. Mas a característica que nos interessa neste protocolo é a sua rapidez. Ao invés do TCP, que para garantir a entrega dos pacotes utiliza várias mensagens entre o emissor e o receptor que tornam a transmissão lenta, o UDP transmite os dados com rapidez, sendo mais apropriado o uso deste protocolo para transmissões de dados referentes a transmissões de vídeo em tempo real, onde se deseja ter a menor diferença temporal possível entre a altura da captura do vídeo e a sua exibição no receptor.

Normalmente é usado o protocolo UDP para *video streaming* e, como o UDP não garante a entrega dos dados, o receptor tem que confiar em protocolos de camadas superiores à do protocolo UDP. O protocolo RTP e o RTCP (*Real-Time Control Protocol*) são exemplos de protocolos num nível superior ao UDP, que são desenhados para ligação ponto-a-ponto e entrega em tempo real de dados de vídeo e voz. O RTP permite a detecção de perda de dados, controla entregas a vários destinatários, identifica o conteúdo e garante a sua segurança, mas não garante qualidade de serviço nem entrega confiável. O RTCP é desenhado para trabalhar em conjunto com o RTP para fornecer *feedback* à aplicação acerca da qualidade da entrega dos dados. Os autores [16] concluíram que normalmente é utilizado o protocolo UDP que conjugado com o RTP fornece uma base óptima para a transferência em tempo real de multimédia.

Devido a uma maior utilização do protocolo UDP para a transmissão de vídeo, existem vários estudos que têm como finalidade aumentar a QoS (*Quality of Service*) da transmissão. Os autores [17] sugerem a criação de uma *framework* com funcionalidades que

aumentam a QoS de uma transmissão de vídeo, como o *Scalable video* que permite que as aplicações tenham conhecimento da rede onde se encontram e das características da mesma, modificando as características do vídeo transmitido para aumentar a qualidade de serviço. O *Scalable video* é a capacidade do sistema decodificar apenas parte dos dados, obtendo uma imagem de qualidade inferior, imagem menor ou *frame-rate* inferior quando a rede não permite que se transfira a totalidade dos dados para se obter a imagem original.

2.6 Tecnologias de Programação

Neste secção iremos abordar as tecnologias de programação existentes para fazer o programa de recepção e captura dos dados, tendo sempre em foco as características comuns e únicas das diferentes tecnologias.

2.6.1 QuickTime

O QuickTime fornece uma API (*application programming interface*) muito funcional que permite visualizar, importar, exportar e capturar muitos tipos de média. Com esta aplicação pode-se criar programas, aplicações que podem ser executados tanto em Mac OS (*Operating System*) como em Windows, desde que se tenha instalado o QuickTime.

O QuickTime baseia-se num conjunto de abstrações, que permite ao criador, no desenvolvimento da sua aplicação, utilizar apenas pequenas partes do todo, permitindo um desenvolvimento mais simples na criação da aplicação para processar os ficheiros multimédia. Fornece ainda várias funcionalidades mais específicas como funções ao nível do pixel, controlo do GPU (*Graphics Processing Unit*), controlo do *buffer* de vídeo e retirar áudio a um ficheiro multimédia para misturar com outro ficheiro de áudio e voltar a incorporar o áudio no ficheiro multimédia.

A comunidade existente que utiliza este recurso mantém um conjunto de fóruns no qual partilham os conhecimentos adquiridos ao longo do desenvolvimento de aplicações e ainda recebe contribuições dos elementos responsáveis pela versão comercial do QuickTime. Além das funcionalidades genéricas, o QuickTime tem várias funcionalidades e funções dedicadas aos produtos da Apple, permitindo ao criador da aplicação usar ambas. Resumindo, permite ao utilizador controlar os ficheiros multimédia com funcionalidades extra com o hardware Apple tanto em Mac OS como em Windows.

2.6.2 GStreamer

O GStreamer, ao contrário das outras duas tecnologias aqui referenciadas, tem a capacidade de ser desenvolvido e executado em diferentes S.O. (*Sistemas Operativos*), podendo-se dizer que a sua vertente multifacetada quase que possibilita que seja usado em muitos sistemas operativos, incluindo o Linux/Unix, MacOSX, Solaris, IBM OS400, SPARC e Microsoft Windows. O GStreamer tem ainda a capacidade de utilizar *codecs* de outras

plataformas e a habilidade de se interligar com outras tecnologias de programação, como o QuickTime e o DirectShow.

A programação tem uma estrutura baseada em gráficos, que permitem uma ligação arbitrária entre as estruturas, de uma forma trivial e transparente, mantendo baixo o processamento necessário conseguindo um desempenho alto com uma baixa latência. Baseada na biblioteca do Gnome (GLib 2.0) com um sistema de depuração disponível para criadores de aplicações, fornecem programas que asseguram qualidade de serviço quando o CPU se encontra com grande carga, e que as aplicações tenham um relógio para manter sincronizado o áudio e o vídeo.

O GStreamer permite a utilização de vários elementos multimédia e tem a capacidade de fazer ligações complexas automaticamente utilizando todo o tipo de fontes, filtros e elementos de captura ou renderização. É ainda fornecida documentação, manuais, aplicações e outras ferramentas para ajudar o criador de aplicações, e a comunidade que trabalha com o GStreamer mantém o lema do Linux:

”Viva livre ou morra,”

ou seja, tudo é livre e partilhado pela comunidade.

Resumindo o GStreamer é uma biblioteca de processamento multimédia, baseado em *pipelines* nos quais os dados fluem, provenientes de elementos, que depois podem ser tratados, modificados por outros elementos e finalmente visualizados ou arquivados.

2.6.3 DirectShow

O Microsoft DirectShow é uma aplicação de criação de software de multimédia, no passado também conhecida como ActiveMovie e fornecida conjuntamente com o Microsoft Direct X, e agora fornecida com o Microsoft Platform SDK para ser executado no SO Windows. O objectivo principal do DirectShow é permitir o desenvolvimento de aplicações dinâmicas e poderosas sem se preocupar com detalhes internos de processos de *streaming*, com detalhes dos ficheiros multimédia e com implementação do hardware permitindo ao criador abstrair-se desses detalhes e concentrar-se na funcionalidade da sua aplicação e na sua robustez.

O DirectShow introduziu uma arquitectura modular baseada em objectos COM (*Component Object Model*) e que com o auxílio das tecnologias DirectDraw e DirectSound isola as aplicações dos dispositivos de hardware com que as aplicações funcionam. O DirectShow fornece suporte a vários tipos de hardware tanto analógicos como digitais, fornecendo ainda varios *codecs* compressores, descompressores, filtros de efeitos, etc.

Os filtros são componentes básicos utilizados na construção das aplicações, sendo cada filtro responsável por uma função específica e crítica na aplicação, sendo geralmente categorizados em três tipos: Source Filters que abrem uma fonte multimédia, seja ela um ficheiro único ou algum dispositivo, e a torna disponível para o resto da aplicação; Transform Filters que modificam os dados provenientes do filtro anterior, quer este tenha sido

um Source Filter ou outro Transform Filter, para codificar os dados, decodificar ou dividir o áudio e o vídeo de um ficheiro multimédia; e por último temos os Renderer Filters que podem gravar os dados resultantes das transformações num dispositivo físico, com por exemplo um disco rígido, ou pode entregar os dados a um dispositivo de renderização, para se poder obter uma visualização e/ou audição dos dados.

Os filtros são conectados de uma forma específica para se obter um resultado previamente definido, como por exemplo reproduzir um ficheiro de áudio e com o conjunto destas ligações obtemos um gráfico. Um gráfico é constituído normalmente por pelo menos um filtro de cada tipo acima referenciado. Para uma melhor visualização existe uma aplicação que dá uma representação gráfica destas ligações, o GraphEdit. Na figura 2.5 encontra-se um gráfico simples, que abre um ficheiro MP3 e o reproduz.

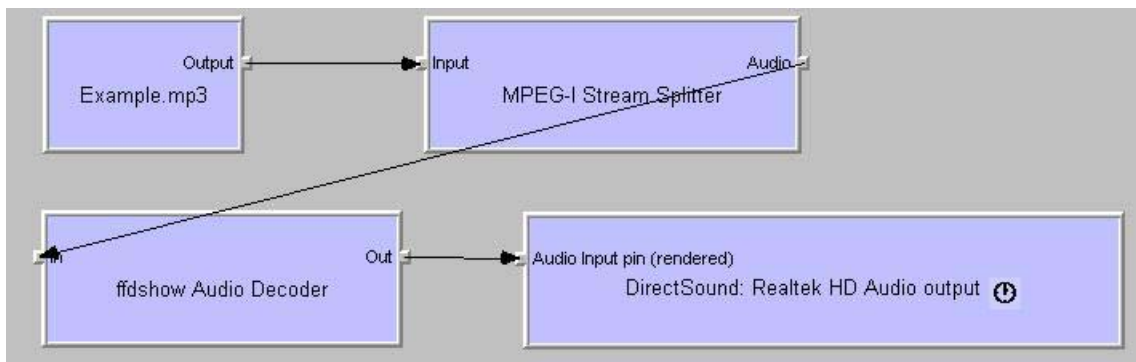


Figura 2.5: Exemplo de um Gráfico

Nesta imagem podemos verificar que as ligações se executam entre pinos existentes nos filtros. Esses pinos são objectos associados aos filtros e são responsáveis pelas transferências de dados entre os filtros. Quando os pinos são requisitados para se ligarem, estes irão verificar se existe compatibilidade entre eles tendo em consideração o tipo de ficheiros multimédia que cada filtro suporta. Dependendo do tipo de filtros podem existir mais do que um pino de entrada ou saída.

As aplicações DirectShow geralmente são desenvolvidas em C++, podendo também serem escritas em C, C# e Basic, não existindo uma API para o DirectShow. Mesmo que as aplicações sejam desenvolvidas em C++ pode-se utilizar funções escritas nas outras linguagens desde que as bibliotecas sejam incluídas no projecto. O ambiente de desenvolvimento usado foi o Microsoft Visual C++ Express Edition e na figura 2.6 encontra-se a interface da aplicação.

O DirectShow permite ao utilizador criar aplicações que são compatíveis com uma enorme variedade de ficheiros multimédia, podendo-se ainda adicionar filtros desenvolvidos por companhias externas à Microsoft, e colocar a aplicação compatível com esses ficheiros. Esta possibilidade leva à criação de um novo problema pois a existência de inúmeros filtros que fazem a mesma acção, quer seja decodificar ou codificar um tipo de ficheiro multimédia, torna a execução da aplicação mais lenta. Existe ainda os DirectShow Editing

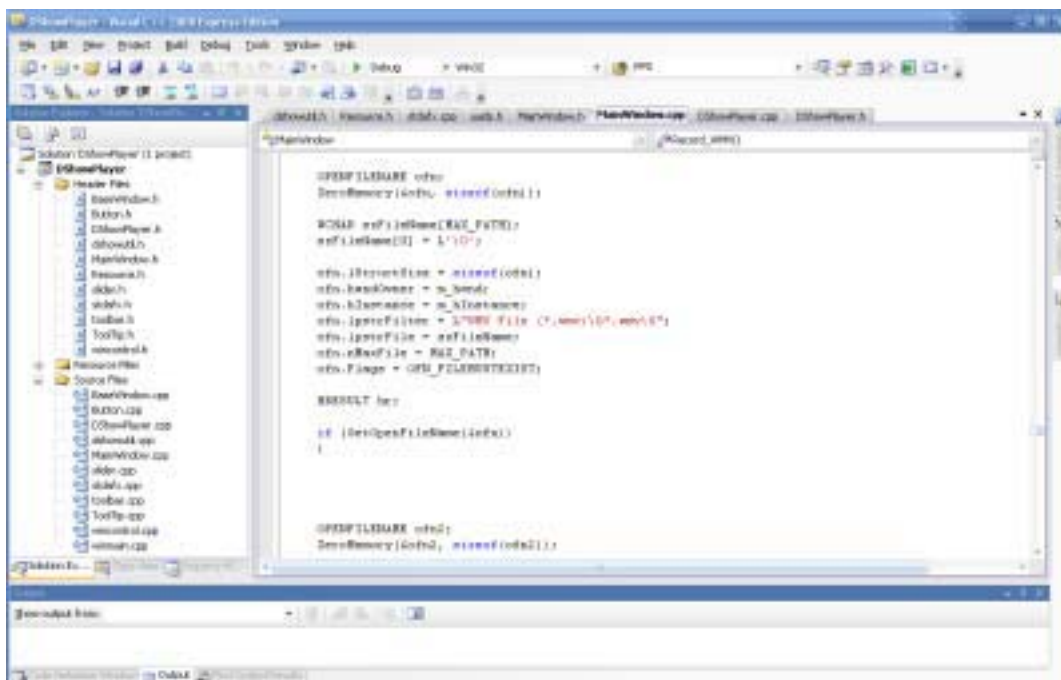


Figura 2.6: Interface de Desenvolvimento

Services que permitem modificações mais específicas dos ficheiros multimédia como por exemplo criar efeitos de vídeo e áudio, redimensionar, *frame-rate* automática e ainda outras.

A criação de filtros pelos criadores de aplicações o DirectShow revela-se muito complicada. Raramente se criam filtros a partir do zero, mas mesmo a criação de filtros utilizando como base classes já existentes revela-se muito complexa. Algumas empresas criam filtros do zero com especificações não genéricas e comercializam esses filtros. A comunidade de criadores de aplicações tem no sítio MSDN (*Microsoft Developer Network*) muita informação, aplicações exemplificativas e um fórum fornecido pela Microsoft no qual os utilizadores partilham ou comercializam informações, mas caracterizado como uma comunidade muito extensa e sociável.

2.6.4 Resumo

Todas as tecnologias têm as funcionalidades necessárias para a execução deste projecto, destacando-se no GStreamer a polivalência em relação ao S.O. no qual a aplicação tenciona ser executada, no Quicktime as ligações e funcionalidades específicas para o hardware fornecido pela Apple e no DirectShow as extensas funcionalidades, o grande número de filtros existentes, a grande comunidade tendo como principal vantagem o facto de ser desenvolvido e executado no S.O. Windows que é o S.O. mais difundido no mundo.

Capítulo 3

Arquitectura, requisitos e especificação

Este projecto foi proposto com o intuito de ser utilizado em eventos desportivos ocasionais e, como já foi dito anteriormente, este projecto prevê a captura de imagens numa piscina para posterior avaliação dos atletas ou a transmissão em tempo real de um evento, pretendendo-se capturar imagens e transmiti-las sem se utilizar fios.

No receptor de dados irá existir uma aplicação capaz de visualizar as imagens e permitir o seu armazenamento.

3.1 Arquitectura

O sistema deve ter um ou mais dispositivos de captura de vídeo e áudio, um emissor de dados, um receptor de dados e ainda um equipamento que permita o armazenamento, tratamento e visualização dos dados.

Existem inúmeros dispositivos de captura de vídeo no mercado, diferenciados pelas características que possuem e consequentemente pelos seus preços. Podendo-se considerar dois cenários diferentes para este sistema, como a captura de dados apenas para efeitos de treino dos atletas e a captura para emissão televisiva do evento desportivo, deve-se ter a possibilidade de escolher diferentes tipos de equipamento para estes dois cenários. Para o primeiro cenário escolher-se-ia um equipamento mais acessível em termos de preço mas com uma resolução e qualidade que satisfaça o propósito para o que foi idealizado. No segundo cenário teria que se exigir um dispositivo com qualidade profissional, dispositivo que tem um elevado valor de compra. As câmaras para estes dois cenários podem ter preços que podem variar de 1000 a 3000 euros no primeiro caso, mas no segundo caso os preços podem rondar os 40000 euros. Existem ainda muitas opções intermédias que se poderiam adaptar no projecto.

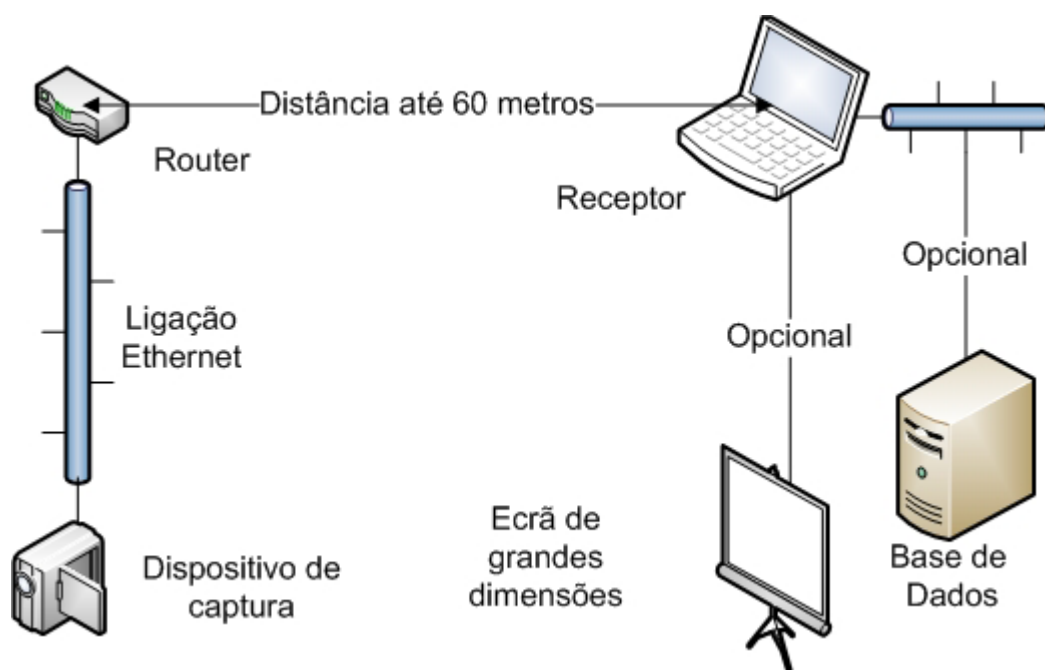


Figura 3.1: Arquitectura do sistema

Existem imensos conversores A/D no mercado, e a falta de decisão acerca do método de transmissão e do dispositivo de captura inibe a escolha de alguns conversores indicados para o projecto. A escolha deste dispositivo de hardware só poderá ser feita após a escolha do método de transmissão e do dispositivo de captura, dispositivo esse que pode trazer incorporado um conversor A/D.

3.2 Requisitos

O sistema tem como requisito a capacidade de transmissão de um grande débito de dados devido ao formato de imagem que se pretende obter no sistema, para possibilitar a recepção dos dados no receptor utilizando uma aplicação para os visualizar e armazenar. No receptor irá existir uma interface de interacção com o utilizador fornecendo-lhe algumas funcionalidades que são enumeradas no diagrama dos casos de utilização exemplificado na figura 3.2.

Estas serão as primeiras funcionalidades disponíveis ao utilizador, e tendo em conta a escolha efectuada activar-se-ão novas funcionalidades.

No caso do utilizador escolher o menu *Open File* terá duas opções; *Visualization*, permite ao utilizador abrir um ficheiro e visualizar o seu conteúdo, tendo as comuns funcionalidades de controlo, como *Play*, *Pause* e *Stop*, tendo ainda o controlo de volume, quer seja utilizando o botão *Mute* ou a barra de intensidade; e, terá ainda, a barra de controlo do instante em que se encontra a visualização do ficheiro multimédia. Se a escolha efectuada for alguma das opções existentes dentro do menu *Recording* as funcionalidades

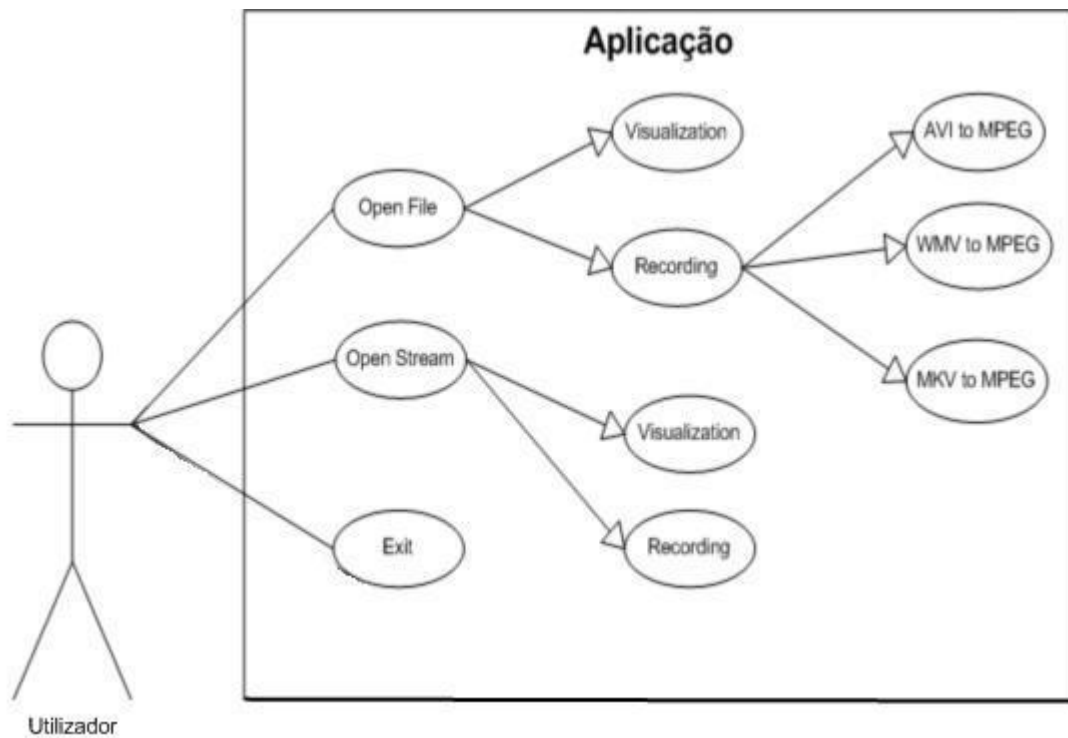


Figura 3.2: Diagrama de casos de utilização

de controlo de volume e a barra de controlo do instante estarão desactivadas como se poderá verificar na figura 3.3.

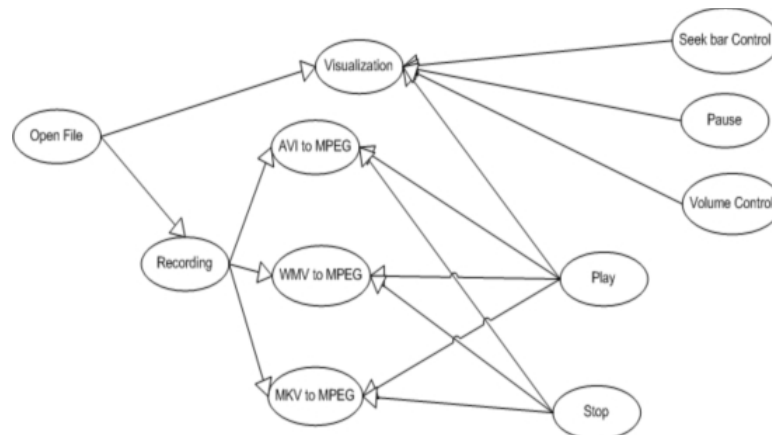


Figura 3.3: Diagrama de casos de utilização da escolha Open File

A escolha, no menu, da opção *Open Stream* pode ser efectuada ou para visualização, *Visualization*, ou para captura, *Recording*. Na visualização dos dados provenientes do dispositivo de captura existem funcionalidades de *Play*, *Pause*, *Stop* e controlo de volume, quer seja utilizando o botão *Mute* ou a barra de intensidade. Na captura o utilizador não dispõe da possibilidade de controlo de volume como se pode verificar na figura 3.4.

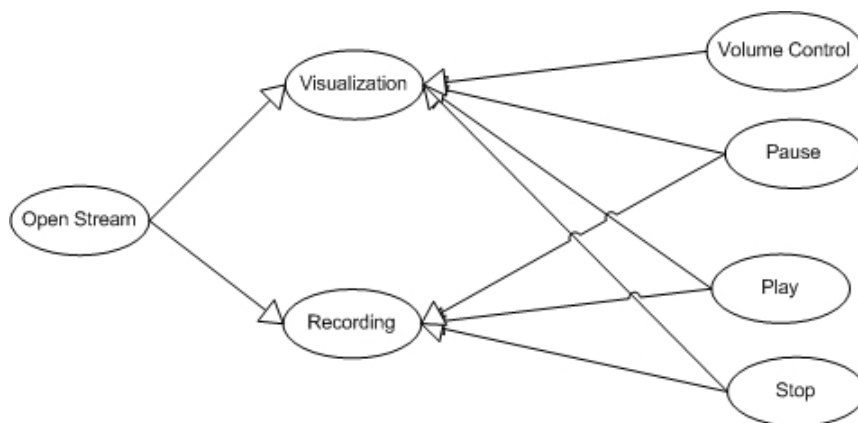


Figura 3.4: Diagrama de casos de utilização da escolha Open Stream

3.3 Usabilidade

A usabilidade oferecida por este sistema é, à partida, muito simples.

A utilização de uma tecnologia que não necessita de fios para a transferência de dados entre o dispositivo de captura e o receptor diminui os problemas inerentes ao uso de cabos num ambiente com muita humidade, e diminui os problemas existentes quando se tem demasiados cabos em instalações que não fornecem infra-estruturas funcionais para capturas de imagens em eventos desportivos.

No caso da aplicação a interacção é intuitiva. O uso de botões semelhantes aos utilizados em equipamentos de reprodução multimédia, as dimensões médias dos botões e barras tornam simples a interacção com a aplicação.

3.4 Compatibilidade

O sistema de transmissão de dados, é por si só, compatível com todos os sistemas operativos, mas a aplicação apenas funcionará num receptor que tenha um sistema operativo fornecido pela Microsoft, devido à tecnologia escolhida na programação da aplicação apenas funcionar nesses sistemas operativos.

A aplicação terá como restrição além do sistema operativo, a recodificação de apenas alguns tipos de ficheiros. A aplicação apenas recodificará para MPEG três tipos de ficheiros, AVI (*Audio Video Interleave*), WMV (*Windows Media Video*) e MKV (*Matroska Video*) mas a visualização de ficheiros multimédia será apenas restringida pelos *codecs* instalados no computador receptor. As câmaras utilizadas neste sistema terão de fornecer um filtro, normalmente distribuído pelo fabricante da câmara, e esse filtro deverá ser instalado no computador receptor.

3.5 Desempenho e fiabilidade

O desempenho e fiabilidade deste projecto podem ser vistos em duas partes, transmissão de dados e aplicação.

O desempenho e fiabilidade da transmissão de dados estão muito dependente da distância entre o emissor e o receptor, mas também do protocolo utilizado. A escolha do protocolo de transmissão vai influenciar a confiabilidade da ligação. Apesar das possíveis perdas de *frames* do vídeo, a transmissão de dados pode ser considerada fiável se a perda de *frames* for muito diminuta, pois um dos objectivos deste projecto será a transmissão televisiva ou a visualização das imagens obtidas num ecrã de grandes dimensões no evento desportivo.

O desempenho e fiabilidade da aplicação estarão apenas dependentes da máquina em que se encontra. Se o computador onde se encontra a aplicação tiver a capacidade de processamento necessária para a visualização, tratamento e armazenamento dos dados recebidos, o desempenho da aplicação será notoriamente satisfatória. A aplicação é estável e fiável não tendo problemas com a importação de vários tipos de ficheiros, desde que os *codecs* estejam instalados na máquina.

Capítulo 4

Implementação

4.1 Aplicação de interacção com o utilizador

Para se implementar a aplicação, que irá interagir com o utilizador do projecto, foi escolhido o DirectShow. Esta escolha deve-se às potencialidades da tecnologia e ao facto de ser indicada para o uso no S.O. Windows, o sistema mais utilizado mundialmente, o que fará com que a aplicação funcione na maior parte das máquinas existentes.

Na implementação da aplicação usou-se o Microsoft Visual C++ 2008 Express Edition, que com a instalação do Microsoft SDK e o Microsof.NET se tornou operacional na criação de aplicações baseadas no DirectShow. Esta instalação fornece as bibliotecas básicas para a criação de aplicações multimédia.

Inicialmente utilizou-se como base de implementação uma aplicação fornecida como exemplo no pacote de instalação da Microsoft SDK v6.0. Esta aplicação apenas tinha como funcionalidade a abertura de um ficheiro único para visualização, mas como é fornecida pela Microsoft num pacote de instalação criado para criadores de aplicações multimédia ainda contém muitas funcionalidades que não são exploradas. A aplicação permite que o criador adicione e ligue filtros, verifique se estão conectados, remove-los, não restringindo o criador ao uso das funcionalidades existentes, sendo possível adicionar-se novas. A interface final obtida após as modificações necessárias para este projecto encontra-se demonstrada na figura 4.4, na qual o utilizador tem várias opções que criará diferentes gráficos de execução.

O gráfico, na qual se poderão adicionar os filtros que são necessários para executar as funcionalidades da aplicação, é criado numa função específica que inicializa o gráfico. A *InitializeGraph()* destrói o gráfico anterior se existir algum previamente criado e inicializa várias funcionalidades da aplicação descritas no seguinte código:

```
//Tears down the old graph.  
TearDownGraph();
```

```

// Create the Filter Graph Manager.
hr = CoCreateInstance(CLSID_FilterGraph, NULL, CLSCTX_INPROC_SERVER,
IID_IGraphBuilder, (void**)&m_pGraph);
// All this lines add methods to the graph
//Methods for the playback of multimedia files
hr = m_pGraph->QueryInterface(IID_IMediaControl, (void**)&m_pControl);
//Exposes events occuring within the graph
hr = m_pGraph->QueryInterface(IID_IMediaEventEx, (void**)&m_pEvent);
//Controls the position of the media stream playback
hr = m_pGraph->QueryInterface(IID_IMediaSeeking, (void**)&m_pSeek);
//Determines the audio settings
hr = m_pGraph->QueryInterface(IID_IBasicAudio, (void**)&m_pAudio);
// Set up event notification.
hr = m_pEvent->SetNotifyWindow((OAHWND)m_hwndEvent, m_EventMsg,
NULL);

```

Estas inicializações são necessárias para todas as funcionalidades requeridas na aplicação, fornecendo informações acerca dos eventos de conexão de filtros, notificando o criador da aplicação para a existência de erros, fornecendo controlo da execução da aplicação e ainda funcionalidades de controlo do instante em que se encontra a visualização do ficheiro multimédia e do volume.

A criação da aplicação requer vários passos, como criar a janela com as especificações fornecidas, definir os botões e importar uma imagem para os preencher, criar a barra de controlo de volume, a barra de estado de visualização e o menu. Mas cada passo tem os seus requisitos e suas funções que no final permitem ao utilizador controlar a reprodução do ficheiro multimédia. O fundo da aplicação é modificado caso a aplicação não tenha um gráfico completamente operacional ou se o ficheiro multimédia não contiver vídeo, colocando um fundo azul. Ao se obter a situação reversa a aplicação inicializa o *video renderer*, que coloca o fundo negro, preparando a aplicação para os comandos de controlo fornecidos pelo utilizador. A visualização do vídeo e os comandos da aplicação não são afectados pelo redimensionamento da aplicação adaptando-se as novas dimensões sem causar problemas à aplicação.

Cada comando fornecido pela aplicação será controlado pela aplicação e o seu acionamento poderá modificar o estado de reprodução da aplicação. Quando o utilizador controla o volume ou a posição de reprodução do vídeo, existem funções responsáveis pela modificação desses estados. A mudança da posição de reprodução implica uma paragem na reprodução do ficheiro multimédia e uma actualização da nova posição seguida de uma

nova reprodução. Quando o gráfico é criado, a aplicação verifica se existe a possibilidade de obter a duração da reprodução para que a mudança da posição de reprodução seja possível, e se tal não acontecer desactiva esta funcionalidade.

Os botões existentes na aplicação são responsáveis pela mudança do estado de reprodução da aplicação. A aplicação pode estar em reprodução, em pausa ou parada. Foi definido que, após a criação de um gráfico, a aplicação permaneça no estado parado até que o utilizador pretenda iniciar o gráfico podendo depois colocá-lo em pausa ou voltar a para-lo.

Para se adicionar um filtro ao gráfico é necessário utilizar uma função específica de adição utilizando a chave CLSID, que identifica o objecto COM no qual se baseia o filtro, utilizando o seguinte código:

```
HRESULT hr = S_OK;

// Add a Avi Mux

hr = AddFilterByCLSID(m_pGraph, CLSID_AviDest, &pAviMux, L"AVI MUX");
```

Para se adicionar novos filtros, a sua chave CLSID terá que ser adicionada, se tal for necessário, num ficheiro que contém as chaves de todos os tipos e subtipos de multimédia e ainda as chaves dos filtros. No código seguinte encontram-se dois exemplos, um relacionado com um tipo de multimédia e outro com a identificação de um filtro:

```
// 73646976-0000-0010-8000-00AA00389B71 'vids' == MEDIATYPE_Video
OUR_GUID_ENTRY(MEDIATYPE_Video,
0x73646976, 0x0000, 0x0010, 0x80, 0x00, 0x00, 0xaa, 0x00, 0x38, 0x9b, 0x71)

// e436ebb5-524f-11ce-9f53-0020af0ba770 Async File Reader
OUR_GUID_ENTRY(CLSID_AsyncReader,
0xe436ebb5, 0x524f, 0x11ce, 0x9f, 0x53, 0x00, 0x20, 0xaf, 0x0b, 0xa7, 0x70)
```

A primeira implementação, focada no tratamento de ficheiros multimédia, deste projecto foi a introdução de uma função que permite que o criador consiga importar ou gravar, um ficheiro específico do GraphEdit, com as alterações e implementações efectuadas na aplicação. Assim após cada implementação será possível verificar se a ligação e se o objectivo idealizado foi executado com sucesso. Após cada mudança no código e com a ajuda do GraphEdit o utilizador pode ver o gráfico da aplicação, e nas duas figuras seguintes pode-se obter uma ideia da visualização, na figura 4.1 apenas temos um *Source Filter*, que abre um ficheiro MP3, e na figura 4.2 existem mais dois *Transform Filters* que vão descodificar o ficheiro importado.

Cada filtro na figura representa um conjunto de linhas de código que implementa uma acção na aplicação, quer seja a importar o ficheiro, modificá-lo ou reproduzi-lo. O



Figura 4.1: Gráfico que importa um ficheiro mp3

utilizador pode verificar que os filtros se encontram conectados quando existem ligações entre eles, e o preenchimento se encontra com uma cor diferente do *background*.

Em seguida foi alterada a interface da aplicação, colocando como opções para o utilizador a abertura de um ficheiro para visualização ou para conversão e gravação desse ficheiro num novo formato.

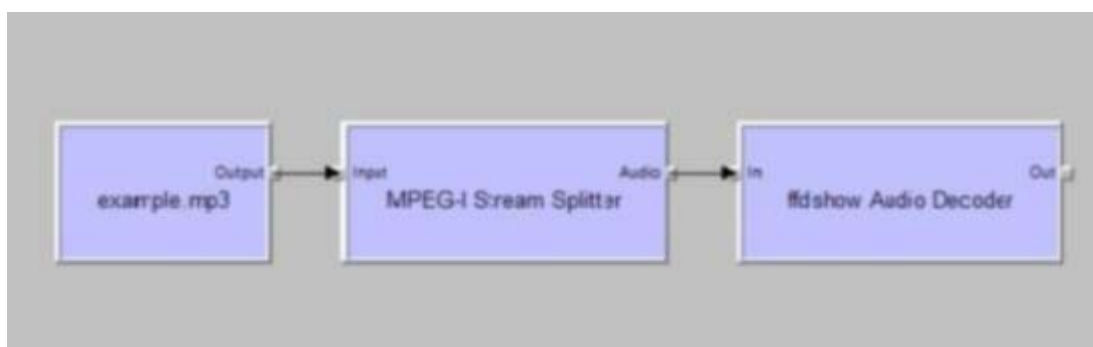


Figura 4.2: Gráfico que importa e descodifica um ficheiro mp3

Para reproduzir um ficheiro multimédia é necessário abrir o ficheiro que se pretende visualizar, adicionar o *VideoMixingRenderer* que será responsável pela imagem e o *DSound Renderer* responsável pela reprodução do áudio, ambos incorporados na *framework* da aplicação. Em seguida circulamos entre os pinos dos filtros existentes e utilizando o comando `RenderEx(pPin, AM_RENDEREX_RENDERTOEXISTINGRENDERERS, NULL)` a aplicação reconhece o tipo de ficheiro importado e adiciona todos os filtros necessários para reproduzir o ficheiro multimédia utilizando de preferência os filtros de reprodução previamente adicionados ao gráfico. Na figura 4.3 vê-se a reprodução de um ficheiro MPEG, sendo usados 3 filtros para se reproduzir o ficheiro multimédia.

Estas ligações automáticas só são possíveis utilizando uma funcionalidade existente no DirectShow, o *Intelligent Connect*, que ao se tentar ligar dois filtros incompatíveis ou reproduzir imediatamente um ficheiro multimédia, se encarrega de adicionar os filtros necessários para que as conexões sejam efectuadas com sucesso.

Esta aplicação já tinha funcionalidades de controlo de execução como *Pause* ou *Stop*, que permitem ao utilizador parar ou retomar a visualização, fornecidas pela interface *IME-*

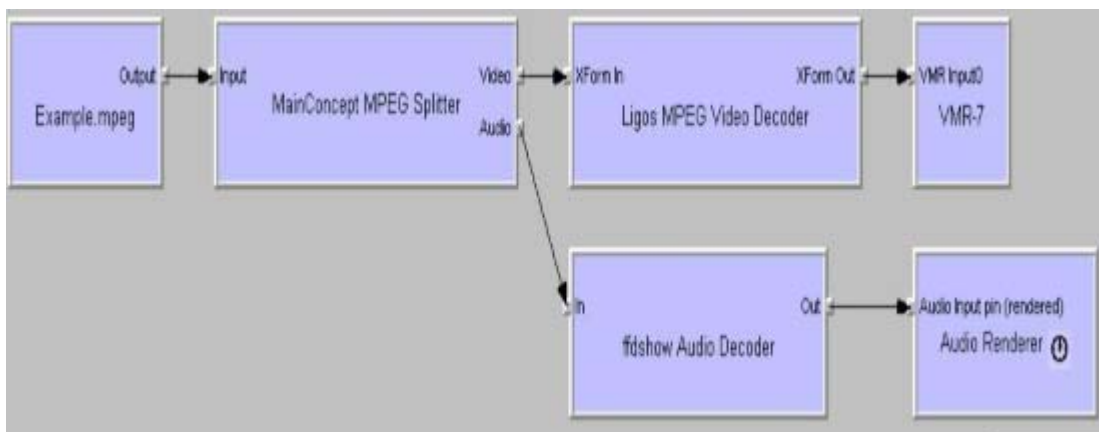


Figura 4.3: Gráfico de reprodução de um ficheiro MPEG

diaControl que pertence a interface *Filter Graph Manager*, incluída no DirectShow. Esta interface apenas fornece comandos de *Play*, *Pause* e *Stop* sendo necessária a introdução de um filtro específico para a gravação de um ficheiro multimédia. Neste caso as ligações não serão feitas automaticamente, isto é, serão manuais utilizando comandos como o *AddFilter* para adicionar filtros e conectando-os com o comando *Connect* fornecido pela interface *IfilterGraph Interface*, fornecida também pela interface *Filter Graph Manager*. Na figura 4.4 pode-se verificar que a interface da aplicação as funcionalidades referidas e ainda o controlo de volume e a barra de controlo do instante em que se encontra a visualização do ficheiro multimédia.



Figura 4.4: Aplicação de interacção com o utilizador

A aplicação utiliza a *Windows OpenFileDialog Box*, que é chamada pela função *GetOpenFileName()*, para receber o caminho da localização do ficheiro que se pretende

importar. Da mesma forma, a função *GetSaveFileName()* utiliza a *Windows SaveFile Dialog Box* para obter a localização de onde se pretende guardar o ficheiro final e o nome final do ficheiro. Se o utilizador fornecer um caminho inválido, para importar ou gravar um ficheiro, a aplicação abre uma janela de erro e não cria o gráfico.

No menu da aplicação temos duas escolhas principais: *Open File* para abrir um ficheiro já existente em algum dispositivo de armazenamento, para visualização ou codificação para MPEG; e *Open Stream* para visualizar os dados provenientes de um dispositivo de captura de vídeo ou para os gravar num dispositivo de armazenamento.

4.1.1 *Open File*

No menu da aplicação pode-se escolher abrir um ficheiro para visualização ou abrir um ficheiro específico para recodificação em MPEG. As escolhas possíveis podem ser vistas na figura 4.5, *Visualization* para apenas se visualizar o ficheiro e alguma das opções dentro de *Recording* para abrir e recodificar um ficheiro.

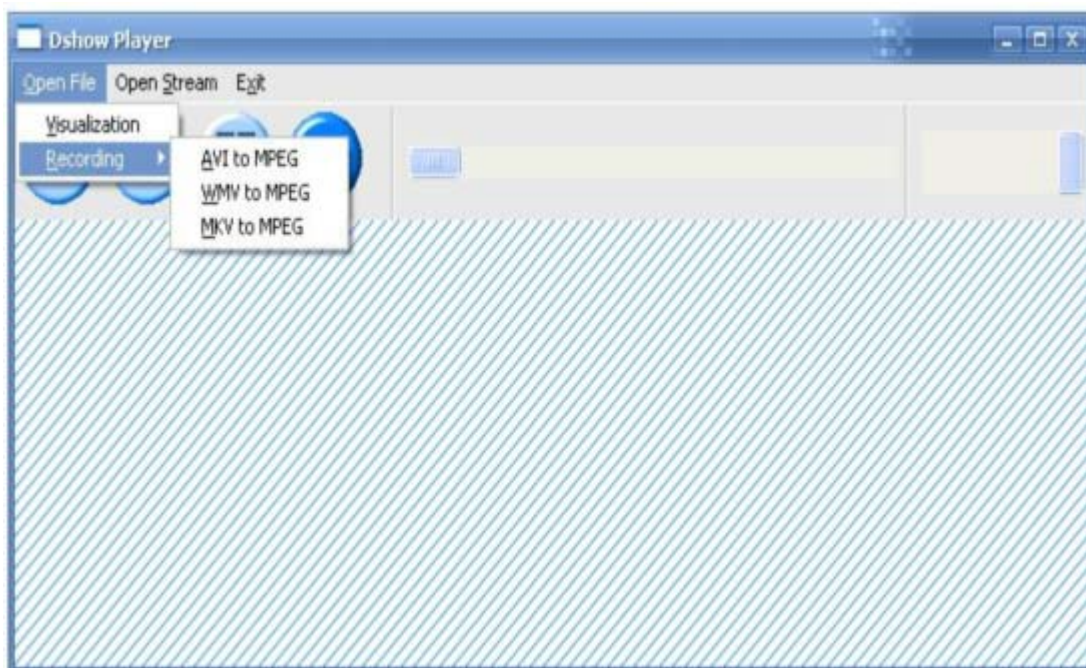


Figura 4.5: Menu de abertura de um ficheiro

Na escolha da opção *Visualization* a aplicação inicia um conjunto de acções que permitem a visualização, começando pela importação dos dados do ficheiro multimédia e inicializando o gráfico de execução. O gráfico da visualização depende do tipo de dados contidos no ficheiro, variando de caso para caso. O utilizador escolhe o ficheiro que deseja abrir, e a localização desse ficheiro é passada como parâmetro (*sFileName*) para uma função que adiciona o *Source Filter*, exemplificado no código seguinte:

```
HRESULT hr = S_OK;
```

```
//Inicialize the pointer
IBaseFilter *pSource = NULL;

//Add the Source Filter

hr = m_pGraph->AddSourceFilter(sFileName, NULL, &pSource);
```

Mesmo com certas variações, existem alguns filtros que estarão sempre presentes em todos os gráficos, como o *Source Filter*, que importa os dados do ficheiro, o *VMR-7*, que reproduz o vídeo e o *Audio Renderer*, que reproduz o áudio contido no ficheiro multimédia. Os filtros de transformação dos dados variam com o tipo de ficheiro importado sendo necessários mais filtros para certos ficheiros e menos para outros.

A introdução automática dos filtros rege-se por vários critérios, criando diferentes gráficos de acordo com o tipo de ficheiros importados, como se pode verificar na figura 4.3 em que se reproduz um ficheiro MPEG e na figura 4.6 onde os filtros são adicionados automaticamente para a reprodução de um ficheiro AVI. Utilizando o último caso como exemplo, pode-se indicar os passos efectuados pelo DirectShow na escolha dos filtros, revelando uma forma de escolhermos os filtros que são mais indicados para a acção pretendida, utilizando na mesma o *Intelligent Connect* que é responsável pela introdução automática dos filtros.

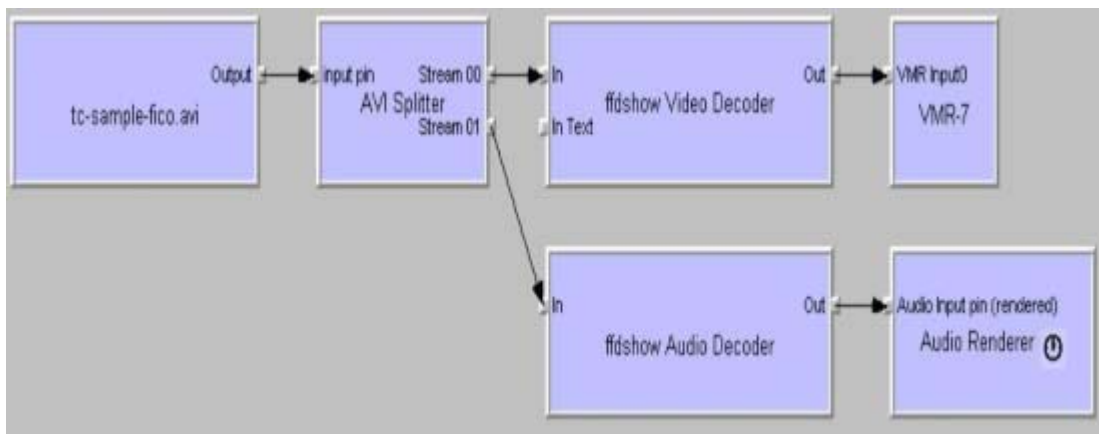


Figura 4.6: Gráfico de visualização de um ficheiro AVI

Em primeiro lugar o sistema procura saber qual é o tipo de *media* do ficheiro importado e sub-tipo, neste caso *MEDIATYPE_Stream* e *MEDIASUBTYPE_Avi*, e depois adiciona o *AVI Splitter* que divide os dados em *streams* individuais de vídeo e áudio, e assim sucessivamente para se conseguir atingir o objectivo de reprodução do ficheiro multimédia. Depois o sistema verifica o *Merit* do filtro, que pode ser descrito como a prioridade do filtro ser inserido em detrimento de outros, e que pode ter vários níveis desde o *Do Not Use* até ao *Preferred*, existindo inúmeros níveis intermédios. Será através deste atributo que o criador da aplicação poderá de alguma forma controlar a introdução de certos filtros através da modificação do *Merit* de um determinado filtro.

Se o utilizador pretender abrir um ficheiro para recodificação a inserção dos filtros terá de ser manual, pois o DirectShow apenas conclui o gráfico de forma automática ao se efectuar uma reprodução do ficheiro multimédia. A inserção manual de filtros vai restringir o tipo de ficheiros que podem ser importados para o *Source filter*, tendo sido escolhidos três tipos mais comuns. Na figura 4.7 demonstra-se os filtros necessários para descodificar e voltar a codificar o ficheiro multimédia.

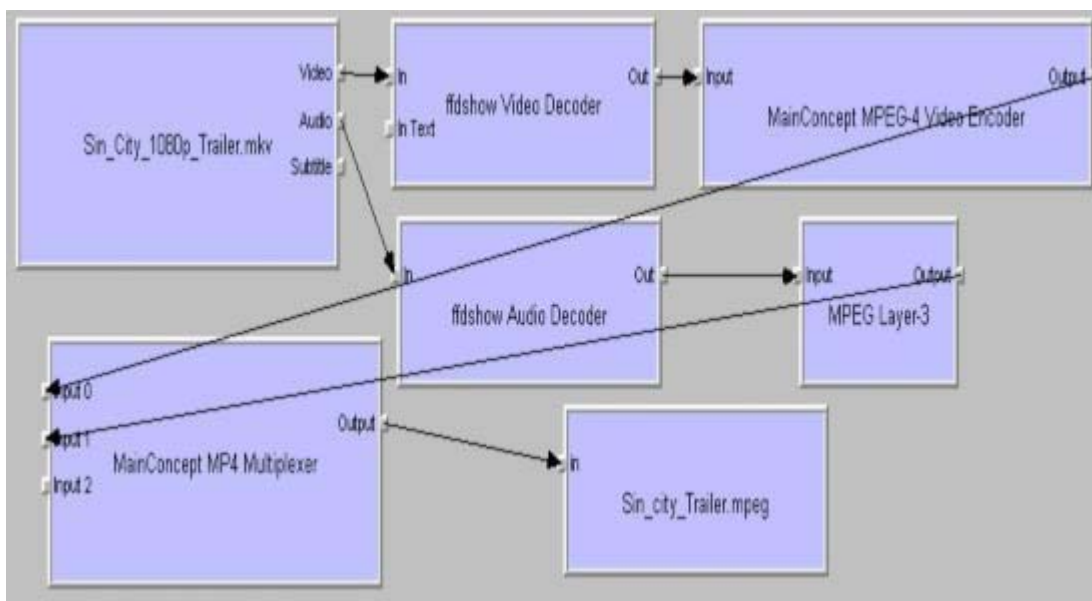


Figura 4.7: Gráfico da recodificação de MKV para MPEG

A criação de cada gráfico implica a execução de várias funções, cada uma responsável por uma acção indispensável para a criação e execução do gráfico. Num dos casos de recodificação é necessário recorrer a duas funções para se obter a localização do ficheiro que se vai abrir e do que se vai guardar. Em seguida adiciona-se o *Source Filter* e chama-se a função responsável pela criação do gráfico, na qual manualmente são adicionados todos os filtros necessários para recodificar o ficheiro e o armazenar. Além disso é necessário verificar se a funcionalidade de controlo de posição da reprodução pode ser activada, neste caso tal não é possível. Finalizando, é actualizado o volume, coloca-se o estado de reprodução em pausa e actualiza-se a interface de visualização activando todos os controlos de reprodução.

A introdução manual de um determinado filtro num gráfico requer que o filtro esteja registado no computador e, se for necessário um filtro específico, este pode ser fornecido com a aplicação, efectuando-se o registo no receptor tornando a aplicação funcional. Previamente à inserção de um filtro deve-se inicializar o gráfico, o apontador para o novo filtro e adicioná-lo ao gráfico, como se demonstra no seguinte código:

```
HRESULT hr = S_OK;
//Co-innicialize the graph
```

```

hr = CoCreateInstance(CLSID_FilterGraph, NULL, CLSCTX_INPROC_SERVER,
IID_IGraphBuilder, (void**)&m_pGraph);

//Innicialyze the filter pointer
IBaseFilter *pAudioRenderer = NULL;

// Add the filter
hr = AddFilterByCLSID(m_pGraph, CLSID_DSsoundRender, &pAudioRenderer, L"Audio Renderer");

```

Apesar da introdução de filtros ser efectuada de forma manual, apenas poderá ser necessária a introdução de filtros chave, pois o DirectShow adiciona filtros intermédios se verificar que são necessários. No caso de se tentar efectuar uma ligação entre o *AVI Splitter*, que divide o ficheiro multimédia em vídeo e áudio, e o *Audio Renderer*, o sistema verifica que falta um filtro para descodificar os dados e introdu-lo para descodificar o áudio. Em seguida liga o *AVI Splitter* a esse filtro e este ao *Audio Renderer*, podendo ser visualizada estas ligações no figura 4.6.

À mesma função responsável pela inserção e conexão de filtros, adicionou-se ainda código para a remoção de filtros inutilizados pelo gráfico. No caso do utilizador abrir um ficheiro multimédia que apenas contenha vídeo, os filtros responsáveis pelo áudio são removidos, para libertar recursos, como se pode ver na figura 4.8.

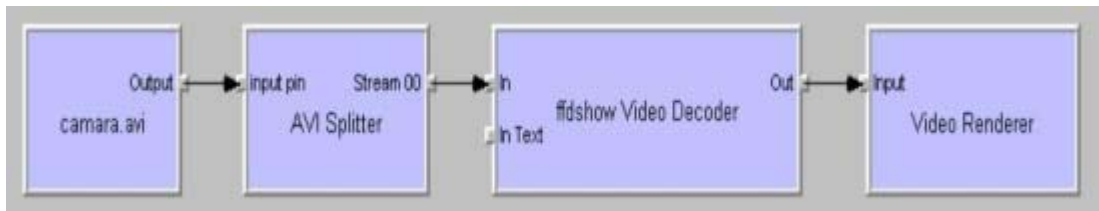


Figura 4.8: Gráfico da visualização de um ficheiro de vídeo

Ao se tentar abrir um ficheiro que não seja suportado ou fornecendo um caminho inexistente à aplicação, esta inicializa uma caixa de erro que informa o utilizador e termina a aplicação. Esta funcionalidade ao ser implementada, implicou o uso de variáveis de controlo, e todas as funções, executadas no decorrer de uma acção requerida pelo utilizador, usam variáveis para impedir erros na execução da aplicação.

4.1.2 *Open Stream*

No menu da aplicação, o utilizador pode escolher a opção *Open Stream* que lhe permite visualizar ou capturar as imagens provenientes de um dispositivo de captura. As opções disponíveis para o utilizador podem ser vistas na figura 4.9, tendo como opção *Visualization* para apresentar as imagens e *Recording* para as capturar.

Para a aplicação conseguir receber dados provenientes do dispositivo de captura, o fabricante desse dispositivo fornece um filtro que deverá ser instalado no computador

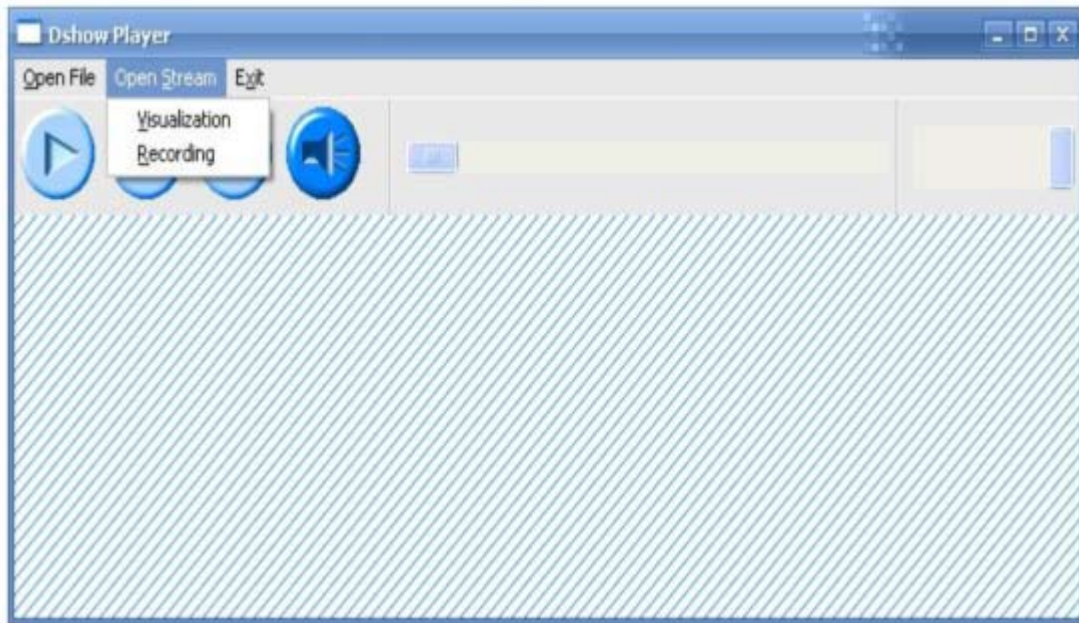


Figura 4.9: Menu de abertura de um dispositivo de captura

onde se está a executar a aplicação, e esse filtro será incorporado no gráfico para se poder visualizar ou capturar as imagens provenientes do dispositivo de captura.

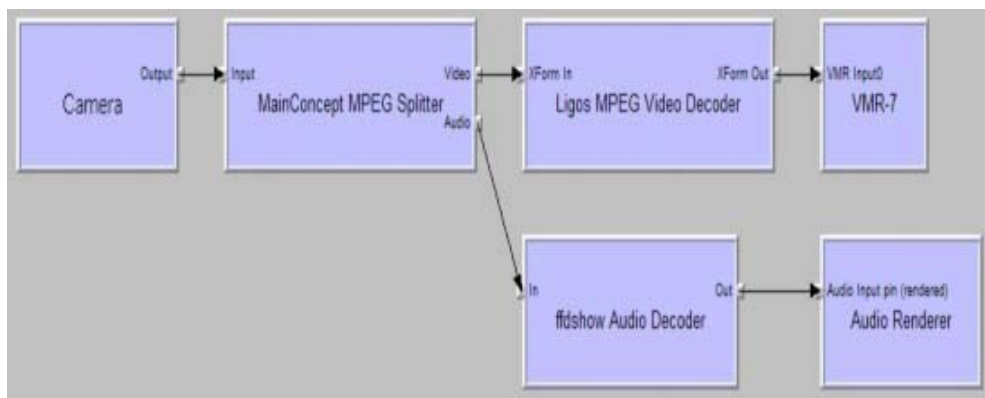


Figura 4.10: Gráfico da visualização do dispositivo de captura

Após o utilizador ter efectuado uma escolha dentro do menu *Open Stream*, a aplicação inicia uma acção, comum em ambas as opções existentes, caracterizada pela inicialização do gráfico adicionando, logo a partida, o filtro necessário para a recepção de dados provenientes do dispositivo de captura. A figura 4.10 mostra o gráfico obtido na visualização das imagens provenientes do dispositivo de captura que também foi criado de uma forma automática, mas ao contrário do caso anterior em que o utilizador seleccionava a localização do ficheiro e esse caminho era passado na aplicação para a adição de um *Source Filter*, neste caso utiliza-se o seguinte código, no qual se chama uma função que identifica o

dispositivo de captura e adiciona o filtro para depois ser conectado a outros para executar a função prevista.

```
pSrc=GetVideoDevice2();
pGraph2->AddFilter(pSrc, L"Video Capture Device");
```

A função identifica o dispositivo de captura através de funções no DirectShow que enumeram os dispositivos existentes tendo em conta a categoria na qual se encontra o dispositivo. Neste sistema será necessário identificar o dispositivo de captura de vídeo, *CLSID_VideoInputDeviceCategory*, e o código necessário para essa identificação encontra-se descrito em seguida:

```
IBaseFilter *GetVideoDevice()
IBaseFilter *pSrc= NULL;
ICreateDevEnum *pDevEnum = NULL;
IEnumMoniker *pClassEnum = NULL;
IMoniker *pMoniker = NULL;
CoCreateInstance(CLSID_SystemDeviceEnum, NULL, CLSCTX_INPROC, IID_ICreateDevEnum,
(void **) &pDevEnum);
pDevEnum->CreateClassEnumerator(CLSID_VideoInputDeviceCategory, &pClassEnum,0);
ULONG cFetched;
if(pClassEnum->Next(1,&pMoniker,&cFetched) ==S_OK)
pMoniker->BindToObject(0,0,IID_IBaseFilter, (void **) &pSrc);
pMoniker->Release();
pClassEnum->Release();
pDevEnum->Release();
return pSrc;
```

Esta função auxilia as funções principais de criação dos gráficos, neste caso fornecendo o *source filter* do gráfico, sendo necessário, no caso da *Visualization*, introduzir os outros filtros de forma automática para se reproduzir os dados recebidos. No caso do *Recording* é necessário introduzir manualmente todos os filtros necessários à modificação dos dados para que se possa proceder à captura, chamando uma outra função para a obtenção da localização final dos dados armazenados. Finalizando, actualiza o volume e coloca-se o estado de reprodução em pausa e actualiza-se a interface de visualização activando todos os controlos de reprodução.

O gráfico da visualização das imagens provenientes de um dispositivo de captura vai variar se esse dispositivo apenas captura vídeo ou se captura vídeo e som. Se o som for

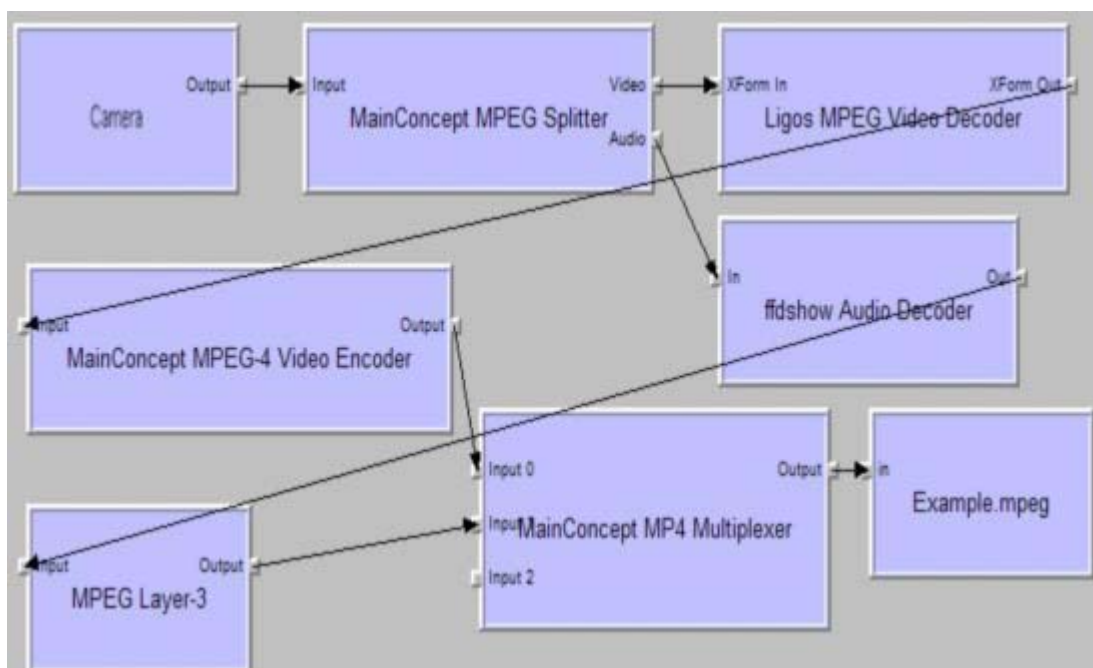


Figura 4.11: Gráfico de captura do dispositivo de captura

proveniente de outros dispositivos de captura, a aplicação fornece ao criador a possibilidade de adição de vários filtros utilizando uma função similar à demonstrada no código anterior, com mudanças para que a pesquisa se efectue sobre dispositivos de captura de áudio, *CLSID_AudioInputDeviceCategory*. Na figura 4.10 são demonstrados os filtros necessários para visualização de um dispositivo de captura que fornece vídeo e áudio. No caso de ser outro dispositivo de captura responsável pelo áudio, teríamos mais um *Source Filter* presente no gráfico.

A obtenção dos dados provenientes dos dispositivos de captura será efectuada no formato MPEG, e o utilizador terá de escolher a localização onde pretende arquivar os dados e na figura 4.11 demonstra o gráfico onde serão adicionados de forma manual todos os filtros necessários para a captura dos dados.

4.1.3 Resumo

A aplicação permite que o utilizador visualize em tempo real as imagens provenientes do dispositivo de captura, permitindo que o utilizador ajuste a posição do dispositivo para obter o ângulo de captura que pretende. Após os ajustes, o utilizador pode capturar vídeos dos eventos e armazená-los, e a aplicação permite que o utilizador possa ainda visualizar capturas efectuadas previamente e que se encontram armazenadas.

A aplicação tem todas as potencialidades necessárias para a visualização e gravação das imagens em alta definição provenientes dos dispositivos de captura.

4.2 Transmissão wireless

Nesta secção iremos abordar a largura de banda necessária para a transmissão de diferentes formatos de transmissão de vídeo. Na obtenção destes dados foram utilizadas duas aplicações distintas, o VLC (*VideoLan Client*) e o DU Meter. O VLC permite aos utilizadores visualizar ficheiros multimédia e transmiti-los pela rede, e com esta aplicação podemos testar a transmissão de vídeos de alta definição verificando as taxas de débito envolvidas, utilizando a aplicação DU Meter.

4.2.1 Débito necessário na transmissão

A tecnologia escolhida para ser utilizada neste sistema é baseada no protocolo 802.11. Com o objectivo de se aumentar as taxas máximas de transmissão e diminuir os atrasos, foram desligados os mecanismos de segurança da rede. Os testes foram efectuados com dois computadores ligados por *ethernet* a um *router*. Estes testes tinham como finalidade a obtenção das taxas de débito necessárias para a transmissão de vídeo de alta definição.

Os ficheiros multimédia utilizados nestes testes serão semelhantes aos fornecidos pelos dispositivos de captura de vídeo de alta definição. Foram utilizados ficheiros no formato HDV (*High-Definition Video*) com resoluções de 1280x720 (720p) e 1440x1080 (1080i). Estes são os formatos de alta definição mais comuns em capturas efectuadas por dispositivos que não são profissionais, sendo o formato HDTV (*High-definition television*), que tem como formatos de alta definição mais comuns o 720p (1280x720) e o 1080p/i (1920x1080), o que é geralmente utilizado nas capturas para emissões televisivas.

Nos testes utilizaram-se ainda ficheiros de formatos SD (*Standard Definition*) para se verificar as diferenças existentes, relativamente, à taxa de dados necessária para o sistema. Na transmissão de vídeo com resolução 720p verificou-se que era necessário transmitir dados à taxa de aproximadamente 21Mbps, no vídeo com resolução de 1080i seria necessário 27Mbps. Na tabela 4.1 encontra-se as diferenças entre os vários formatos testados.

Sistema (sem compressão)	Resolução	Frame-Rate (fps)	Débito (Mbps)
576 PAL	720x576	25	6,8
720p	1280x720	29.97	21
1080i	1440x1080	29.97	27

Tabela 4.1: Débito necessário para transmissão de vídeo

Os valores obtidos encontram-se dentro de uma gama de valores que pode ser transmitida utilizando a tecnologia 802.11g, mas apenas para distâncias curtas. O débito de transmissão diminui à medida que a distância entre o emissor e o receptor aumenta, levando à impossibilidade de utilização desta tecnologia para a distância proposta para este projecto.

Os valores obtidos em 4.1 são valores de pico e não valores médios. Estes valores são influenciados pelas cores existentes no vídeo, porque o sistema de compressão é mais eficiente se o vídeo tiver cores mais escuras e em menor número do que um vídeo com muita claridade e com muitas cores. Como se vê na figura 4.12 o débito binário varia ao longo da transmissão do mesmo ficheiro multimédia.



Figura 4.12: Gráfico da utilização da largura de banda

4.2.2 Teste às tecnologias 802.11

O protocolo 802.11g consegue transmitir vídeo de alta definição, pois as taxas de dados envolvidas (tabela 4.1) são inferiores ao débito máximo da tecnologia, mas apenas a curtas distâncias. Com o aumento da distância, entre o emissor e o receptor, o débito máximo irá diminuir progressivamente obtendo-se valores inferiores aos necessários para a distância máxima requerida no projecto. Na tabela 4.2 pode-se obter uma ideia dessa perda de capacidade com o aumento da distância.

Distância (m)	Débito (Mbps)
15	48
30	18
45	10
60	6

Tabela 4.2: Tabela da diminuição do débito com a distância

Foram testadas transmissões utilizando esta tecnologia e foram visualizados com sucesso, vídeos em 1080i até distâncias de 20 metros e 720p até distâncias de 25 metros. Após estes limites a imagem degradava-se ocorrendo ainda perdas de *frames* e até o bloqueio da transmissão.

O protocolo 802.11n ainda se encontra em desenvolvimento, mas já foram aprovados alguns *drafts* da tecnologia, tendo sido produzidos equipamentos que utilizam o protocolo. O 802.11n consegue obter taxas de dados superiores à 802.11g, conseguindo obter 70Mbps a 100 metros de distância. Este débito permite a transmissão de vídeo de alta definição, quer seja 720p ou 1080i, à distância proposta inicialmente para este projecto.

4.2.3 Resumo

As tecnologias baseadas no protocolo 802.11 têm a capacidade necessária para transmitir vídeo de alta definição HDV. A tecnologia 802.11g permite transmitir HDV mas tem como distância limite 20 metros para a resolução 720p e 25 metros a 1080i, tornando-se incapaz de satisfazer o requisito de transmissão de vídeo em que a distância entre o emissor e o receptor é de 60 metros.

A tecnologia 802.11n consegue satisfazer todos os requisitos necessários neste projecto podendo transmitir vídeo de alta definição a distâncias até superiores a 60 metros e, dependendo da tecnologia MIMO utilizada, podendo-se obter taxas de transmissão a curtas distâncias 10 vezes superiores à necessária. Em campo aberto consegue obter débito de 70 Mbps à distância de 100 metros.

Capítulo 5

Conclusões e Trabalho Futuro

O estudo proposto centrou-se na procura de uma solução que permitisse a transmissão de vídeo de alta definição entre dispositivos de captura de vídeo e um dispositivo de recepção, utilizando as tecnologias de transmissão de dados sem fios existentes no mercado, tendo sempre em conta a evolução dessas tecnologias e normas futuras. No estudo verificou-se que apenas tecnologias de grande débito binário, que consigam garantir esse débito a uma distância de transmissão de 60 metros, poderão ser utilizadas para a transmissão de vídeo de alta definição. Estas restrições definiram à partida que tecnologias como o Bluetooth e o UWB não poderiam ser utilizadas neste estudo, tendo sido feito testes apenas nas tecnologias 802.11g e 802.11n.

A codificação de dados revelou-se muito importante neste estudo, pois sem a compressão de dados as taxas de transferência necessárias para a transmissão de vídeo de alta definição estariam entre 100 e 160 MByte/s. A escolha do método de compressão que será utilizado foi definida no início do estudo, e será usado o MPEG para a captura dos dados provenientes dos dispositivos de entrada. Os dados irão ser comprimidos utilizando um método de compressão fornecido pelo fabricante, sendo normalmente utilizado o MPEG. Essa compressão tornará o débito de dados necessário muito inferior, permitindo a transmissão dos mesmos utilizando tecnologias *wireless*.

Os testes efectuados na transmissão de dados entre o emissor e o receptor revelaram que o protocolo 802.11n será o mais indicado e capaz de satisfazer todos os requisitos necessários. O protocolo 802.11g consegue transmitir vídeo de alta definição, quer o 720p como o 1080i, a pequenas distâncias tornando-se incapaz depois dos 20 a 25 metros. O 802.11n consegue obter taxas de transmissão de 70 Mbps com 100 metros de distância entre o emissor e o receptor.

A aplicação desenvolvida tem como objectivo provar que a solução proposta neste estudo se adequa ao proposto, permitindo ao utilizador visualizar o vídeo de alta definição proveniente dos dispositivos de captura. A aplicação tem as funcionalidades básicas de

um reprodutor de ficheiros multimédia e tem uma interface que permite uma interacção simples e funcional com o utilizador.

A implementação da aplicação em DirectShow permitiu adicionar várias funcionalidades ao sistema, fornecendo uma interface de criação poderosíssima permitindo atingir todos os objectivos propostos para comprovar a viabilidade da solução proposta.

Esta solução permite cobrir a lacuna existente na captura ou transmissão de eventos desportivos ocasionais em infra-estruturas que não tenham características funcionais adequadas para a instalação de dispositivos com fios, e diminui a dificuldade da logística de instalação e diminui ainda o hardware necessário. Pode ainda ser utilizada para a retransmissão de vídeo em ecrãs de grandes dimensões para fornecer uma nova perspectiva aos espectadores do evento.

No futuro, a aplicação desenvolvida poderá ser melhorada em alguns aspectos. A aplicação poderá ser implementada aumentando o número de recodificações possíveis, sendo possível ao utilizador abrir todo o tipo de ficheiros recodificando-os para MPEG. A aplicação poderá ainda ser implementada para suportar metadados, que darão ao utilizador inúmeras informações acerca dos dados multimédia que reproduz, como o título, duração, autor e ainda outros.

Além disso a visualização dos dados provenientes do dispositivo de captura, que nesta aplicação poderá ser visualizada simultaneamente à captura.

Referências

- [1] European Telecommunications Standards Institute, editor. *3GPP TM TS 45.-series specifications*.
- [2] European Telecommunications Standards Institute, editor. *Release 97, TR 01.01*, 1997.
- [3] European Telecommunications Standards Institute, editor. *3GPPTM TS 25.-series specifications*.
- [4] Carlos Alexandre Fujimura and Robson Velasco Garcia. Uwb aplicacoes indoor, Fevereiro 2006. Consultada em Novembro de 2007, disponível em <http://www.teleco.com.br/tutoriais/tutorialuwbII/Default.asp>.
- [5] N.H. Lehmann and A.M. Haimovich. The power spectral density of a time hopping uwb signal: a survey. *Ultra Wideband Systems and Technologies, 2003 IEEE Conference on*, pages 234–239, Nov. 2003.
- [6] Joo A. Ney da Silva and Marcello L. R. de Campos. Spectrally efficient uwb pulse shaping with application in orthogonal psm. *Communications, IEEE Transactions on*, 55(2):313–322, Feb. 2007.
- [7] IEEE P802.11 TASK GROUP N. Standard for enhancements for higher throughput, Maio 2008. Consultado em Junho de 2007, e disponível em <http://grouper.ieee.org/groups/802/11/Reports/>.
- [8] The Working Group for WLAN Standards. Ieee 802.11tm wireless local area networks, Maio 2008. Consultado em Maio de 2007, e disponível em <http://www.ieee802.org/11/>.
- [9] Yang Xiao. Packing mechanisms for the ieee 802.11n wireless lans. *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, 5:3275–3279 Vol.5, Nov.-3 Dec. 2004.
- [10] BroadCom Inc. 802.11n speed, Junho 2005. Consultado em Dezembro de 2007, e disponível em <http://80211n.com/80211n-speed.html>.
- [11] Mohammed Ghanbari. *Standard Codecs: Image Compression to Advanced Video Coding*. Institution of Electrical Engineers, 2003.
- [12] Borivoje Furht Raymond Westwater. *Real-time Video Compression: Techniques and Algorithms*. Springer, 1996.

- [13] S. Wenger. H.264/avc over ip. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):645–656, July 2003.
- [14] Ramjee Prasad e Marina Ruggieri. *Technology Trends in Wireless Communications*. Artech House, 2003.
- [15] Stefan Winkler. *Digital Video Quality Vision Models and Metrics*. Genista Corporation, Montreux, Switzerland, 2003.
- [16] Colin Perkins. *RTP: Audio and Video for the Internet*. Addison-Wesley, 2003.
- [17] Dapeng Wu, Y.T. Hou, and Ya-Qin Zhang. Scalable video transport over wireless ip networks. *Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on*, 2:1185–1191 vol.2, 2000.