



Fernando Rodrigues Morais

# APLICAÇÃO DE TRANSFORMADAS VARIÁVEIS ESPECIALMENTE À CODIFICAÇÃO DE VÍDEO

Júri:

Professor Doutor José Manuel Fernandes Craveirinha  
Professor Doutor Vítor Manuel Mendes da Silva  
Professor Doutor Luís Alberto da Silva Cruz (Orientador)

Coimbra  
Julho 2012



UNIVERSIDADE DE COIMBRA



Universidade de Coimbra  
Faculdade de Ciências e Tecnologia  
Departamento de Engenharia Eletrotécnica e de Computadores

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores;  
Ramo de Especialização em Telecomunicações

# **APLICAÇÃO DE TRANSFORMADAS VARIÁVEIS ESPACIALMENTE À CODIFICAÇÃO DE VÍDEO**

**Fernando Rodrigues Morais**

Júri:

**Professor Doutor José Manuel Fernandes Craveirinha  
Professor Doutor Vítor Manuel Mendes da Silva  
Professor Doutor Luís Alberto da Silva Cruz (Orientador)**

Coimbra  
Julho 2012



## Resumo

Um sinal de vídeo necessita de um elevado número de bits para ser representado no seu formato original, o que faz com que as aplicações de vídeo não-comprimido não sejam viáveis devido à limitação de largura de banda dos canais de comunicação e ao limite de armazenamento. Por essa razão, codificadores de vídeo são usados em aplicações de vídeo digital para comprimirem o sinal de vídeo, para que possa ser representado com um número menor de bits. O objetivo de todos os codificadores de vídeo é maximizar a qualidade de vídeo e ao mesmo tempo diminuir a taxa de bits. O projeto de investigação aqui representado tem como ponto central o estudo e desenvolvimento de métodos para contribuir para a melhoria na eficiência de codificação, através do desenvolvimento e investigação de soluções para redução de complexidade computacional de transformadas espacialmente variáveis (SVT).

**Palavras-chave:** vídeo digital, codificação de vídeo, redução de complexidade, SVT.

## **Abstract**

A video signal requires a very large number of bits to be represented in its uncompressed form. This fact makes uncompressed video transmission and storage not feasible in general, due to the limited bandwidth of communication channels and the limit of the storage. To alleviate these problems video coders are used in digital video applications, to compress the video signal, so that a smaller number of bits is required for its representation. The goal of all video coders is to maximize the video quality and at the same time decrease the bitrate. The research project present in this text has as its main objective the study and development of methods to contribute to the improvement in coding efficiency, through the development and research of solutions to reduce computational complexity of spatially varying transforms (SVT).

**Keywords:** digital video, video coding, complexity reduction, SVT.

# Índice

Capítulo 1: Introdução.....	11
1.1 – Motivação e Objetivos .....	12
1.2 – Organização da Monografia .....	12
Capítulo 2: Estado da Arte .....	15
2.1 – Fundamentos de Vídeo Digital.....	15
2.2 – Compressão com e sem perdas.....	16
2.3 – Sistema de Cores e Subamostragem de Cores .....	17
2.4 – Redundância de Dados .....	19
2.4.1 – Redundância Espacial .....	19
2.4.2 – Redundância Temporal .....	20
2.4.3 – Redundância Entrópica .....	20
2.5 – A norma H.264/AVC .....	20
2.6 – Transformadas Espacialmente Variáveis .....	25
2.6.1 – Parâmetros de localização .....	26
Capítulo 3: Trabalho Desenvolvido .....	29
3.1 Introdução de novos formatos de Bloco SVT .....	29
3.1.1 Seleção dos Parâmetros de Localização .....	30
3.2 Decisão de Posicionamento e Formato de Bloco SVT .....	31
3.3 Estrutura da função SVT no código TENTM .....	34
3.4 Alterações realizadas no código TENTM .....	35
3.4.1 – Alterações realizadas para a decisão de formato e posicionamento de bloco SVT .....	36
3.4.2 – Alterações realizadas para introdução de novos formatos de bloco SVT .....	36
3.5 Resultados Experimentais .....	37
Capítulo 4: Conclusão .....	49
Bibliografia.....	53

## Lista de Figuras

Figura 1 - Formatos de subamostragem de crominância (adaptado de [6]) .....	18
Figura 2 - Modos de predição intra-trama para blocos de 4x4 <i>pixels</i> (adaptado de [6]).....	22
Figura 3 - Modos de predição intra-trama para blocos de 16x16 <i>pixels</i> (adaptado de [6]).....	22
Figura 4 - Estimação de movimento (adaptado de [6]) .....	23
Figura 5 - Diagrama em blocos de um codificador H.264/AVC .....	24
Figura 6 - Diagrama em blocos de um decodificador H.264/AVC .....	25
Figura 7 - SVT utilizando a transformada 8x8 [9] .....	26
Figura 8 - SVT utilizando as transformadas 16x4 e 4x16 [11] .....	26
Figura 9 - SVT utilizando as transformadas 4x4, 4x8 e 16x8.....	30
Figura 10 - Posicionamento do bloco SVT obtido através do cálculo do centro de massa da SAD das linhas e das colunas.....	32
Figura 11 - Posicionamento do bloco SVT obtido através da linha e coluna com maior SAD	33
Figura 12 - Posicionamento dos blocos 4x16 (vermelho), 8x8 (azul) e 8x4 (verde) obtido através do centro de massa .....	34
Figura 13 – Pequeno excerto da função <i>encode_luma_SVT_P()</i> no código TENTM [10].....	35
Figura 14 - Aplicação do método Bjontegaard Delta.....	39
Figura 15 - Curva R-D da sequência BasketballPass .....	41
Figura 16 - QPs em destaque da Curva R-D da sequência BasketballPass.....	41
Figura 17 - Curva R-D da sequência BQTerrace .....	42
Figura 18 - QPs em destaque da Curva R-D da sequência BQTerrace .....	42

## Lista de Tabelas

Tabela 1 - Parâmetros de Localização para os novos formatos de blocos SVT.....	31
Tabela 2 - Características das sequências usadas .....	38
Tabela 3 - Redução da Taxa de bits .....	40
Tabela 4 - Redução do tempo de codificação (QP=22) .....	43
Tabela 5 - Redução do tempo de codificação (QP=27) .....	44
Tabela 6 - Redução do tempo de codificação (QP=32) .....	44
Tabela 7 - Redução do tempo de codificação (QP=37) .....	45
Tabela 8 - Redução do tempo de codificação – Média dos diferentes QPs em todas as sequências utilizadas .....	46



## **Acrónimos e Siglas**

*AVC – Advanced Video Coding*

*Cb – Chrominance blue*

*CIF – Common Intermediate Format*

*Cr – Chrominance red*

*DCT – Discrete Cosine Transform*

*HDTV – High Definition Digital Television*

*ISO – International Organization for Standardization*

*ITU-T – International Telecommunication Union – Telecommunication*

*JVT – Joint Video Team*

*LP – Location Parameter*

*MC – Motion Compensation*

*ME – Motion Estimation*

*MPEG – Moving Picture Experts Group*

*PSNR – Peak signal-to-noise ratio*

*QP – Quantization Parameter*

*RD – Rate-Distortion*

*RDO – Rate-Distortion Optimization*

*RGB – Red, Green, Blue*

*SAD – Sum of Absolute Differences*

*SVT – Spatially Varying Transforms*

*TENTM – Tandberg, Ericsson and Nokia Test Model*

*VCEG – Video Coding Experts Group*

*VGA – Video Graphics Array*

*YCbCr – Luminance, Chrominance Blue, Chrominance Red*





# Capítulo 1: Introdução

Tem-se assistido ao longo dos últimos anos a uma grande evolução no que diz respeito à tecnologia de vídeo digital. Durante este período, a qualidade de vídeo melhorou de forma significativa, a criação de conteúdos e a sua distribuição tornou-se mais simples, os sistemas de vídeo foram integrados em inúmeras aplicações, como TV digital e dispositivos móveis, entre outras. Apesar deste avanço tecnológico os codificadores e decodificadores de vídeo atuais requerem um grande esforço computacional na realização das suas operações. Devido a esta complexidade computacional elevada que é requerida para o processamento de vídeos digitais em tempo real, é necessário o desenvolvimento de circuitos integrados específicos, já que de outra forma, com soluções em *software*, seriam necessários processadores de uso geral e de alto desempenho, o que levaria a um aumento do consumo energético e inviabilizaria a execução de vídeos digitais em alguns dispositivos portáteis, tais como *smartphones*, *tablets*, *PDA*s. Além disso, como o processamento de vídeo digital envolve uma quantidade enorme de dados, torna-se inviável a utilização deste tipo de vídeo em aplicações de tempo real sem o uso de técnicas de compressão. Os codificadores atuais baseados no modelo híbrido usados para compressão de vídeo digital, apresentam diversas etapas de processamento, cada uma delas com o objetivo de explorar algum tipo de redundância existente nos vídeos. Desta forma, é possível reduzir a quantidade de dados a ser armazenada e transferida, com mínimas perdas na qualidade de imagem ou até sem perda alguma.

Um vídeo digital é composto por uma sequência de imagens com as mesmas dimensões apresentadas a um ritmo temporal regular, denominada taxa temporal (em tramas por segundo). Esta taxa define quantas imagens são apresentadas por segundo e é fundamental para a qualidade final do vídeo apresentado. Um vídeo digital não comprimido contém um volume extremamente elevado de dados, o que dificulta muito a sua manipulação por dispositivos eletrônicos, apesar de todo o avanço tecnológico.

Por exemplo, um vídeo de 15 minutos com resolução 1080p (1920 x 1080), tipicamente usada em televisão digital de alta definição (HDTV), com uma taxa temporal de 24 imagens por segundo e 24 bits por *pixel* utilizaria 128 GB num disco de armazenamento. A taxa de transmissão necessária para comunicação em tempo real seria de aproximadamente 1 Gbit por segundo. Um canal com largura de banda de 6 MHz, com máximo débito de dados de 20 Mb/s, teria que ter uma relação de compressão 50:1. Compreende-se com este exemplo,

a enorme importância da compressão de vídeos digitais para fins de transmissão e armazenamento.

De entre as normas de compressão existentes, a norma H.264/AVC (*Advanced Video Coding*) [1], desenvolvida pela JVT (*Joint Video Team*), uma equipa de especialistas do VCEG do ITU-T [2] e do MPEG do ISO/IEC [3] é a mais eficiente em termos de desempenho na compressão de dados, com custo de uma maior complexidade computacional. Desde a sua aprovação em 2003, a norma H.264/AVC, tem vindo a ser investigada exaustivamente. Nos últimos anos, com a evolução da capacidade computacional dos equipamentos eletrónicos, foram possíveis imensas contribuições para melhorar a eficiência desta norma, tornando a sua implementação mais fácil em *software* para comunicação em tempo real.

## **1.1 – Motivação e Objetivos**

Com todas as ferramentas e técnicas de compressão incluídas na norma H.264/AVC, a complexidade computacional continua a ser uma preocupação na implementação de codecs em *software* para comunicação em tempo real. O projeto de investigação aqui exposto tem como principal objetivo o estudo, desenvolvimento e aplicação de transformadas variáveis espacialmente à codificação de vídeo. O trabalho desenvolvido pretende contribuir para a melhoria na eficiência de codificação e para a redução da complexidade de codificação, através do desenvolvimento e investigação de soluções para redução de complexidade computacional de transformadas espacialmente variáveis (SVT).

## **1.2 – Organização da Monografia**

O Capítulo 2 deste texto apresenta uma revisão do estado da arte em codificação de vídeo e em redução de complexidade para codificação de vídeo, assim como alguns conceitos sobre compressão de vídeo digital. Mais especificamente, a secção aborda as ferramentas de compressão presentes na norma H.264/AVC, com especial ênfase na técnica SVT.

O Capítulo 3 apresenta os algoritmos desenvolvidos no contexto deste projeto de tese e as suas implementações, assim como uma descrição detalhada dos testes efetuados e dos resultados obtidos.

O Capítulo 4 finaliza este texto com as conclusões retiradas dos resultados alcançados através dos testes efetuados e descritos no Capítulo 3.



## Capítulo 2: Estado da Arte

Neste capítulo, são apresentados resumidamente alguns conceitos fundamentais sobre vídeos digitais e também sobre a sua forma de representação. Será também apresentada uma revisão sobre o funcionamento da norma H.264/AVC, o atual estado da arte em codificação de vídeo. Por último é apresentada a técnica de Transformadas Espacialmente Variáveis (SVT), o tema central deste projeto de investigação.

### 2.1 – Fundamentos de Vídeo Digital

Um vídeo digital consiste numa sequência de imagens, com a mesma dimensão, captadas com um determinado intervalo de tempo entre as mesmas. Para que seja obtida uma sensação de movimento contínuo, ou seja, de tempo real, a taxa de captura entre imagens deve ser de pelo menos 24 a 30 imagens a cada segundo dependendo do ambiente em que o vídeo é observado. Estas imagens independentes, que formam uma sequência de vídeo a um ritmo temporal regular, podem também ser chamadas de tramas, sendo estas formados por *pixels*. *Pixel* é o nome dado aos pontos que formam a trama quando organizada sob a forma de matriz. Como descrito, os vídeos podem ser divididos em sequências de tramas, sendo que estas tramas podem ainda ser divididas em blocos. Os blocos podem apresentar diferentes tamanhos, de acordo com as normas de codificação de vídeo.

A dimensão do vídeo, chamada de resolução, pode variar livremente, embora existam alguns formatos pré-definidos, tais como o CIF (*Common Intermediate Format*), com 352x288 *pixels*, o VGA (*Video Graphics Array*), com 640x480 *pixels*, o 720p com 1280x720 *pixels* (p significa vídeo com varrimento progressivo) e o 1080p, com 1920x1080 *pixels*. Quanto maior for a resolução da trama, maior é o número de *pixels* e, logo, melhor será a definição ou qualidade de imagem e mais pormenorizada é a sua perceção.

Outra característica que define a qualidade de vídeo é a taxa temporal com que as tramas são apresentadas. Quanto maior for a taxa temporal, maior será a suavidade de transição entre as imagens. A taxa temporal também é por vezes designada por taxa de amostragem ou frequência temporal. Para que o sistema visual humano não perceba uma



transição grosseira entre as imagens, a taxa de amostragem, na maioria dos formatos de vídeo digital é de pelo menos 24 tramas por segundo [4].

## **2.2 – Compressão com e sem perdas**

A quantidade de dados envolvidos no processamento de vídeo digital é muito grande, o que torna a utilização deste tipo de vídeos em aplicações de transmissão em tempo real impraticável sem recorrer ao uso de técnicas de compressão. Com a utilização de técnicas de compressão é possível reduzir a quantidade de dados a transmitir, com perdas mínimas na qualidade ou até sem perda alguma.

Tendo em conta as perdas de informação que podem ocorrer durante o processo de compressão, os codificadores de vídeo podem ser classificados em dois tipos, com perdas e sem perdas [4]. Quando a compressão sem perdas é utilizada, é garantido que após a descompressão o vídeo reconstituído é igual ao vídeo original. Apesar dos arquivos sem perdas reduzirem o tamanho dos arquivos, este tipo de compressão não é indicada para a utilização em vídeos digitais, uma vez que as taxas de compressão são reduzidas, devido ao fato de num vídeo digital os valores dos *pixels* apresentarem uma probabilidade de ocorrência muito semelhante.

A compressão com perdas é a mais apropriada a ser usada em vídeos digitais, pois os compressores de vídeos atuais visam explorar o facto de que algumas informações da imagem não são relevantes para a percepção visual humana, podendo ser removidas [5]. Com este tipo de técnica é possível obter taxas de compressão elevadas mantendo a qualidade de imagem sem que, na maioria dos casos, se perceba qualquer alteração na mesma. Uma das normas de vídeo que usa este tipo de técnica é a norma H.264/AVC, que consegue atingir taxas de compressão de algumas dezenas, mantendo uma qualidade de imagem muito próxima ao vídeo original.

## 2.3 – Sistema de Cores e Subamostragem de Cores

A interpretação das cores pelo sistema visual humano está diretamente ligada à representação de vídeos digitais, sendo que existem diversas maneiras das cores serem representadas. A escolha do sistema de cores é muito importante para a eficiência na codificação de vídeos digitais. Os sistemas de cores mais utilizados são o RGB (*Red, Green and Blue*) e o YCbCr (*Luminance, Blue Chrominance and Red Chrominance*). No sistema RGB as cores são formadas a partir de uma combinação de diferentes intensidades das componentes primárias R (*red* – vermelho), G (*green* - verde) e B (*blue* – azul). O sistema de cores YCbCr, bastante utilizado na representação de vídeos digitais, é composto por três componentes, luminância (Y), que define a intensidade luminosa ou o brilho, a cromaância azul (Cb) e a cromaância vermelha (Cr). Devido ao facto de o sistema visual humano ser muito mais sensível a informações de luminância do que a informações de cromaância, a utilização do sistema YCbCr é mais vantajosa, em relação ao sistema RGB, pois a informação da cor está completamente separada da informação do brilho. Deste modo, as informações podem ser tratadas de forma diferenciada pelos codificadores de vídeo digital, algo muito difícil no sistema RGB, pois os tri-estímulos R, G e B possuem um elevado grau de correlação, tornando difícil o processamento de cada uma das informações de cor de forma independente.

Como referido anteriormente, o sistema visual humano é muito mais sensível a informações de luminância do que a informações de cromaância. É então conveniente que a taxa de amostragem espacial dos componentes de cromaância seja reduzida em relação à componente de luminância. Esta operação tem o nome de subamostragem de cores e é utilizada no sistema YCbCr nas normas de vídeo atuais. Existem várias formas de relacionar a amostragem espacial do componente de luminância com a amostragem espacial dos componentes de cromaância, ou seja várias relações de subamostragem, sendo que as configurações mais comuns são o 4:4:4, o 4:2:2 e o 4:2:0, que se encontram representadas na Figura 1. Na configuração 4:4:4 para cada quatro amostras de luminância (Y), existem quatro amostras de cromaância azul (Cb) e quatro amostras de cromaância vermelha (Cr), por isso, os três componentes de cor possuem a mesma resolução e existe uma amostra de cada elemento de cor para cada *pixel* da imagem e, assim, a subamostragem não é aplicada. Na configuração 4:2:2, existem duas amostras de cromaância azul e duas amostras de cromaância vermelha para cada quatro amostras de luminância. Neste caso, as amostras de

crominância possuem a mesma resolução vertical das amostras de luminância, mas possuem metade da resolução horizontal. Já na configuração 4:2:0, para cada quatro amostras de luminância, existe apenas uma amostra de cada tipo de crominância. Neste caso, as amostras de crominância possuem metade da resolução horizontal e metade da resolução vertical do que as amostras de luminância. A configuração deveria ser chamada 4:1:1, para assim os números representarem a relação lógica entre os componentes de cor, mas por motivos históricos esta não é a nomenclatura utilizada [5].

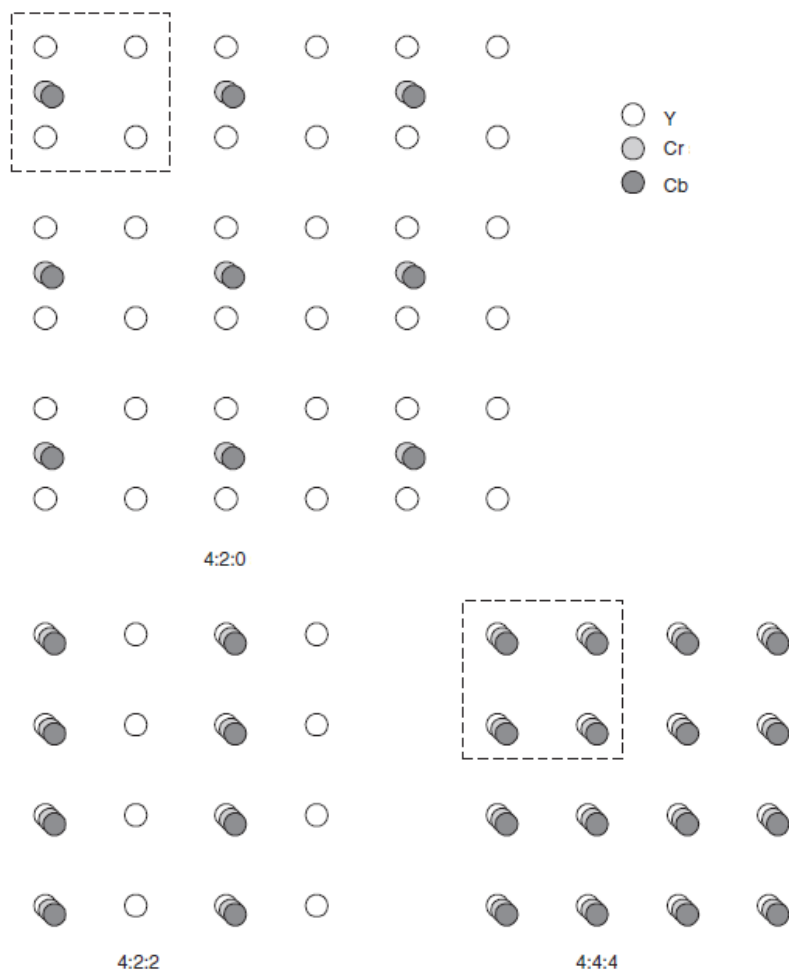


Figura 1- Formatos de subamostragem de crominância (adaptado de [6])

A utilização da subamostragem de cores permite um aumento significativo na eficiência da codificação, uma vez que, partes dos dados são descartados sem que essa perda de informação seja perceptível ao olho humano. Tomando com exemplo a configuração 4:2:0, uma vez que cada componente de crominância possui exatamente um quarto das amostras

presentes na componente de luminância, então um vídeo YCbCr que use a configuração 4:2:0 irá utilizar exatamente a metade das amostras para um vídeo YCbCr que usa a configuração 4:4:4, o que implica uma taxa de compressão de 50% tendo em conta apenas a subamostragem.

## **2.4 – Redundância de Dados**

A compressão tem como objetivo diminuir o número de bits necessários para a representação de imagens e de vídeos, eliminando informações que são redundantes, ou seja, dados que não representam uma nova informação relevante para a representação do vídeo. Assim, é possível também, aumentar a eficiência da codificação de vídeo. Existem três tipos de redundância de dados que podem ser explorados na compressão de vídeos digitais: espacial, temporal e entrópica.

### **2.4.1 – Redundância Espacial**

A redundância espacial, que pode também ter o nome de redundância intra-trama, é resultante da relação entre *pixels* que compõem a mesma imagem. Em sinais de vídeo a correlação pode ser modelada quer no domínio espacial quer no domínio das frequências. Se for percebida no domínio das frequências, é usada a quantização dos coeficientes usando passos de quantização maiores para os coeficientes que correspondem às frequências menos relevantes. Caso a correlação seja percebida no domínio espacial, a redundância é reduzida através de predição intra-trama seguida de codificação do resíduo de predição. Esta forma de redução de redundância é utilizada na maioria das normas de vídeo atuais. Esta correlação é perceptível no domínio espacial e observável quando *pixels* vizinhos numa trama possuem valores de luminância e crominância semelhantes.

## 2.4.2 – Redundância Temporal

A redundância temporal, ou redundância inter-trama, é consequente da semelhança entre tramas temporalmente vizinhas. Considerando duas tramas temporalmente próximas, muitos *pixels* co-localizados mas pertencentes a tramas consecutivas não mudam de valor de trama para trama, como é o caso de fundos de imagens (céu ou parede, por exemplo), ou variam muito pouco, caso haja, por exemplo, uma variação de iluminação. Pode também acontecer que alguns *pixels* reapareçam em outra trama mas deslocados em relação à trama anterior, como, por exemplo, no movimento de um objeto numa cena. A exploração eficiente da redundância temporal leva a elevadas taxas de compressão e é fundamental para o sucesso dos codificadores, sendo explorada em todas as normas de vídeo atuais [7].

## 2.4.3 – Redundância Entrópica

A redundância entrópica está diretamente relacionada com as probabilidades de ocorrência dos símbolos codificados. Quanto maior for a probabilidade de um símbolo ocorrer, menor é a quantidade de informação nova transmitida por este símbolo. Então, os codificadores que exploram este tipo de redundância procuram transmitir o máximo de informação possível por símbolo codificado, para assim transmitir um número menor de bits. A codificação de entropia utiliza diferentes técnicas e algoritmos de compressão sem perdas para atingir este objetivo [8].

## 2.5 – A norma H.264/AVC

A norma H.264/AVC foi desenvolvida pelo JVT (*Join Video Team*), que foi formado a partir da união entre os especialistas do VCEG (*Video Coding Experts Group*) da ITU [2] e do MPEG (*Moving Picture Experts Group*) da ISO/IEC [3]. A primeira versão da norma H.264/AVC foi aprovada em 2003. Foi desenvolvida tendo como principal objetivo, duplicar a taxa de compressão em relação às normas anteriores.

A norma H.264/AVC atingiu com sucesso o objetivo inicialmente proposto, mas para isso foi necessário um grande aumento na complexidade computacional, porque para atingir

tal objetivo foi necessário a introdução de diversos recursos no processo de codificação, tais como, o uso de tamanho de blocos menores que  $8 \times 8$  *pixels*, a predição intra-trama realizada no domínio espacial, as transformadas Hadamard e DCT, a utilização de múltiplas tramas de referência para predição inter-trama e a utilização de um filtro redutor de efeito de bloco.

Os *pixels*, na norma H.264/AVC, são agrupados em regiões denominadas macrobloco, sendo que cada macrobloco é formado por  $16 \times 16$  amostras de luminância associadas a outras duas áreas de crominância, que possuem dimensões dependentes da subamostragem de cores utilizada. Para além dos *pixels*, a informação que representa as tramas codificadas inclui também dados de controlo que indicam o tipo de codificação utilizada, o início de cada macrobloco, o tipo de macrobloco, etc. Cada trama é formada por um ou mais grupos de macroblocos, chamados de *slices*, que podem ser de três tipos diferentes, tipo I, tipo P e tipo B.

*Slices* do tipo I (intra) são *slices* formados por macroblocos que foram codificados utilizando a predição intra-trama. *Slices* do tipo P foram codificados através da predição inter-trama e utilizam apenas uma trama de vídeo como referência. Os *slices* do tipo B podem conter macroblocos do tipo I ou P, são formados através da codificação inter-trama e possuem pelo menos duas tramas como referência. De salientar, que a norma apresenta ainda um tipo de macrobloco denominado SKIP, no qual nenhuma informação é codificada. Este tipo de macrobloco pode ser utilizado em *slices* do tipo P ou do tipo B.

Cada macrobloco da trama em codificação pode ser predito a partir dos macroblocos vizinhos processados anteriormente, neste caso através da predição intra-trama. A predição intra-trama reduz a redundância espacial, pois é realizada através da interpolação de valores dos *pixels* das zonas marginais de blocos espacialmente vizinhos que foram anteriormente codificados. Os cálculos podem ser realizados sobre blocos de luminância  $4 \times 4$  de acordo com nove modos diferentes de predição intra-trama ilustrados na Figura 2. Os modos 0 e 1 fazem uma simples cópia dos *pixels* das arestas verticais ou horizontais para todas as posições do bloco. O modo DC (2) faz uma média entre as amostras das arestas e copia o resultado para todas as posições do bloco. Os restantes modos (3 a 8) fazem uma média ponderada das amostras das arestas, de acordo com a direção das setas da Figura 2. Contudo, os cálculos podem também ser realizados sobre blocos de luminância com tamanho  $16 \times 16$ , existindo neste caso 4 modos possíveis, representados na Figura 3. Os modos 0 e 1 são simples cópias na vertical ou na horizontal das amostras reconstruídas das arestas. O modo 2 é o modo DC e é formado por uma média simples dos elementos das arestas, cujo resultado é copiado para

todas as posições do bloco. O modo 3 aplica uma função linear que considera as amostras horizontais e verticais.

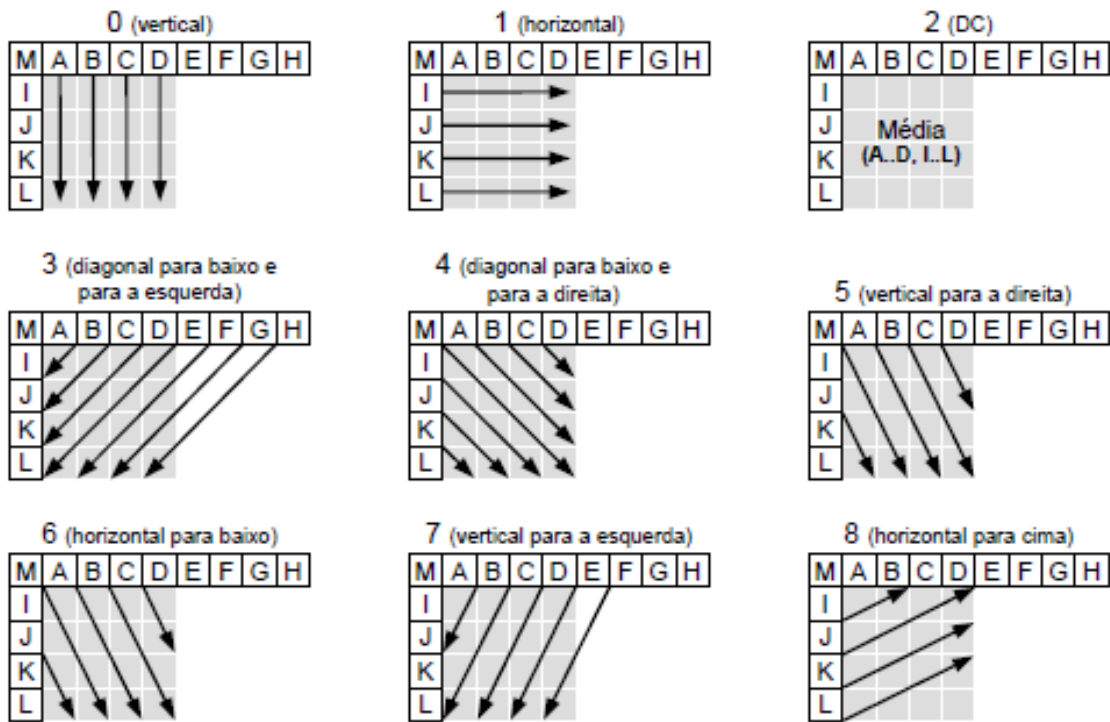


Figura 2 - Modos de predição intra-trama para blocos de 4x4 pixels (adaptado de [6])

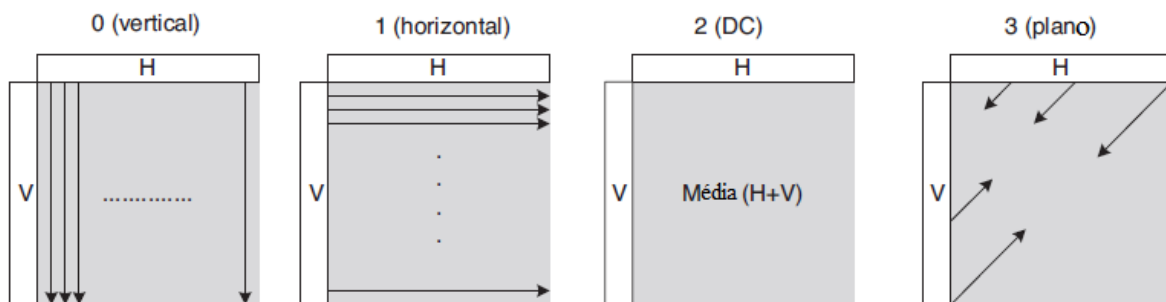


Figura 3 - Modos de predição intra-trama para blocos de 16x16 pixels (adaptado de [6])

Se cada macrobloco da trama em codificação for previsto a partir de um macrobloco contido numa trama de referência processada anteriormente, estamos perante a predição inter-trama. A predição inter-trama é formada pela estimação de movimento (ME) e pela compensação de movimento (MC) e é responsável por reduzir a redundância temporal através da comparação dos blocos da trama em codificação com os blocos existentes na trama de referência. A estimação de movimento (ME), representada na Figura 4, localiza nas tramas de referência, qual o bloco mais semelhante ao bloco atual. Assim que o bloco é encontrado, é gerado um vetor indicando a posição deste bloco na trama de referência. Este vetor é chamado de vetor de movimento. A compensação de movimento (MC) é o processo de compensação para o deslocamento de objetos entre tramas.

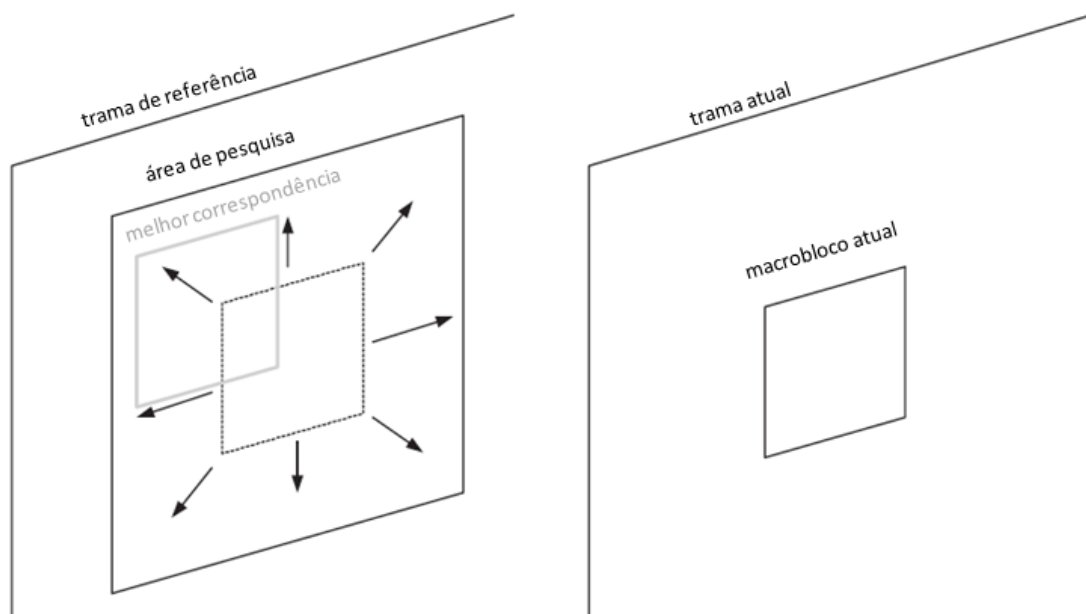


Figura 4 - Estimação de movimento (adaptado de [6])

Após o bloco ser codificado através da predição intra-trama ou inter-trama, é feita uma subtração entre os *pixels* do bloco original e do bloco previsto para se obter a diferença causada pela predição, para ser codificada. A esta diferença dá-se o nome de resíduo. O resíduo é enviado para os módulos de transformada (T) e quantização (Q), como se pode observar na Figura 5. O módulo de transformada tem como objetivo transformar os dados do domínio espacial para o domínio das frequências. O módulo de quantização realiza a



quantização direta e a correção da escala do cálculo das transformadas. Ambos têm como principal objetivo a diminuição da redundância espacial no domínio das frequências. Por fim, a codificação de entropia processa os resultados da quantização e reduz a redundância entrópica. No final da codificação é aplicado um filtro redutor de efeito de bloco para aumentar a qualidade subjetiva da imagem [5].

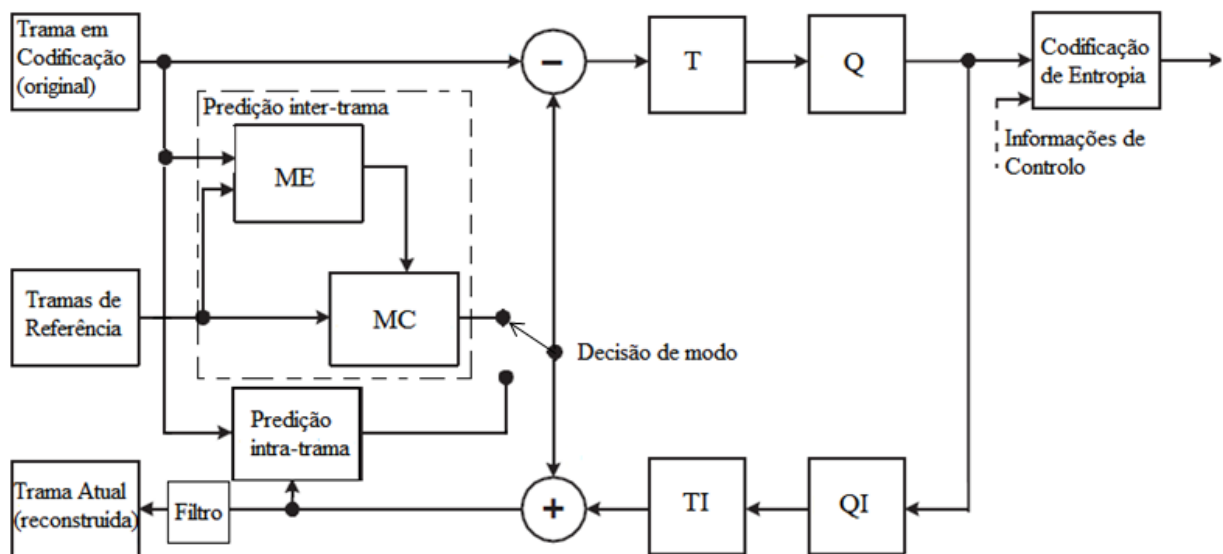


Figura 5 - Diagrama em blocos de um codificador H.264/AVC

A reconstrução, no codificador, é realizada pelos módulos de quantização inversa, transformada inversa e pelo filtro redutor de efeito de bloco. Após estas etapas, os resíduos reconstruídos são obtidos e são somados ao resultado da predição intra-trama ou inter-trama. Durante a reconstrução é utilizado o filtro redutor de efeitos de bloco para reduzir efeitos indesejáveis gerados pela quantização [7].

Um decodificador de vídeo H.264/AVC, ilustrado na Figura 6, tem um funcionamento semelhante ao caminho de reconstrução do codificador. É constituído pelos blocos de predição intra-trama, predição inter-trama, transformada inversa (TI) e quantização inversa (QI), filtro redutor de efeitos de bloco e decodificação de entropia.

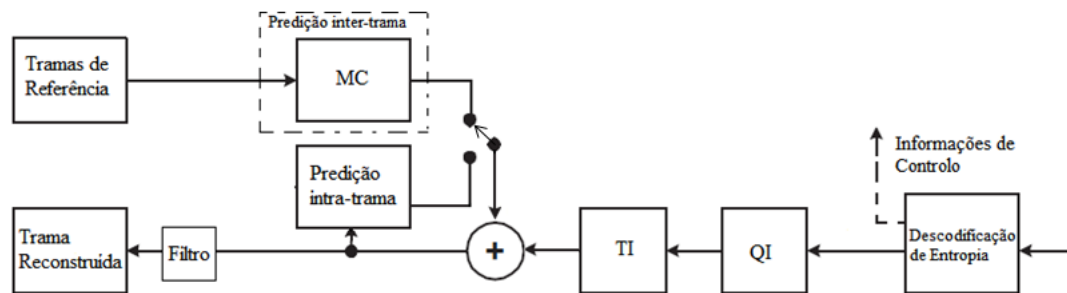


Figura 6 - Diagrama em blocos de um decodificador H.264/AVC

## 2.6 – Transformadas Espacialmente Variáveis

A técnica SVT pressupõe que a posição do bloco transformado dentro de um macrobloco não seja fixa, mas que possa ser variada [9]. Codificar um bloco de resíduo completo não seria favorável, pois o sinal residual pode conter ruído que contribui pouco ou quase nada para a qualidade de vídeo e dificulta a codificação. Com a flexibilidade desta técnica, uma melhoria na eficiência da codificação pode ser alcançada selecionando e codificando a região mais significativa do erro de predição em termos de relação taxa-distorção. Geralmente, isto pode ser feito procurando no bloco de resíduos a região em que os resíduos têm maior magnitude e codificando apenas essa região. Finalmente, os parâmetros de localização (*LP – Location Parameters*) indicam a posição dessa subzona dentro do bloco de resíduos, que é então codificada para o bitstream, se existirem coeficientes transformados diferentes de zero.

A Figura 7 e a Figura 8 mostram como a técnica SVT é implementada no *Tandberg, Ericsson and Nokia Test Model (TENTM)* [10], onde blocos de tamanho 16x4, 4x16 e 8x8 são utilizados nas transformadas. As áreas cinzentas ilustram o bloco SVT codificado. Os valores dos resíduos fora das áreas cinzentas são convertidos para zero.

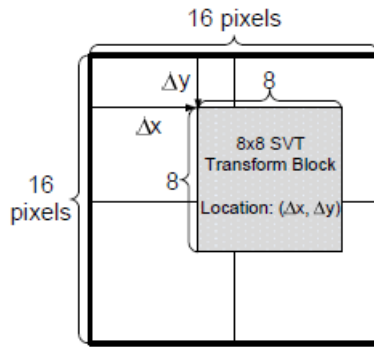


Figura 7 - SVT utilizando a transformada 8x8 [9]

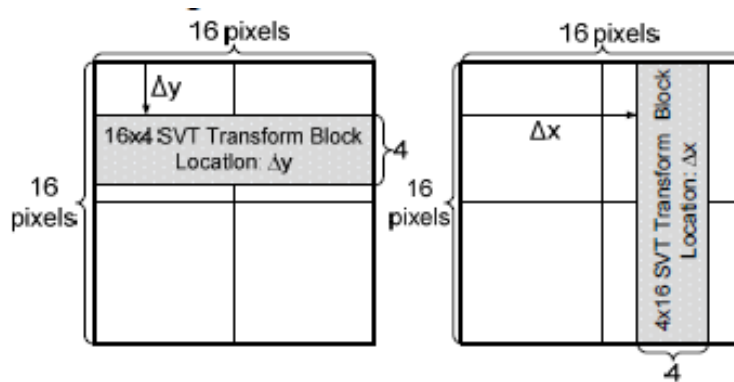


Figura 8 - SVT utilizando as transformadas 16x4 e 4x16 [11]

### 2.6.1 – Parâmetros de localização

Quando existem coeficientes não nulos no bloco selecionado (8x8, 16x4 e 4x16), a sua localização dentro do macrobloco e o seu tipo (8x8, 16x4 ou 4x16) necessitam ser codificados e transmitidos para o decodificador. Como ilustrado na Figura 7 a localização do bloco 8x8 selecionado dentro do macrobloco é denotado como  $(\Delta x, \Delta y)$ , onde  $\Delta x$  e  $\Delta y$  podem tomar valores entre 0 e 8. Existem no total 81 combinações possíveis e é necessário escolher as melhores, de acordo com um certo critério. O critério usado é denominado de Otimização Taxa-Distorção (*RDO – Rate-Distortion Optimization*) e é usado para selecionar o melhor  $(\Delta x, \Delta y)$  em termos de relação taxa-distorção, minimizando  $J$  na equação (1),

$$J = D + \lambda \cdot R \quad (1)$$

onde  $J$  é o custo RD (*Rate-Distortion*) da combinação selecionada,  $D$  é a distorção,  $R$  é a taxa de bits e  $\lambda$  é o multiplicador de Lagrange que nos codificadores normalizados é função do passo de quantização. Os resíduos restantes do macrobloco são forçados a zero.

A seleção de candidatos dos parâmetros de localização é importante, uma vez que afeta diretamente a complexidade de codificação e a performance da técnica SVT. Em [9] foi estudada a frequência da distribuição de  $(\Delta x, \Delta y)$ , concluindo-se que o  $(\Delta x, \Delta y)$  escolhido mais frequentemente são  $(0..8,0)$ ,  $(0..8,8)$ ,  $(0,1..7)$ ,  $(8,1..7)$ , ou seja, 32 combinações das 81 combinações possíveis.

A Figura 8 mostra a localização dos blocos  $16 \times 4$  e  $4 \times 16$ , dentro do macrobloco. Assim, considerando os dois blocos como um todo, os seus parâmetros de localização podem ser representados como  $(shape, \Delta y/\Delta x)$  que pode ser selecionado do conjunto  $\mu = \{(0..1, 0..12)\}$ , onde  $shape=0$  significa que o bloco  $16 \times 4$  está selecionado e  $shape=1$  significa que o bloco  $4 \times 16$  está selecionado. Neste caso, existem no total 26 parâmetros de localização candidatos [11].



## Capítulo 3: Trabalho Desenvolvido

Este capítulo apresenta uma descrição cuidada de um melhoramento introduzido num procedimento de codificação de resíduos de predição de vídeo descrito originalmente em [9], assim como testes detalhados e experiências algorítmicas da solução melhorada proposta. O objetivo primário quer do método original quer da extensão aqui apresentada é a redução da complexidade computacional e a melhoria na eficiência de codificação de vídeo recorrendo ao conceito de transformação espacialmente variável.

Primeiramente são esclarecidas as diversas modificações ao método original e também as alterações efetuadas ao *software* TENTM, proposto pelas empresas *Tandberg*, *Ericsson* e *Nokia* [10], para incorporação do melhoramento proposto. Em seguida são apresentados os diferentes testes realizados, bem como os respetivos resultados.

### 3.1 Introdução de novos formatos de Bloco SVT

O formato do bloco transformado, na técnica SVT (*Spatially Varying Transforms*), pode variar entre 8x8, 4x16 e 16x4. Grande parte do trabalho aqui desenvolvido passa por aumentar este número de opções de escolha do formato do bloco transformado. Para além dos formatos já existentes foram adicionados os formatos: 4x4, 8x4, 4x8, 16x8 e 8x16. A Figura 9 mostra a posição de alguns dos novos blocos dentro do macrobloco. Uma vez que o erro de predição existente nos macroblocos não possui sempre as mesmas características, ao aumentar o número de blocos para a sua codificação espera-se obter um aumento da eficiência de codificação, pois a probabilidade do erro de predição ser melhor localizado é maior.

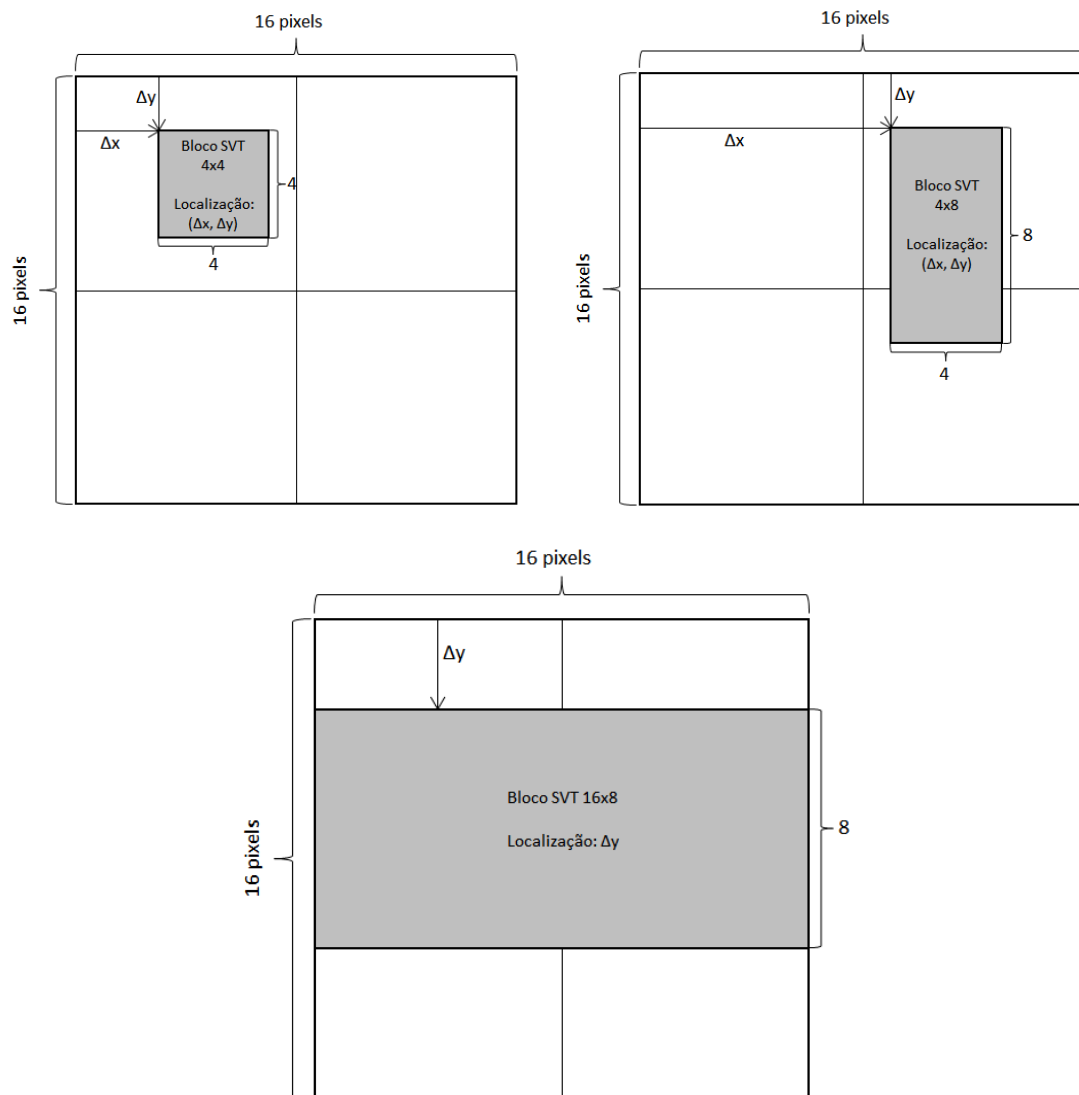


Figura 9 - SVT utilizando as transformadas 4x4, 4x8 e 16x8

### 3.1.1 Seleção dos Parâmetros de Localização

Na escolha dos parâmetros de localização dos novos formatos (4x4, 8x4, 4x8, 16x8 e 8x16), não foi feito um estudo tão aprofundado como o referido em [9], mas a sua seleção foi feita seguindo uma relação, idêntica à anterior, entre o formato do bloco e a sua posição dentro do macrobloco. Na Tabela 1 encontram-se os valores dos parâmetros de localização escolhidos para os novos formatos de blocos SVT introduzidos no *software* TENTM [10].

Tabela 1 - Parâmetros de Localização para os novos formatos de blocos SVT

Formatos de blocos SVT	Parâmetros de localização ( $\Delta x$ , $\Delta y$ )
4x4	(0...12,0), (0...12,4), (0...12,8), (0...12,12), (0,1...11), (4,1...11), (8,1...11), (12,1...11)
8x4	(0...8,0), (0...8,4), (0...8,8), (0...8,12), (0,1...11), (8,1...11)
4x8	(0...12,0), (0...12,8), (0,1...7), (4,1...7), (8,1...7), (12,1...7)
16x8	$\Delta y = (0...8)$
8x16	$\Delta x = (0...8)$

### 3.2 Decisão de Posicionamento e Formato de Bloco SVT

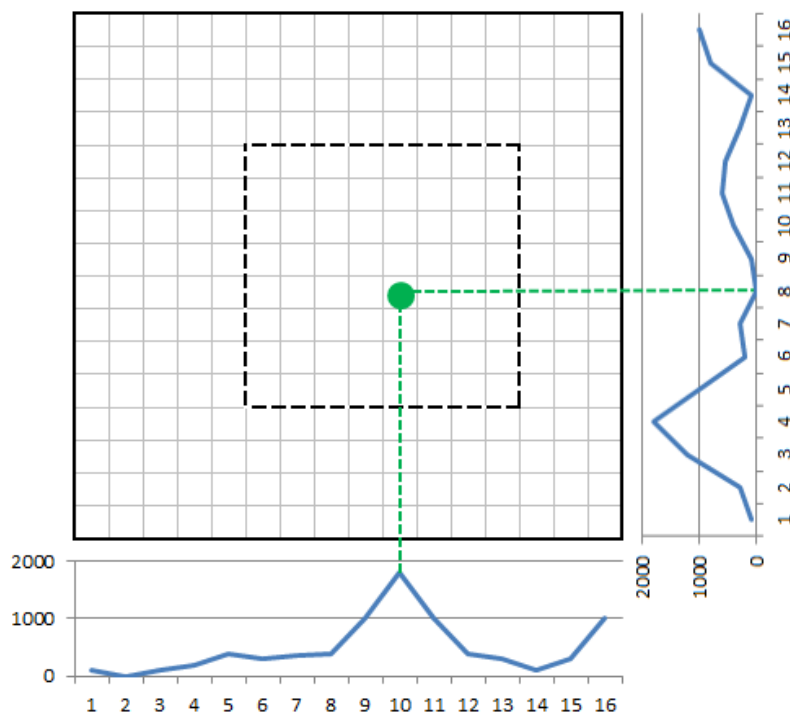
Duas ideias foram exploradas para redução de complexidade envolvida na decisão do melhor posicionamento do bloco transformado, segundo a técnica SVT. Ambas as ideias aqui representadas são baseadas no cálculo da intensidade do resíduo existente no macrobloco a ser transformado, contudo, variam em termos de nível de precisão. Sabendo que os valores residuais de áreas vizinhas são bastantes próximos (redundância espacial), as duas ideias procuram posicionar o bloco a ser transformado em áreas cujos valores dos resíduos (*SAD - Sum of Absolute Differences*) são mais elevados. A SAD é uma medida de distorção utilizada em codificadores de vídeo. A função SAD é calculada conforme mostrado na equação 2, onde  $m$  e  $n$  são as dimensões da imagem em *pixels* e o  $O$  e  $R$  representam as componentes de luminância da imagem original e da imagem reconstruída, respetivamente.

$$SAD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |R_{i,j} - O_{i,j}| \quad (2)$$

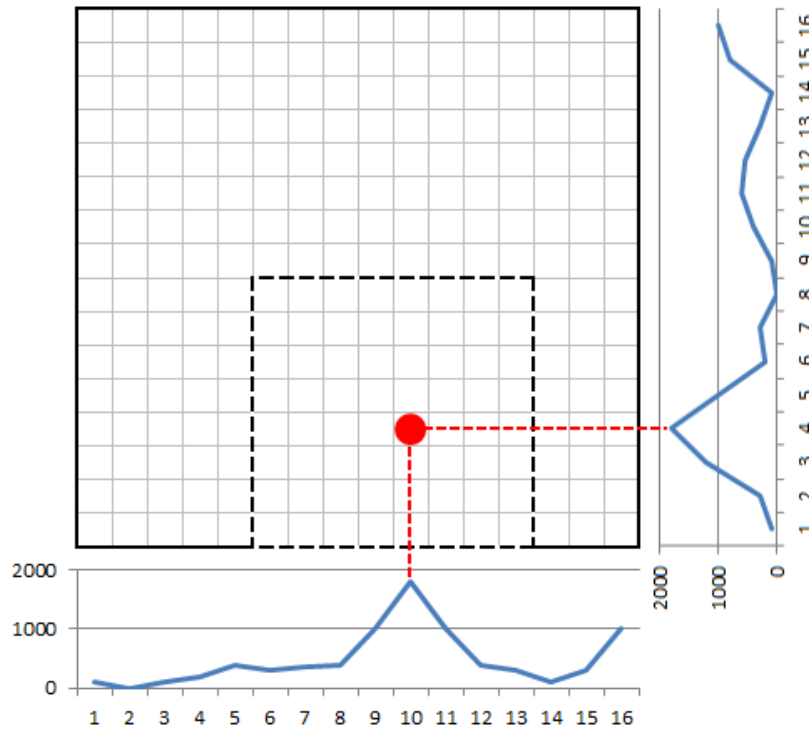


A Figura 10 e a Figura 11 ilustram as ideias exploradas, utilizando como exemplo o posicionamento de um bloco transformado 8x8 dentro de um macrobloco (16x16). Os gráficos abaixo e à direita do macrobloco a ser transformado representam os somatórios dos valores residuais das colunas e das linhas, respetivamente.

O ponto de intersecção verde é obtido através do cálculo do centro de massa da SAD das linhas e das colunas. Enquanto que o ponto de intersecção vermelho é obtido seleccionando a coluna e a linha com a maior SAD. Cada um destes pontos indica a posição central da área onde possivelmente estarão concentrados os maiores valores residuais. Desta forma, o bloco 8x8 a ser transformado pode ser centrado no ponto encontrado, enquanto os valores residuais do restante macrobloco podem ser convertidos para zero.

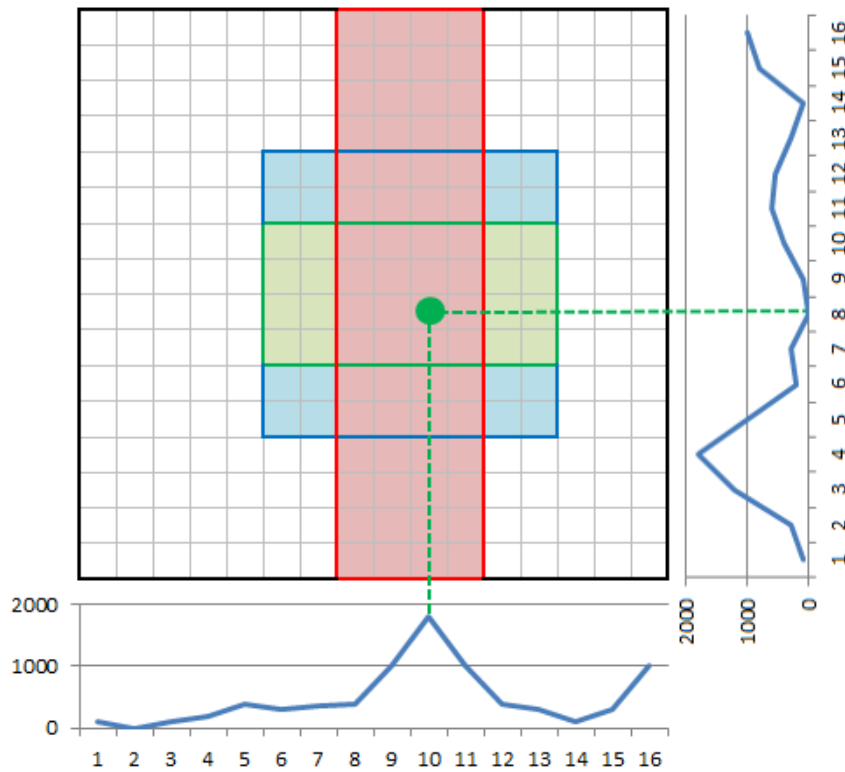


**Figura 10 - Posicionamento do bloco SVT obtido através do cálculo do centro de massa da SAD das linhas e das colunas**



**Figura 11 - Posicionamento do bloco SVT obtido através da linha e coluna com maior SAD**

A decisão de formato e posicionamento de bloco SVT é obtida de forma igual, independentemente da ideia utilizada. A coluna obtida fornece a posição horizontal central do bloco 4x16 e 8x16 (quer seja através do cálculo do centro de massa ou a coluna que tenha a maior SAD), e a linha obtida fornece a posição vertical central do bloco 16x4 e 16x8 (quer seja através do cálculo do centro de massa ou a linha que tenha a maior SAD). Por fim, a interseção entre a coluna e a linha obtidas fornece a posição dos blocos 8x8, 4x4, 4x8 e 8x4. O bloco que apresentar a maior SAD é utilizado na técnica SVT. A Figura 12 ilustra um exemplo de blocos 4x16 (vermelho), 8x8 (azul) e 8x4 (verde), posicionados de acordo com os resultados obtidos através do cálculo do centro de massa da SAD das linhas e das colunas.



**Figura 12 - Posicionamento dos blocos 4x16 (vermelho), 8x8 (azul) e 8x4 (verde) obtido através do centro de massa**

### 3.3 Estrutura da função SVT no código TENTM

No código TENTM [10] a codificação SVT é realizada pela função *encode\_luma\_SVT\_P()*. A função é composta, entre outras coisas, por um ciclo que varia sobre todos os índices de um macrobloco, um pequeno excerto com a parte inicial do ciclo referido encontra-se representado na Figura 13. Posições de *pixels* fora da zona delimitada pelo macrobloco são excluídas. Dentro do ciclo, são realizados os cálculos de transformada e quantização para todos os blocos 8x8 possíveis dentro do macrobloco, além da quantização inversa, transformadas inversas e filtragens. São também realizados os mesmos cálculos para todos os blocos 16x4 e 4x16 possíveis dentro do macrobloco. No final do ciclo, o custo taxa-distorção de cada bloco SVT candidato é comparado com o melhor custo taxa-distorção até ao momento. Caso o custo do macrobloco em questão seja menor, os seus dados são salvos e o

melhor bloco é atualizado para as comparações do próximo passo do ciclo. Quando o ciclo termina a sua execução, o posicionamento e o formato do melhor bloco SVT é conhecido.

```

.....for (svt_idx=0; svt_idx<SVT_TABLE_SIZE; svt_idx++)
.....{
.....    if (svt_idx<32)
.....        svt_type=0;
.....    else
.....        svt_type=1;
.....    if (svt_type==0)
.....        {
.....        beginy=svt_pos[svt_idx][1];
.....        beginx=svt_pos[svt_idx][0];
.....        dir=0;
.....        offset=0;
.....        }
.....    else
.....        {
.....        beginy=0;
.....        beginx=0;
.....        dir=svt_pos[svt_idx][0];
.....        offset=svt_pos[svt_idx][1];
.....        }

.....    if (skip_svt_idx[svt_idx]==1)
.....        {
.....        svt_idx_num--;
.....        avail_svt_idx[svt_idx]=0;
.....        continue;
.....        }
.....    else if (sad_skip_svt_idx[svt_idx]==1)
.....        continue;

.....    coded_sum=0;
.....    if (svt_type==0)
.....        {
.....        for (y=0; y<8; y++)
.....        {
.....        for (x=0; x<8; x++)
.....        {
.....        dblock8[y][x]=dblock16[y+beginy][x+beginx];
.....        }
.....        }
.....        transform_8x8(dblock8,macroblockInfo->ycoeff[0]);

```

Figura 13 – Pequeno excerto da função *encode\_luma\_SVT\_P()* no código TENTM [10]

### 3.4 Alterações realizadas no código TENTM

As alterações realizadas no código TENTM [10] concentram-se na eliminação do ciclo descrito anteriormente (aplicando as ideias de decisão de formato e posicionamento de bloco SVT) e na introdução dos novos formatos de bloco (4x4, 8x4, 4x8, 16x8 e 8x16).

### **3.4.1 – Alterações realizadas para a decisão de formato e posicionamento de bloco SVT**

Imediatamente antes do ciclo removido, os valores residuais das linhas e das colunas foram somados numa estrutura vetorial. Os valores residuais foram somados e armazenados em dois vetores diferentes, um para as linhas e outro para as colunas. Os maiores valores são salvos, caso a decisão seja através da linha e coluna com maior SAD. Caso a decisão seja feita pelo cálculo do centro de massa, é calculado o centro de massa do vetor que possui os valores residuais das linhas e do vetor que possui os valores residuais das colunas, salvando os resultados.

Os valores salvos indicam assim a posição horizontal e a posição vertical onde irão estar posicionados os blocos transformados. A SAD de todos os blocos é calculado e o maior valor define a posição e o formato do bloco SVT.

### **3.4.2 – Alterações realizadas para introdução de novos formatos de bloco SVT**

Na introdução de novos formatos foi necessário criar funções idênticas às já existentes mas adaptadas aos novos formatos de bloco, como por exemplo funções de transformada, transformada inversa, quantização, quantização inversa e de codificação de coeficientes.

O número de valores de parâmetros de localização também aumentou bastante, o que levou à alteração da tabela de índices SVT já existente no código TENTM [10]. Cada índice corresponde a uma posição possível do bloco transformado dentro do macrobloco. Inicialmente existiam 58 índices (32 para o bloco 8x8, 13 para o bloco 16x4 e 13 para o bloco 4x16), com a adição dos novos blocos, a tabela de índices SVT ficou com 284 índices.

Na técnica SVT está presente uma variável denominada SKIP que de acordo com a posição do bloco transformado dentro do macrobloco e com os vetores de movimento permite concluir se a técnica SVT é útil, caso não seja, não é utilizada. Neste sentido foram introduzidas alterações para que de acordo com os novos blocos seja possível concluir a utilidade da técnica SVT.

Na técnica SVT é usada filtragem entre as arestas do bloco de valores residuais transformado e os restantes valores do macrobloco. Novamente foram feitas as alterações adaptadas aos novos formatos de bloco.

Para além das alterações supracitadas, outras foram também efetuadas. Apenas estas são referenciadas, pois são as que considero ser mais relevantes e que julgo serem fundamentais para uma perceção global.

### 3.5 Resultados Experimentais

Uma grande variedade de testes foi realizada para definir o desempenho das propostas apresentadas neste trabalho. Foram utilizadas dez sequências diferentes, cada uma delas testada com quatro QPs (*Quantization Parameter*) distintos (22, 27, 32 e 37). As características de cada uma das sequências utilizadas encontram-se representadas na Tabela 2. Os testes foram realizados a partir do *software* TENTM [10].

Foram testadas 8 configurações diferentes, para cada uma das dez sequências:

1. SVT1: Decisão de posicionamento feito através da escolha da linha e coluna com maior SAD. Formatos de bloco SVT utilizados: 8x8, 16x4, 4x16, 4x4, 16x8, 8x16, 8x4 e 4x8.
2. SVT2: Decisão de posicionamento feito através da escolha da linha e coluna obtido pelo cálculo do centro de massa da SAD das linhas e colunas. Formatos de bloco SVT utilizados: 8x8, 16x4, 4x16, 4x4, 16x8, 8x16, 8x4 e 4x8.
3. SVT3: Decisão de posicionamento feito através da escolha da linha e coluna com maior SAD. Formatos de bloco SVT utilizados: 8x8, 16x4, 4x16, 4x4.
4. SVT4: Decisão de posicionamento feito através da escolha da linha e coluna obtido pelo cálculo do centro de massa da SAD das linhas e colunas. Formatos de bloco SVT utilizados: 8x8, 16x4, 4x16, 4x4.
5. SVT5: Decisão de posicionamento feito através da escolha da linha e coluna com maior SAD. Formatos de bloco SVT utilizados: 8x8, 16x4, 4x16, 16x8, 8x16.
6. SVT6: Decisão de posicionamento feito através da escolha da linha e coluna obtido pelo cálculo do centro de massa da SAD das linhas e colunas. Formatos de bloco SVT utilizados: 8x8, 16x4, 4x16, 16x8, 8x16.
7. SVT7: Decisão de posicionamento feito através da escolha da linha e coluna com maior SAD. Formatos de bloco SVT utilizados: 8x8, 16x4, 4x16, 8x4 e 4x8.

8. SVT8: Decisão de posicionamento feito através da escolha da linha e coluna obtido pelo cálculo do centro de massa da SAD das linhas e colunas. Formatos de bloco SVT utilizados: 8x8, 16x4, 4x16, 8x4 e 4x8.

**Tabela 2 - Características das sequências usadas**

<b>Sequência</b>	<b>Tramas</b>	<b>Resolução</b>	<b>Taxa de Tramas (Hz)</b>
<b>BasketBallPass</b>	500	416x240	50
<b>BQSquare</b>	600	416x240	60
<b>BasketBallDrill</b>	500	832x480	50
<b>BQMall</b>	600	832x480	60
<b>Mobcal</b>	504	1280x720	50
<b>Parkrun</b>	504	1280x720	50
<b>Shields</b>	504	1280x720	50
<b>BQTerrace</b>	600	1920x1080	60
<b>Cactus</b>	500	1920x1080	50
<b>Traffic</b>	150	2560x1600	30

Todas as configurações apresentadas anteriormente possuem diferenças, seja na decisão de posicionamento do bloco transformado, seja na variedade de formato dos blocos transformados utilizados.

De forma a testar a eficácia das alterações efetuadas, irá ser usada como base de comparação a técnica SVT original [11], que usa como formatos de bloco transformados os formatos 8x8, 16x4 e 4x16 *pixels* e usa a técnica de otimização taxa-distorção na decisão do melhor posicionamento e formato do bloco transformado.

Deste modo irá ser possível averiguar se as alterações efetuadas apresentam ou não melhorias relativamente à técnica SVT original [11], quer seja a nível de redução da taxa de bits, quer seja a nível de redução de tempo de codificação.

O cálculo da redução da taxa de bits foi obtido de acordo com [12] para todas as oito configurações descritas. O método utilizado é denominado Bjontegaard Delta e permite calcular o ganho relativo entre duas simulações, medindo a diferença média (diferença entre os integrais dividida pelo intervalo de integração) entre duas curvas R-D, formadas a partir dos valores de PSNR (*Peak signal-to-noise ratio*) e de taxa de bits para QPs iguais a 22, 27, 32 e 37.

A Figura 14 representa a aplicação do método Bjontegaard Delta a um conjunto de dados. O logaritmo da taxa de bits é usado de forma a conseguir um melhor equilíbrio entre taxas de bits altas e taxas de bits baixas no cálculo final das alterações da taxa de bits e PSNR.

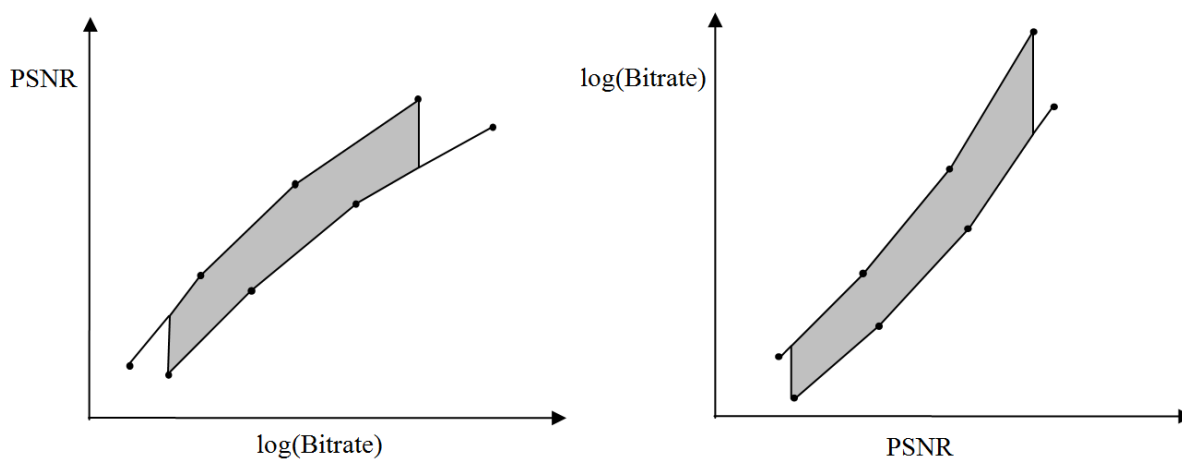


Figura 14 - Aplicação do método Bjontegaard Delta

Para além de permitir calcular a diferença média da taxa de bits em percentagem em toda a gama de valores de PSNR, o método Bjontegaard Delta permite também calcular a diferença média de PSNR em dB (decibel) em toda a gama de valores de taxa de bits.



Através da análise da Tabela 3, é fácil notar que independentemente da configuração utilizada houve sempre uma redução da taxa de bits comparativamente à técnica SVT original [11] e que a configuração com melhores resultados é a SVT2, que obtém uma média de 0,52% de redução da taxa de bits. Ainda assim a diferença para as restantes configurações não é muito grande. É também possível concluir que os melhores resultados, sempre que os formatos de bloco não diferem entre as configurações, são obtidos quando a decisão de posicionamento do bloco transformado é adquirido através do cálculo do centro de massa da SAD das linhas e das colunas.

**Tabela 3 - Redução da Taxa de bits**

Sequência	Taxa de bits							
	SVT1	SVT2	SVT3	SVT4	SVT5	SVT6	SVT7	SVT8
<b>BasketBallPass</b>	-0,54%	-0,69%	-0,39%	-0,66%	-0,32%	-0,67%	-0,45%	-0,59%
<b>BQSquare</b>	-0,57%	-0,76%	-0,75%	-0,65%	-0,73%	-0,82%	-0,64%	-0,74%
<b>BasketBallDrill</b>	-0,33%	-0,21%	-0,30%	-0,07%	-0,26%	-0,18%	-0,19%	-0,16%
<b>BQMall</b>	-0,53%	-0,47%	-0,22%	-0,48%	-0,31%	-0,49%	-0,55%	-0,37%
<b>Mobcal</b>	-0,36%	-0,41%	-0,20%	-0,40%	-0,23%	-0,30%	-0,20%	-0,41%
<b>Parkrun</b>	-0,77%	-0,60%	-0,30%	-0,48%	-0,31%	-0,37%	-0,31%	-0,34%
<b>Shields</b>	-0,52%	-0,67%	-0,58%	-0,60%	-0,70%	-0,62%	-0,68%	-0,65%
<b>BQTerrace</b>	-0,72%	-0,87%	-0,64%	-1,09%	-0,56%	-0,65%	-0,68%	-0,75%
<b>Cactus</b>	-0,17%	-0,16%	-0,14%	-0,23%	-0,15%	-0,30%	-0,16%	-0,31%
<b>Traffic</b>	-0,52%	-0,31%	-0,34%	-0,40%	-0,27%	-0,43%	-0,26%	-0,37%
<b>Média</b>	-0,50%	-0,52%	-0,39%	-0,51%	-0,38%	-0,48%	-0,41%	-0,47%

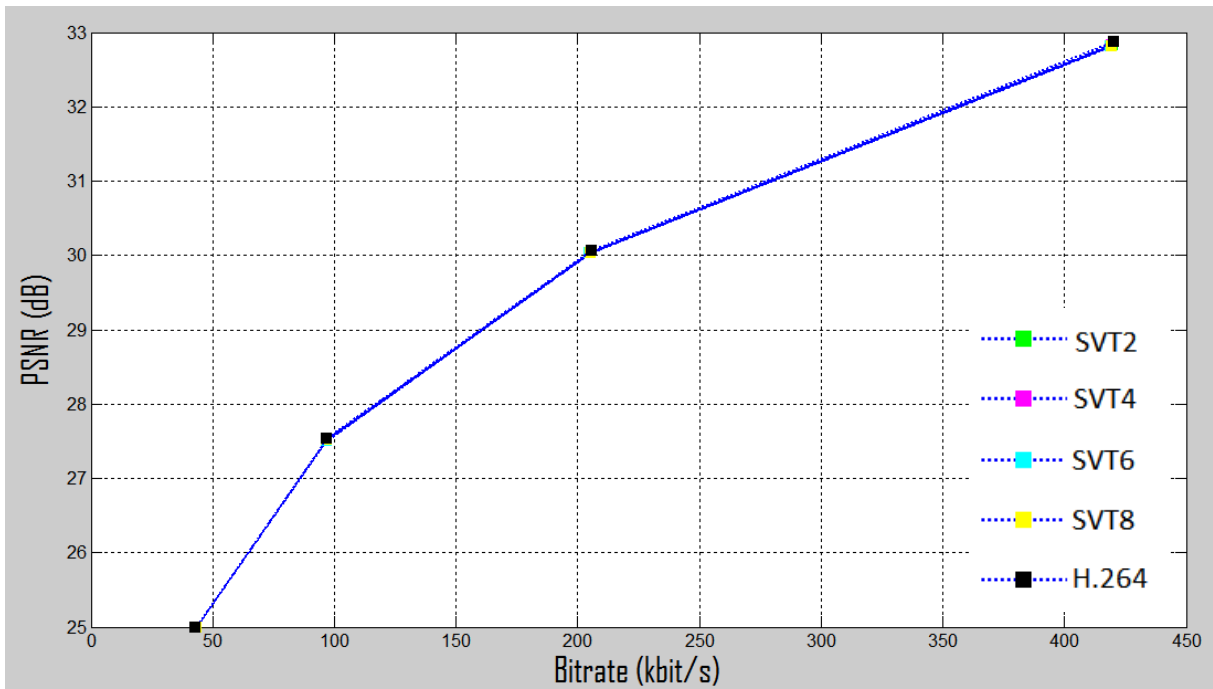


Figura 15 - Curva R-D da sequência BasketballPass

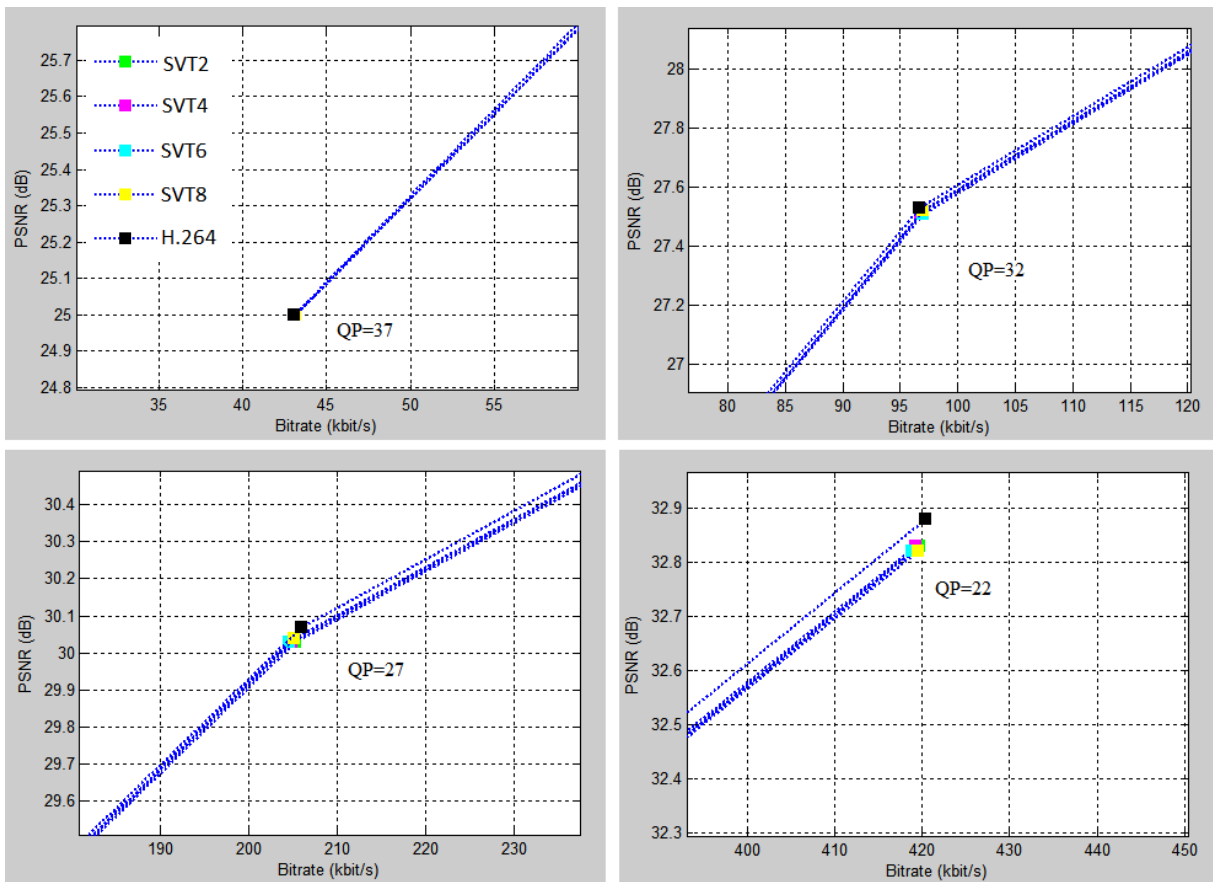


Figura 16 - QPs em destaque da Curva R-D da sequência BasketballPass

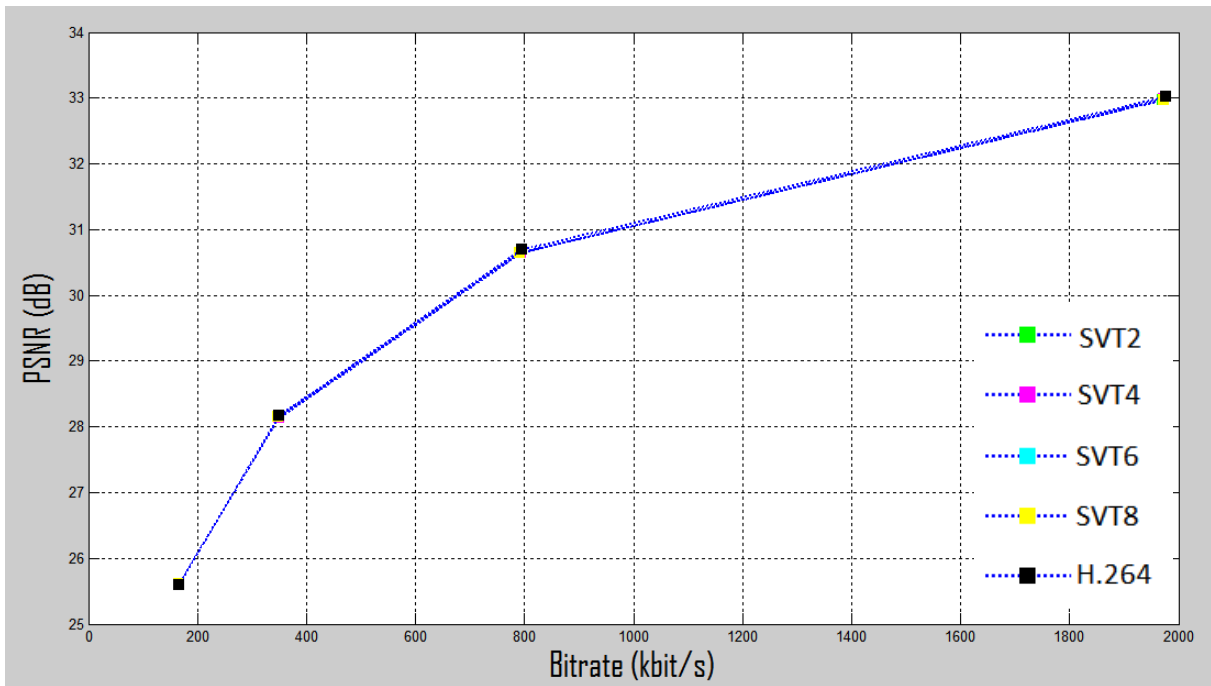


Figura 17 - Curva R-D da seqüência BQTerrace

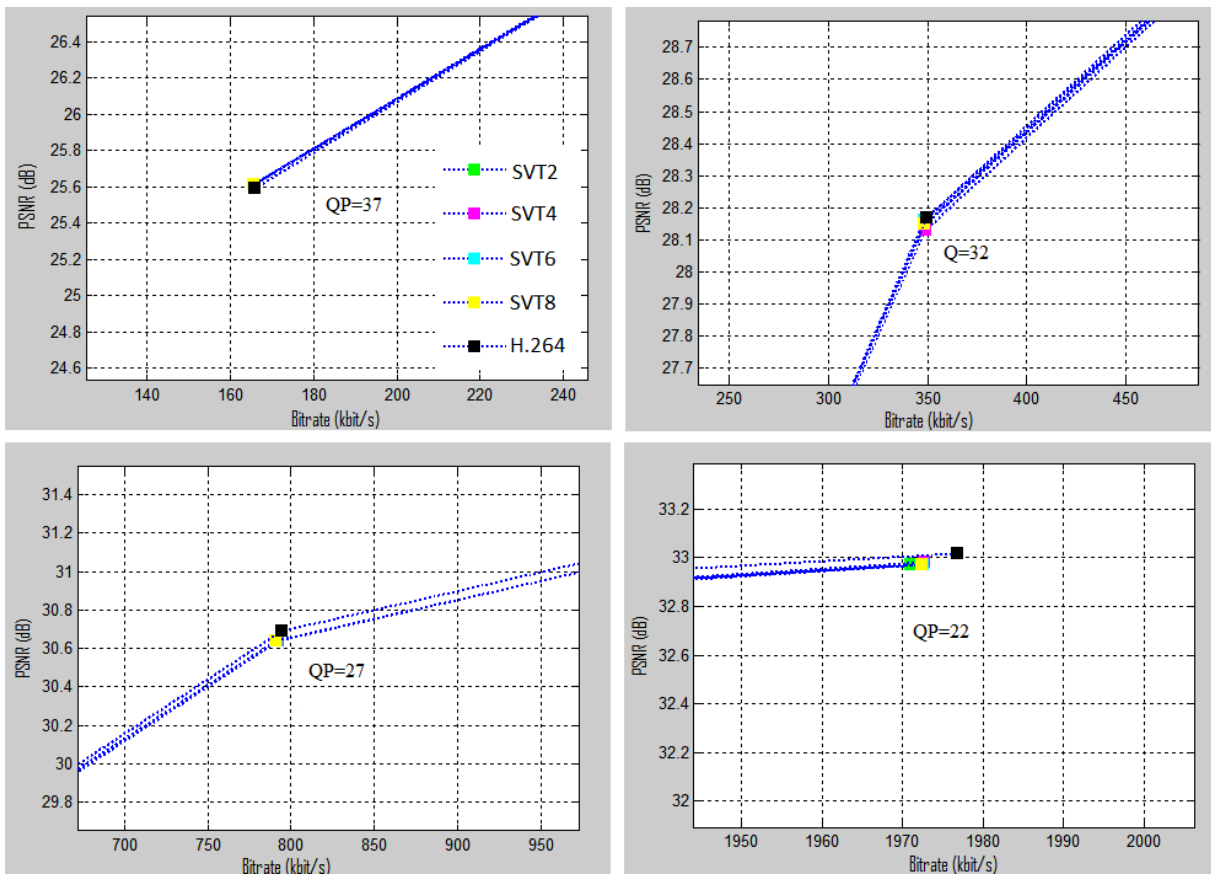


Figura 18 - QPs em destaque da Curva R-D da seqüência BQTerrace

A Figura 15 e a Figura 17 mostram as curvas R-D para as sequências BasketballPass e BQTerrace, respetivamente, usando apenas as configurações em que a decisão de posicionamento do bloco transformado é adquirido através do cálculo do centro de massa da SAD das linhas e das colunas, que como vimos anteriormente é a mais benéfica. Devido ao facto de os valores serem muito próximos, é necessário ampliar e ver individualmente as soluções para os diferentes QPs. Como se pode ver através da Figura 16 e Figura 18, que mostram a curva R-D com maior detalhe, a redução da taxa de bits é mais significativa para taxas de bits altas. Isto acontece para a maioria das sequências testadas neste trabalho.

**Tabela 4 - Redução do tempo de codificação (QP=22)**

Sequência	Tempo de Codificação / QP = 22							
	SVT1	SVT2	SVT3	SVT4	SVT5	SVT6	SVT7	SVT8
<b>BasketBallPass</b>	-5,50%	-6,03%	-3,35%	-9,84%	-9,17%	-9,28%	-4,62%	-7,02%
<b>BQSquare</b>	-11,14%	-11,55%	-7,34%	-11,76%	-6,06%	-6,24%	-7,79%	-11,22%
<b>BasketBallDrill</b>	-6,02%	-5,46%	-3,53%	-8,03%	-5,94%	-6,11%	-2,78%	-5,65%
<b>BQMall</b>	-6,78%	-6,13%	-2,38%	-8,53%	-6,15%	-5,26%	-3,97%	-7,64%
<b>Mobcal</b>	-5,00%	-5,87%	+0,08%	-6,73%	-4,69%	-5,98%	+0,07%	-5,90%
<b>Parkrun</b>	-6,68%	-6,04%	-2,65%	-7,24%	-5,52%	-6,42%	-0,92%	-6,37%
<b>Shields</b>	-8,06%	-7,05%	-3,90%	-8,82%	-8,01%	-7,95%	-3,21%	-7,69%
<b>BQTerrace</b>	-6,09%	-5,68%	-2,03%	-6,47%	-5,64%	-4,94%	+0,13%	-5,81%
<b>Cactus</b>	-5,92%	-6,05%	-2,62%	-6,30%	-6,34%	-6,10%	-1,97%	-6,04%
<b>Traffic</b>	-6,57%	-6,34%	-2,68%	-7,05%	-2,09%	-3,67%	-2,58%	-2,05%
<b>Média</b>	-6,78%	-6,62%	-3,04%	-8,08%	-5,96%	-6,20%	-2,76%	-6,54%

**Tabela 5 - Redução do tempo de codificação (QP=27)**

Sequência	Tempo de Codificação / QP = 27							
	SVT1	SVT2	SVT3	SVT4	SVT5	SVT6	SVT7	SVT8
<b>BasketBallPass</b>	-5,41%	-8,93%	-3,03%	-9,58%	-9,55%	-7,42%	-2,50%	-6,87%
<b>BQSquare</b>	-5,74%	-4,77%	-0,60%	-4,28%	-4,32%	-4,84%	-0,65%	-5,46%
<b>BasketBallDrill</b>	-5,46%	-5,34%	-3,03%	-7,35%	-4,11%	-5,19%	-2,14%	-7,06%
<b>BQMall</b>	-7,86%	-4,89%	-0,77%	-4,84%	-5,12%	-4,03%	+0,66%	-5,97%
<b>Mobcal</b>	-4,67%	-6,28%	-0,80%	-6,30%	-5,29%	-6,24%	+0,60%	-5,83%
<b>Parkrun</b>	-5,59%	-6,08%	-1,25%	-6,65%	-5,45%	-6,20%	-0,03%	-6,14%
<b>Shields</b>	-5,51%	-5,26%	-1,15%	-5,79%	-5,47%	-5,65%	-0,21%	-5,38%
<b>BQTerrace</b>	-3,87%	-5,00%	-0,17%	-4,69%	-4,09%	-3,03%	+2,00%	-4,50%
<b>Cactus</b>	-5,26%	-5,55%	-1,76%	-5,54%	-5,74%	-5,71%	-1,07%	-5,55%
<b>Traffic</b>	-5,09%	-5,28%	-1,02%	-5,56%	-0,15%	-0,25%	-0,68%	-4,81%
<b>Média</b>	-5,45%	-5,74%	-1,36%	-6,06%	-4,93%	-4,86%	-0,40%	-5,76%

**Tabela 6 - Redução do tempo de codificação (QP=32)**

Sequência	Tempo de Codificação / QP = 32							
	SVT1	SVT2	SVT3	SVT4	SVT5	SVT6	SVT7	SVT8
<b>BasketBallPass</b>	-5,56%	-4,99%	+2,18%	-5,56%	-5,44%	-5,65%	+1,53%	-4,15%
<b>BQSquare</b>	-6,04%	-5,37%	+7,01%	-6,10%	-4,77%	-4,35%	-1,26%	-5,51%
<b>BasketBallDrill</b>	-5,26%	-5,34%	-1,81%	-7,55%	-6,98%	-6,88%	-1,98%	-6,57%
<b>BQMall</b>	-6,61%	-4,43%	+0,23%	-7,12%	-3,22%	-4,29%	-1,70%	-2,59%
<b>Mobcal</b>	-4,50%	-6,33%	-0,58%	-6,41%	-4,39%	-6,36%	+0,69%	-6,16%
<b>Parkrun</b>	-6,28%	-7,60%	-1,31%	-7,39%	-6,13%	-6,87%	-0,13%	-7,04%
<b>Shields</b>	-6,32%	-5,82%	-1,54%	-6,09%	-5,53%	-6,04%	-0,89%	-5,88%
<b>BQTerrace</b>	-3,89%	-4,72%	-0,07%	-4,95%	-4,59%	-3,99%	-0,89%	-4,86%
<b>Cactus</b>	-5,33%	-5,60%	-1,52%	-5,77%	-5,87%	-6,17%	-1,58%	-5,80%
<b>Traffic</b>	-4,19%	-4,47%	-0,84%	-4,79%	-4,04%	-4,42%	-0,10%	-4,15%
<b>Média</b>	-5,40%	-5,47%	+0,18%	-6,17%	-5,10%	-5,50%	-0,63%	-5,27%

**Tabela 7 - Redução do tempo de codificação (QP=37)**

Sequência	Tempo de Codificação / QP = 37							
	SVT1	SVT2	SVT3	SVT4	SVT5	SVT6	SVT7	SVT8
<b>BasketBallPass</b>	-5,21%	-4,86%	-0,60%	-5,29%	-4,59%	-0,86%	-2,05%	-5,17%
<b>BQSquare</b>	+3,92%	-5,39%	+0,70%	-6,04%	-4,15%	-4,32%	+0,15%	+3,65%
<b>BasketBallDrill</b>	-4,61%	-4,57%	-1,62%	-5,88%	-6,50%	-6,63%	-1,96%	-6,39%
<b>BQMall</b>	-7,55%	-5,37%	-2,79%	-7,62%	-5,23%	-4,92%	-2,61%	-7,08%
<b>Mobcal</b>	-4,36%	-5,76%	-0,19%	-5,94%	-4,15%	-5,69%	+0,73%	-5,39%
<b>Parkrun</b>	-8,04%	-7,80%	-1,83%	-7,33%	-6,16%	-7,14%	-0,31%	-7,02%
<b>Shields</b>	-6,06%	-6,07%	-1,45%	-6,19%	-5,87%	-6,02%	-0,70%	-5,77%
<b>BQTerrace</b>	-3,22%	-3,63%	-1,19%	-3,12%	-3,35%	-2,88%	-0,09%	-4,11%
<b>Cactus</b>	-4,48%	-4,81%	-0,93%	-4,85%	-5,73%	-6,70%	-1,32%	-5,32%
<b>Traffic</b>	-4,19%	-4,22%	-0,79%	-4,46%	-3,87%	-4,07%	-0,16%	-3,57%
<b>Média</b>	-4,38%	-5,25%	-1,07%	-5,67%	-4,96%	-4,92%	-0,83%	-4,62%

O tempo de codificação foi, em média, reduzido em relação à técnica SVT original [11] em todas as configurações usadas, independentemente do QP. É possível notar que na maior parte das configurações usadas, à medida que o QP vai aumentando a redução do tempo de codificação tem tendência a diminuir. Essa diferença é mais notória comparando o QP igual a 22 com os restantes (27,32 e 37). Ainda assim, qualquer uma das configurações apresenta sempre uma redução comparativamente à técnica SVT original [11].

A Tabela 8 representa a média geral, onde se pode ver que tal como na redução da taxa de bits, os melhores resultados são obtidos quando a decisão de posicionamento do bloco transformado é adquirido através do cálculo do centro de massa da SAD das linhas e das colunas (sempre que os formatos de bloco não diferem entre as configurações). Neste caso, a que obtém uma maior diminuição de tempo de codificação é a configuração SVT4.

**Tabela 8 - Redução do tempo de codificação – Média dos diferentes QPs em todas as sequências utilizadas**

<b>QP</b>	<b>Tempo de Codificação</b>							
	<b>SVT1</b>	<b>SVT2</b>	<b>SVT3</b>	<b>SVT4</b>	<b>SVT5</b>	<b>SVT6</b>	<b>SVT7</b>	<b>SVT8</b>
<b>22</b>	-6,78%	-6,62%	-3,04%	-8,08%	-5,96%	-6,20%	-2,76%	-6,54%
<b>27</b>	-5,45%	-5,74%	-1,36%	-6,06%	-4,93%	-4,86%	-0,40%	-5,76%
<b>32</b>	-5,40%	-5,47%	+0,18%	-6,17%	-5,10%	-5,50%	-0,63%	-5,27%
<b>37</b>	-4,38%	-5,25%	-1,07%	-5,67%	-4,96%	-4,92%	-0,83%	-4,62%
<b>Média Geral</b>	-5,50%	-5,77%	-5,29%	-6,50%	-5,24%	-5,37%	-1,16%	-5,55%







## Capítulo 4: Conclusão

Nesta monografia foi apresentado um estudo sumário do estado de arte em codificação de vídeo, bem como conceitos fundamentais, utilizando a norma H.264/AVC como modelo.

O objetivo de redução de complexidade computacional da técnica SVT foi alcançado com êxito, conforme apresentado no Capítulo 3 deste texto. A adição de novos formatos de bloco aos já existentes originalmente na técnica SVT, juntamente com a nova decisão de posicionamento e formato de bloco, foram uma mais-valia. Das duas ideias propostas para decisão de posicionamento e formato de bloco, a que se concluiu ser mais benéfica foi aquela na qual a posição do bloco transformado era obtida através do cálculo do centro de massa da SAD das linhas e colunas. A ideia de decisão em que a posição era obtida através da linha e coluna com maior SAD, apesar de também obter resultados positivos comparativamente à técnica SVT original [11], comparando com a ideia de decisão calculada através do centro de massa, teve sempre resultados inferiores (quando os formatos de bloco não diferem entre as configurações). Estes resultados mencionados dizem respeito, quer à redução da taxa de bits, quer à redução do tempo de codificação. Embora na decisão de posicionamento obtida através do centro de massa, a redução do tempo de codificação seja menor em alguns QPs (comparativamente à decisão de posicionamento obtida através da linha e coluna com maior SAD), esta possui uma média geral superior.

A configuração, de entre as oito testadas, que obteve os melhores resultados na redução da taxa de bits foi a configuração SVT2, que conseguiu uma redução média de 0,52%, comparativamente à técnica SVT original [11]. Nesta configuração o posicionamento do bloco transformado é obtido através do cálculo do centro de massa da SAD das linhas e colunas que, como referido anteriormente, se concluiu ser a mais benéfica. Esta é a que usa todos os formatos de blocos (8x8, 16x4, 4x16, 4x4, 8x4, 4x8, 16x8 e 8x16), usando na totalidade 8 formatos de blocos diferentes. Não é de admirar que tenha sido esta a configuração com melhores resultados, porque devido ao facto do erro de predição existente nos macroblocos não possuir sempre as mesmas características, a probabilidade de este ser melhor localizado é maior. Apesar disto, a diferença para a segunda melhor configuração (SVT4) não é muito grande, sendo apenas de 0,01%.

A configuração que obteve a maior redução de tempo de codificação foi a configuração SVT4, com uma redução média de 6,50%, comparativamente à técnica SVT

original [11]. Nesta configuração o posicionamento do bloco transformado é obtido através do cálculo do centro de massa da SAD das linhas e colunas e usa 4 diferentes formatos de bloco (8x8, 16x4, 4x16 e 4x4). A configuração SVT2, que obteve a maior redução da taxa de bits, no que toca a redução de tempo de codificação foi a segunda melhor configuração, com uma redução média de 5,77% em relação ao original.

Por último, foi também possível concluir, através da representação gráfica das curvas R-D, que a redução da taxa de bits é maior para taxas de bits mais altas. Este facto verificou-se para a maioria das sequências testadas. Em relação ao QP, na maioria das configurações utilizadas, quanto mais baixo for, maior é a redução de tempo de codificação. Esta diferença é mais visível comparando o QP igual a 22 com os restantes (27, 32 e 37).

Em jeito de conclusão geral pode-se dizer que a adição de novos formatos de bloco transformado juntamente com as novas decisões de posicionamento e formato de bloco transformado foram uma mais-valia para a técnica SVT.





## Bibliografia

- [1] G. J. S. G. B. a. A. L. T. Wiegand, *Overview of the H.264/AVC video coding standard*, 2003.
- [2] <http://www.itu.int/ITU-T/studygroups/com16/>, *ITU-T Video Coding Experts Group (VCEG)*, 2011.
- [3] <http://mpeg.chiariglione.org/>, *ISO/IEC Moving Picture Experts Group (MPEG) (ISO/IECJTC1/SC29/WG11)*, 2011.
- [4] I. E. Richardson, *Video Codec Design – Developing Image and Video Compression Systems*, Chichester: John Wiley and Sons, 2002.
- [5] I. E. Richardson, *H.264 and MPEG-4 Video Compression : Video Coding for Next-Generation Multimedia*, Chichester: John Wiley and Sons, 2003.
- [6] I. E. Richardson, *The H.264 Advanced Video Compression Standard - 2nd ed.*, Wiley.
- [7] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*, UK: The Institute of electrical Engineers, 2003.
- [8] Y. S. a. H. Sun, *Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms and Standards*, Boca Raton: CRC Press, 1999.
- [9] K. U. J. L. a. M. G. C. Zhang, *Video coding using spatially varying transform*, PSIVT 2009.
- [10] ISO/IEC-JTC1/SC29/WG11, *Description of video coding technology proposal by Tandberg, Nokia, Ericsson*, ed. Dresden, Germany, 2010.
- [11] K. U. J. L. a. M. G. Z. Cixun, *Video coding using Variable Block-Size Spatially Varying Transforms*, 2009.
- [12] G. Bjontegaard, *Calculation of average PSNR differences between RD-curves*, VCEG Doc.VCEG-M33, March 2001.