



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

DEPARTAMENTO DE ENGENHARIA DE ELECTRÓNICA E
TELECOMUNICAÇÕES E DE COMPUTADORES

ENGENHARIA DE ELECTRÓNICA E TELECOMUNICAÇÕES
RAMO DE TELECOMUNICAÇÕES E COMPUTADORES

COMPRESSÃO DE IMAGENS EM SISTEMAS DE VIDEOVIGILÂNCIA

João Guilherme Folgado Tonel Monteiro Lobato
(Bacharel)

TRABALHO DE PROJECTO PARA OBTENÇÃO DO GRAU DE MESTRE
EM ENGENHARIA DE ELECTRÓNICA E TELECOMUNICAÇÕES

Orientadores:

Fernando Manuel Gomes de Sousa
Artur Jorge Ferreira

Júri:

Coordenador do Mestrado
Pedro Miguel Torres Mendes Jorge
Fernando Manuel Gomes de Sousa
Artur Jorge Ferreira

JANEIRO DE 2009

Resumo

As características dos vídeos recolhidos num sistema de videovigilância são bastante diferentes dos vídeos recolhidos para filmes. Cada sistema de videovigilância é único. Este trabalho trata a implementação, e especialização, de um algoritmo de codificação de vídeo para um sistema particular de videovigilância composto por três câmaras fixas.

O algoritmo desenvolvido tira partido do facto de se conhecer previamente as características físicas e de utilização do local. Utilizando uma codificação inteligente, a codificação das imagens recolhidas depende do dia da semana, hora, câmara e da detecção de alterações na área que se encontra sob vigilância. A especialização é realizada através da configuração do codificador, tendo em conta critérios pré-definidos, de modo a obter uma relação adequada entre a qualidade de imagem e a compressão dos dados.

As imagens são independentemente codificadas pela seguinte ordem de operações: transformada discreta do co-seno, quantização, *run-length* e codificação de Huffman. São armazenadas todas as imagens onde for detectado movimento, não a totalidade da imagem, mas apenas as regiões onde foi detectado movimento sendo utilizado uma decomposição em árvore quaternária.

Foi implementado um protótipo de modo a avaliar o desempenho do algoritmo de codificação. O algoritmo, quando comparado com a técnica de codificação usada no sistema de videovigilância, apresenta uma maior taxa de compressão. As imagens decodificadas apresentam qualidade suficiente para identificar as pessoas em pontos críticos e seguir os seus movimentos ao longo do corredor.

Palavras-Chave: Compressão de imagens, codificação inteligente, MJPEG, detecção de movimento, árvore quaternária.

Abstract

In video surveillance applications, the nature of the video captured is very different from that captured for movies. Moreover, each video surveillance system is unique. The aim of this project is to develop a specialized video encoder for a particular video surveillance system with three static cameras.

The developed algorithm takes advantage of knowing, in advance, the physical and the utilization characteristics of the place. Using an intelligent coding approach, the codification of the collected images depends on the day of the week, hour, camera, and on the detection of changes in the place that is under surveillance. Taking into account pre-defined criteria, the specialization is performed through encoder configuration in order to achieve the adequate tradeoff between image quality and data compression.

The images are independently encoded through the following sequence of operations: discrete cosine transform, quantization, run-length and Huffman encoding. All images where motion was detected are coded, not the full image, but only the regions where the activity was detected using a quadtree decomposition.

To evaluate the encoder algorithm performance, a prototype was implemented. When compared with the coding technique used in the video surveillance system, the algorithm achieves greater compression ratio thus saving memory. The decoded images exhibit enough quality to identify the actors at critical points as well as their movements along the scene.

Keywords: Image compression, intelligent encoding, MJPEG, motion detection, quadtree.

Agradecimentos

Quero agradecer aos professores Artur Ferreira e Fernando Sousa por todo o apoio prestado e tempo dispendido, pela sua orientação que possibilitou o prosseguimento do trabalho em momentos de dúvidas, por toda a disponibilidade e conselhos úteis.

Agradeço a todos os familiares e amigos que me ajudaram na concretização deste trabalho, motivando-me e apoiando-me ao longo de todo o curso.

Agradeço a todos os docentes do ISEL pelos conhecimentos transmitidos ao longo do curso, bem como toda a ajuda e tempo que sempre tiveram.

Índice

Resumo	iii
Abstract.....	iv
Agradecimentos	v
Capítulo 1 – Introdução	1
Capítulo 2 – Técnicas de codificação	5
2.1 JPEG e MJPEG.....	5
2.1.1 DCT	6
2.1.2 Quantização	8
2.1.3 Codificação por repetição.....	9
2.1.4 Codificação entrópica.....	10
2.2 Codificação inteligente.....	10
Capítulo 3 – Descrição da solução	13
3.1 Detecção de movimento e decomposição da imagem.....	14
3.1.1 Detecção de movimento	14
3.1.2 Decomposição da imagem.....	18
3.1.3 Escolha do cenário.....	22
3.2 Codificação.....	25
3.3 Armazenamento.....	26
3.4 Descodificação.....	28
Capítulo 4 – Resultados Experimentais.....	29
4.1 Detecção de movimento	30
4.2 Influência de diferentes valores para o limiar	39
4.3 Comparação do sistema implementado com outras técnicas.....	42
4.4 Tempo de execução do algoritmo.....	52
Capítulo 5 – Conclusões.....	55
Referências	57

Capítulo 1 – Introdução

Este trabalho tem como objectivo a implementação, e especialização, de um algoritmo de codificação de vídeo para dotar o sistema de videovigilância, instalado no pavilhão Centro de Cálculo do Instituto Superior de Engenharia do Lisboa, com um algoritmo de codificação inteligente.

Este sistema dispõe de três câmaras fixas, instaladas no tecto, que são usadas para vigiar a entrada e saída e o movimento de pessoas no corredor do referido pavilhão.

No contexto deste trabalho, entende-se sistema de videovigilância como a observação de um local, a distribuição das imagens recolhidas para observação remota do local, e o seu armazenamento para visualização posterior. O crescente número de câmaras de videovigilância trouxe consigo o aumento da quantidade de memória para armazenamento e o congestionamento da rede de comunicação. Para fazer frente a estes problemas surgiu a necessidade de compressão dos dados de modo a reduzir o tempo de transmissão e a quantidade de memória para armazenamento. Conhecidas as características e os hábitos de utilização do local vigiado, na codificação inteligente combina-se esse conhecimento prévio com uma codificação flexível. O conhecimento prévio é usado para parametrizar o codificador tendo em vista o uso adequado de recursos.

Contudo, a optimização da codificação para locais específicos tem a desvantagem de o algoritmo implementado estar intrinsecamente relacionado com as condições para onde é optimizado com prejuízo para a sua utilização noutras circunstâncias.

O algoritmo desenvolvido tira partido do facto de se conhecer os hábitos de utilização do local para optimizar a compressão das imagens.

A Figura 1 mostra o diagrama do local onde está instalado o sistema de videovigilância, o corredor do Centro de Cálculo (Edifício 4) situado no Instituto Superior de Engenharia de Lisboa.

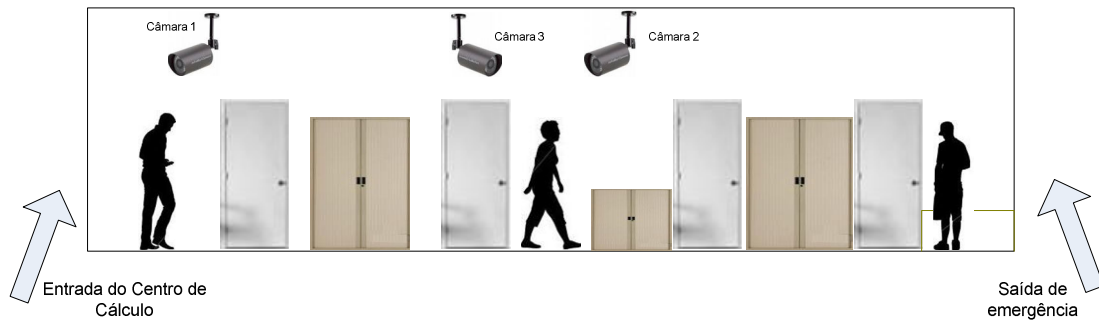


Figura 1 – Local específico da vídeo vigilância (vista lateral)

Em relação a este local, sabe-se que provavelmente não haverá circulação de pessoas das 0 até às 6 horas da manhã e durante todo o fim-de-semana, à excepção do sábado de manhã. Isto é, assume-se que há movimento durante o horário em que decorrem aulas no ISEL e durante o período das operações de limpeza (de Segunda-feira a Sábado das 6 às 8 horas.

A Figura 2 mostra uma imagem captada por cada câmara, durante o horário de funcionamento.



Figura 2 – Imagem captada por cada câmara

No sistema de videovigilância considerado, as imagens recolhidas pelas câmaras são distribuídas em formato AVI com resolução 480×640 pontos. O ritmo de aquisição varia entre 1 e 5 imagens por segundo.

Neste trabalho, assumindo esse formato de representação, desenvolve-se uma técnica de codificação inteligente para distribuição e armazenamento remoto mais eficientes.

O algoritmo a especializar para codificação inteligente, baseia-se no modo de funcionamento com perda da recomendação JPEG (*Joint Photographic Experts Group*) [1] para codificar cada imagem em separado. Tal como nesta recomendação, é introduzida distorção de forma controlada, tirando partido das limitações da visão humana, nomeadamente na percepção da cor e das componentes de alta frequência. Todas as imagens onde for detectado movimento serão armazenadas, mas codificando apenas as regiões em que houve alteração.

Para avaliar o codificador foi implementado um protótipo em MATLAB.

Em relação ao formato de distribuição usado no sistema de videovigilância, verifica-se a redução da dimensão dos ficheiros usados para armazenar as imagens recolhidas, embora havendo degradação da qualidade das imagens. Contudo, as imagens decodificadas apresentam qualidade suficiente para identificar as pessoas e seguir os seus movimentos ao longo do corredor.

O resto do documento está organizado da forma que se apresenta de seguida. No capítulo 2 são descritos os conceitos abordados pelo trabalho. O capítulo 3 apresenta o trabalho desenvolvido. No capítulo 4 são apresentados os resultados experimentais. O capítulo 5 contém as conclusões relativas ao trabalho desenvolvido e ideias para trabalho futuro.

Capítulo 2 – Técnicas de codificação

2.1 JPEG e MJPEG

O acrónimo JPEG [1][2] é o nome original do grupo de trabalho que está na origem da recomendação ITU-T (ISO/IEC 10917-1) do ITU-T (*Telecommunication Standardization Sector of the International Telecommunications Union*).

O JPEG é, actualmente, um dos formatos de codificação de imagem mais utilizados, uma vez que obtém elevadas taxas de compressão com qualidade elevada de imagem. A codificação JPEG é normalmente utilizada no modo de funcionamento com perdas, o que significa que existe degradação de qualidade na imagem sendo que estas perdas são reflectidas na compressão da mesma, ou seja, para uma imagem com maior qualidade corresponde a menor compressão e vice-versa.

A recomendação JPEG foi desenvolvida para imagens naturais com poucas variações de brilho ou de cor, não tendo bom desempenho em imagens gráficas, como texto ou imagens digitais (criadas em computador). Estas apresentam contornos e áreas bem definidas e elevado contraste entre pixels adjacentes.

Apesar de ser uma recomendação específica para imagens, é possível aplicá-la em sequências de vídeo, surgindo o conceito de MJPEG. Neste, cada imagem do vídeo é tratada como uma imagem independente. Na Figura 3, apresenta-se o diagrama de blocos da codificação e da descodificação especificada pela recomendação JPEG.

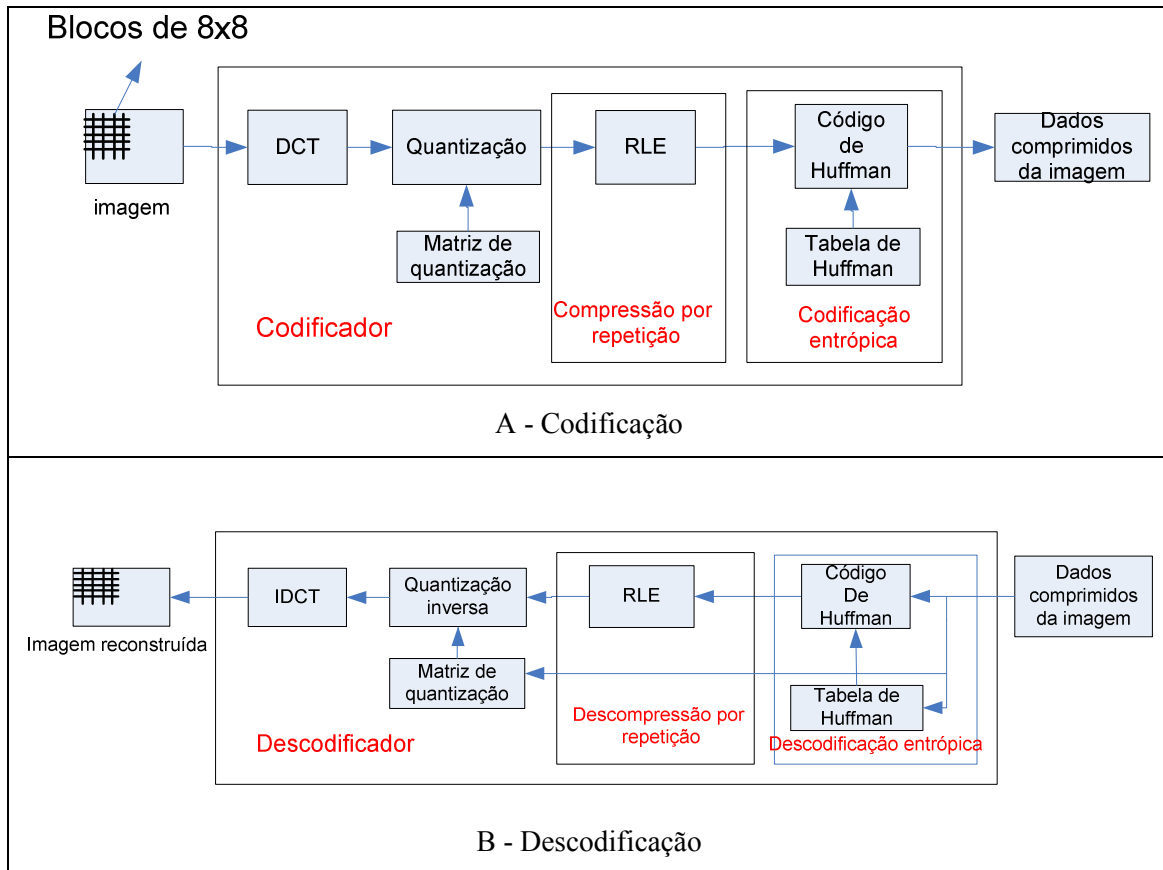


Figura 3 – Codificação/Decodificação JPEG

A imagem é decomposta em blocos de 8×8 , para redução da complexidade de processamento. As operações envolvidas na codificação são:

- DCT (*Discrete Cosine Transform*) – concentra a informação de energia da imagem nos primeiros coeficientes da transformada.
- Quantização – introdução de perda controlada de informação.
- Codificação – minimiza o número de dígitos binários usados para representar a imagem.

As subsecções seguintes descrevem em detalhe cada bloco.

2.1.1 DCT

No JPEG [1][2], a transformada discreta do co-seno é aplicada a blocos de 8×8 pixels. À entrada do codificador, a imagem é dividida em blocos dessa dimensão, em que os seus valores são deslocados de inteiros sem sinal no intervalo de $\{0, 2^P - 1\}$ para valores inteiros com sinal no intervalo entre $\{-2^{P-1}, 2^{P-1} - 1\}$, em que P representa o número de bit por pixel.

Para um bloco V de dimensão 8×8 , a DCT é descrita por

$$T[i, j] = c(i, j) \left[\sum_{x=0}^7 \sum_{y=0}^7 V[x, y] \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2y+1)j\pi}{16}\right) \right], \quad (1)$$

com

$$c(i, j) = \frac{2}{8}, \text{ para } i, j \neq 0,$$

$$c(i, j) = \frac{1}{8}, \text{ para } i, j = 0.$$

Cada bloco da imagem é um sinal discreto, com 64 pontos, a duas dimensões (x e y). A DCT desse sinal é o conjunto dos coeficientes da sua representação usando 64 sinais ortogonais entre si correspondentes a cada uma das 64 frequências espaciais, de duas dimensões, únicas. Os 64 coeficientes, normalmente referidos como coeficientes da DCT, são determinados unicamente pelos 64 pontos do sinal de entrada. Estes coeficientes podem ser entendidos como o valor relativo das frequências espaciais, em duas dimensões, contidas no sinal. O primeiro coeficiente é designado por DC e os restantes 63 são designados coeficientes AC. Havendo semelhança entre os valores do sinal há concentração de energia nas frequências mais baixas. Para um bloco de 8×8 de uma imagem típica, a maior parte dos coeficientes tem amplitude próxima de zero.

Na decodificação, a IDCT reverte este processo. A partir dos 64 coeficientes da DCT obtém-se o bloco da imagem com 64 pontos, de acordo com

$$V[x, y] = \left[\sum_{i=0}^7 \sum_{j=0}^7 c(i, j) T[i, j] \cos\left(\frac{(2x+1)i\pi}{16}\right) \cos\left(\frac{(2y+1)j\pi}{16}\right) \right], \quad (2)$$

com

$$c(i, j) = \frac{2}{8}, \text{ para } i, j \neq 0,$$

$$c(i, j) = \frac{1}{8}, \text{ para } i, j = 0.$$

2.1.2 Quantização

Após o cálculo da DCT, cada coeficiente da DCT é uniformemente quantizado usando uma tabela de quantização com 64 elementos, sendo que esta é um parâmetro do codificador. Cada elemento da tabela é um valor inteiro, entre 1 e 255, que define o intervalo do quantizador para o coeficiente da DCT correspondente. A recomendação JPEG inclui dois exemplos de matrizes de quantização, para a luminância e crominância, reproduzidos na Figura 4.

<p><i>MatrizQuantizacaoY</i></p> $= \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 97 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$ <p>A – Matriz de quantização Y</p>	<p><i>MatrizQuantizaçãoCbCr</i></p> $= \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$ <p>B – Matriz de quantização CbCr</p>
---	--

Figura 4 – Matriz de quantização

Na quantização é introduzida distorção, sendo a única fonte de perda na codificação JPEG. Isto é, os coeficientes da DCT são representados com a precisão necessária para obter uma imagem com a qualidade desejada. A quantização consiste na divisão de cada coeficiente da DCT pelo elemento da matriz de quantização correspondente, sendo o resultado arredondado para o valor inteiro mais próximo

$$ValorQuantizado = round\left(\frac{CoeficienteDCT}{ValorQuantização}\right). \quad (3)$$

No decodificador, cada coeficiente da DCT é obtido multiplicando o valor quantizado pelo correspondente elemento da matriz de quantização.

$$CoeficienteDCT = ValorQuantizado * ValorQuantização. \quad (4)$$

Em geral, os valores obtidos não são iguais aos valores originais uma vez que na quantização os valores quantizados são arredondados para o inteiro mais próximo.

2.1.3 Codificação por repetição

Após a quantização, o coeficiente DC é tratado de forma separada dos 63 coeficientes AC. Assumindo que existe forte correlação entre blocos adjacentes, é codificada a diferença entre esse coeficiente e o valor DC do bloco anterior na ordem de codificação, como se pode observar na Figura 5.

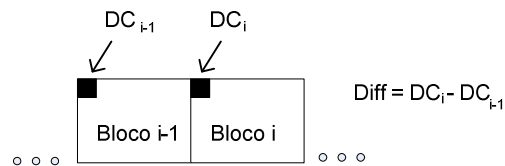


Figura 5 – Codificação diferencial da componente DC

Este tratamento diferenciado dos coeficientes DC é compensador porque estes contêm, normalmente, uma significativa fracção da energia total da imagem. Em seguida, todos os coeficientes quantizados são ordenados em resultado do varrimento em zigue-zague, como se observa na Figura 6. Esta ordenação dos coeficientes coloca as frequências mais baixas, cujos valores dos coeficientes tendem a ser diferentes de zero, antes das altas frequências que, normalmente, apenas apresentam zeros.

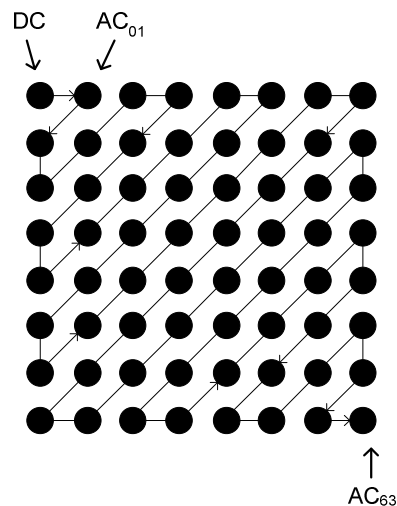


Figura 6 – Sequência zigue-zague

O varrimento em zigue-zague procura maximizar as sequências de zeros. O RLE (*Run-Length Encoding*) é uma técnica usada para codificar cadeias de caracteres onde existem sequências longas de caracteres iguais. Na sua aplicação no JPEG [2][3], cada coeficiente diferente de zero é caracterizado, pela dimensão do número de coeficientes

nulos que o precedem e pelo número de bits necessário para representar esse coeficiente.

2.1.4 Codificação entrópica

A codificação entrópica efectua compressão sem perdas dos coeficientes quantizados da DCT baseando-se nas suas características estatísticas. Um código de codificação entrópica é o de Huffman [3][4], sendo este um método de compressão que usa as probabilidades de ocorrência dos símbolos no conjunto de dados a ser comprimido para estabelecer o código, de tamanho variável, para cada símbolo. Este tipo de codificação requer que a tabela de Huffman seja especificada e que a que é usada para efectuar a compressão dos dados seja a mesma para efectuar a descompressão dos mesmos dados. Esta tabela pode ser pré-definida ou calculada individualmente para cada imagem, sendo que para este caso necessário é efectuar uma análise estatística das características da imagem antes de se efectuar a compressão.

2.2 Codificação inteligente

Em [5] é introduzido o conceito de codificação inteligente nos sistemas de videovigilância.

Na codificação inteligente procura-se usar a codificação adequada a cada situação. Cada imagem recolhida é analisada de modo a determinar para qual das regras definidas, para cada cenário, esta se adequa. Estas regras incluem a presença de movimento, a quantificação desse mesmo movimento e a ausência deste. Os resultados das análises são usados para configurar o codificador, de modo a que este se adapte ao cenário em causa, e garanta níveis de compressão baseados na dinâmica dos personagens numa sequência de imagens. A capacidade para os parâmetros do codificador se alterarem rapidamente consoante a dinâmica dos personagens numa sequência de imagens permite obter maiores taxas de compressão.

A parametrização do codificador tem em conta, neste caso, o dia da semana e a hora em que o movimento é detectado, a percentagem da imagem em que houve alteração e a câmara onde o movimento é detectado.

A Figura 7 ilustra a parametrização do codificador consoante os parâmetros.

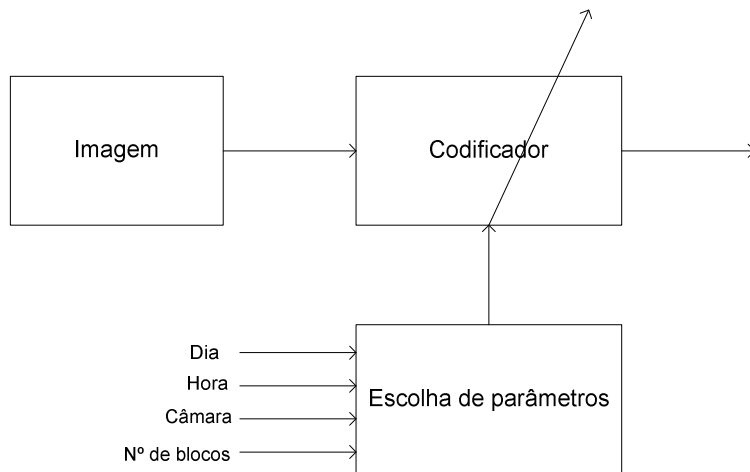


Figura 7 – Comutação/escolha do cenário e parametrização do codificador

Como se observa na Figura 7 consoante a escolha de parâmetros, o codificador será parametrizado com valores associados a esses mesmos parâmetros, sendo que deste modo se obtêm diferentes taxas de compressão.

Capítulo 3 – Descrição da solução

Este capítulo trata da descrição detalhada da solução adoptada para a implementação do algoritmo de codificação para o sistema de vídeo vigilância. Mostra-se na Figura 8 o diagrama de blocos da solução adoptada e, nas próximas secções analisa-se em detalhe cada um destes blocos.

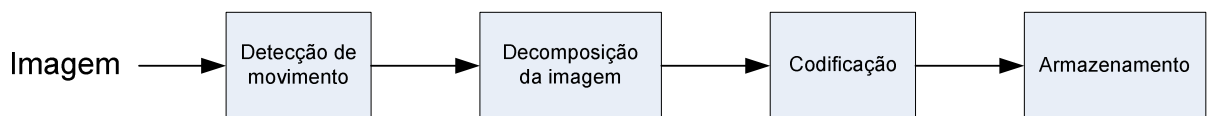


Figura 8 – Diagrama de blocos

Para a detecção de movimento usou-se dois métodos:

- a subtracção de uma imagem com uma imagem de fundo
- subtracção de duas imagens consecutivas

Estes dois métodos são explicados, detalhadamente, na secção 3.1.1.

O limiar usado para detecção de alterações é definido para cada câmara, tendo em conta a parte do local que esta vigia.

É realizada uma divisão, em blocos, da imagem resultante do método da detecção de movimento sendo estes blocos divididos sucessivamente, nas regiões cujo valor seja superior ao limiar, até atingirem uma dimensão mínima de 8×8 . Em seguida são armazenadas as coordenadas relativas a esses blocos, uma vez que se pretende apenas codificar as regiões em que houve alteração.

Foi pré-definida uma região de interesse para a zona observada por cada câmara. A codificação de cada imagem restringe-se à região de interesse e depende do instante em que foi adquirida. O codificador é parametrizado tendo em conta a câmara, a percentagem da imagem em que houve alteração, a hora e o dia. A distorção introduzida depende desta parametrização.

Após transformação de cada bloco e quantização dos coeficientes, é usado codificação por repetição e como código de compressão o código de Huffman [3][4].

No armazenamento é guardada a hora, o dia, as coordenadas dos blocos da detecção de movimento, os coeficientes codificados, as repetições provenientes do código RLE, a tabela de Huffman, a componente DC associada a cada bloco, a qualidade com que a imagem foi codificada e a câmara respeitante a cada imagem.

3.1 Detecção de movimento e decomposição da imagem

Os primeiros dois blocos da Figura 8 podem ser observados com mais detalhe na Figura 9.

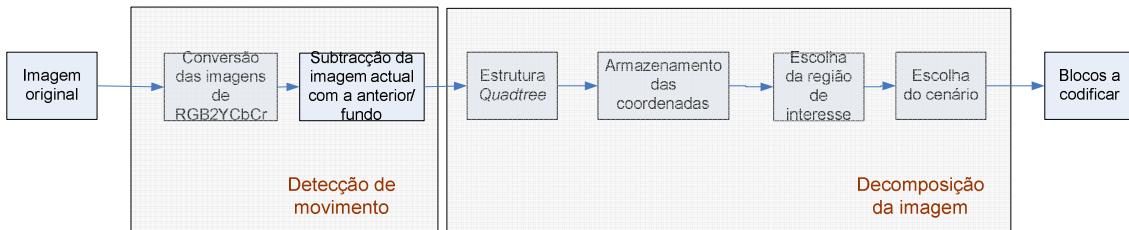


Figura 9 – Bloco da detecção movimento e da decomposição da imagem

3.1.1 Detecção de movimento

No método da subtração de duas imagens consecutivas (Método A), é usada a diferença da imagem actual e da anterior. Assim, se uma imagem for expressa por

$$I(x, y) = f(x, y, t) \quad (5)$$

sendo $f(x, y, t)$ a representação matricial da imagem com coordenadas x e y e o instante da imagem a considerar t obtém-se,

$$\Delta I(x, y) = |f(x, y, t_2) - f(x, y, t_1)|, \quad (6)$$

em que $\Delta I(x, y)$ representa o módulo da diferença entre a imagem actual $f(x, y, t_2)$ e a anterior $f(x, y, t_1)$. O módulo da diferença entre duas imagens consecutivas tem valores de zero ou próximos de zero nas regiões onde existe pouca variação entre as duas imagens e valores superiores nas outras regiões.

Definido um valor de limiar, considera-se que nos pontos que contenham valores acima desse valor existiu movimento. O valor do limiar não pode ser muito baixo

porque, mesmo não havendo movimento, as imagens sucessivas não são iguais, e também não pode ser muito alto pois desse modo não se detecta o movimento das pessoas.

Para o método da subtracção da imagem actual com uma imagem de fundo (Método B) é armazenada uma imagem de fundo de ambiente (imagem de referência) sendo esta subtraída à imagem actual. Assim, se a imagem de fundo for representada por,

$$I_{fundo} = f_{fundo}(x, y) , \quad (7)$$

o módulo da diferença entre a imagem de fundo e a imagem actual $f(x, y, t)$ é,

$$\Delta I(x, y) = |f(x, y, t) - f_{fundo}(x, y)| , \quad (8)$$

Considera-se como existência de movimento quando o módulo da diferença de duas imagens é superior ao limiar.

É utilizado o método A para as câmaras 2 e 3. Pretende-se deste modo detectar os movimentos de passagem pelo corredor descartando os movimentos das pessoas que se encontram praticamente paradas no corredor. Este método é usado durante o horário em que se considera ser normal a existência de movimento no Centro de Cálculo.

Dadas as condições para onde o algoritmo é optimizado, este método tem a vantagem em que o módulo da diferença entre duas imagens consecutivas regista pouca ou nenhuma variação na luz, sendo possível através do valor de limiar ter algum controlo sobre o efeito desta variação, ficando apenas com os movimentos das pessoas.

É utilizado o método B para a câmara 1. Uma vez que esta câmara está a vigiar a porta do Centro de Cálculo é de extrema importância que qualquer entrada/saída bem como a presença de alguém junto à porta seja guardada e que esta informação permita a identificação posterior da pessoa. Com este método sempre que se encontrar alguém na zona observada pela câmara 1, independentemente de se encontrar parado ou em movimento, esta informação é guardada.

Quando o sistema arranca a primeira imagem é guardada e definida como imagem de fundo, daí ser de extrema importância garantir que não se encontra ninguém aquando do arranque do sistema. Apenas a primeira imagem de fundo é armazenada juntamente com os ficheiros que são codificados.

A imagem de fundo é actualizada sempre que não existe detecção de movimento, independentemente do método de detecção de movimento que é utilizado.

Deste modo, quando usado o método B, procura-se ter uma imagem de fundo o mais actual possível reduzindo o efeito da variação da intensidade luminosa ao longo do dia, para que estas não sejam consideradas como movimento. Deste modo caso exista uma alteração no cenário, existe sempre uma imagem de fundo actualizada que pode ser usada.

Para os períodos em que, provavelmente, não haverá circulação de pessoas é utilizado o método B para as 3 câmaras. Deste modo garante-se que qualquer alteração em relação a uma imagem de fundo é detectada e guardada. Presume-se que neste período não há movimento e qualquer alteração pode significar estar-se perante uma intrusão.

A utilização deste método tem implicações a nível da taxa de compressão uma vez que são detectados mais blocos para posterior codificação, logo a taxa de compressão diminui. Os resultados destes dois métodos são apresentados no Capítulo 4.

Optou-se por uma mudança no espaço de cor, das imagens, de RGB para YCbCr. Esta transformação é realizada através da relação

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,169 & -0,331 & 0,5 \\ 0,5 & -0,419 & -0,081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (9)$$

Na Figura 10 mostra-se duas imagens consecutivas em RGB e a luminância, bem como o módulo da diferença entre elas.

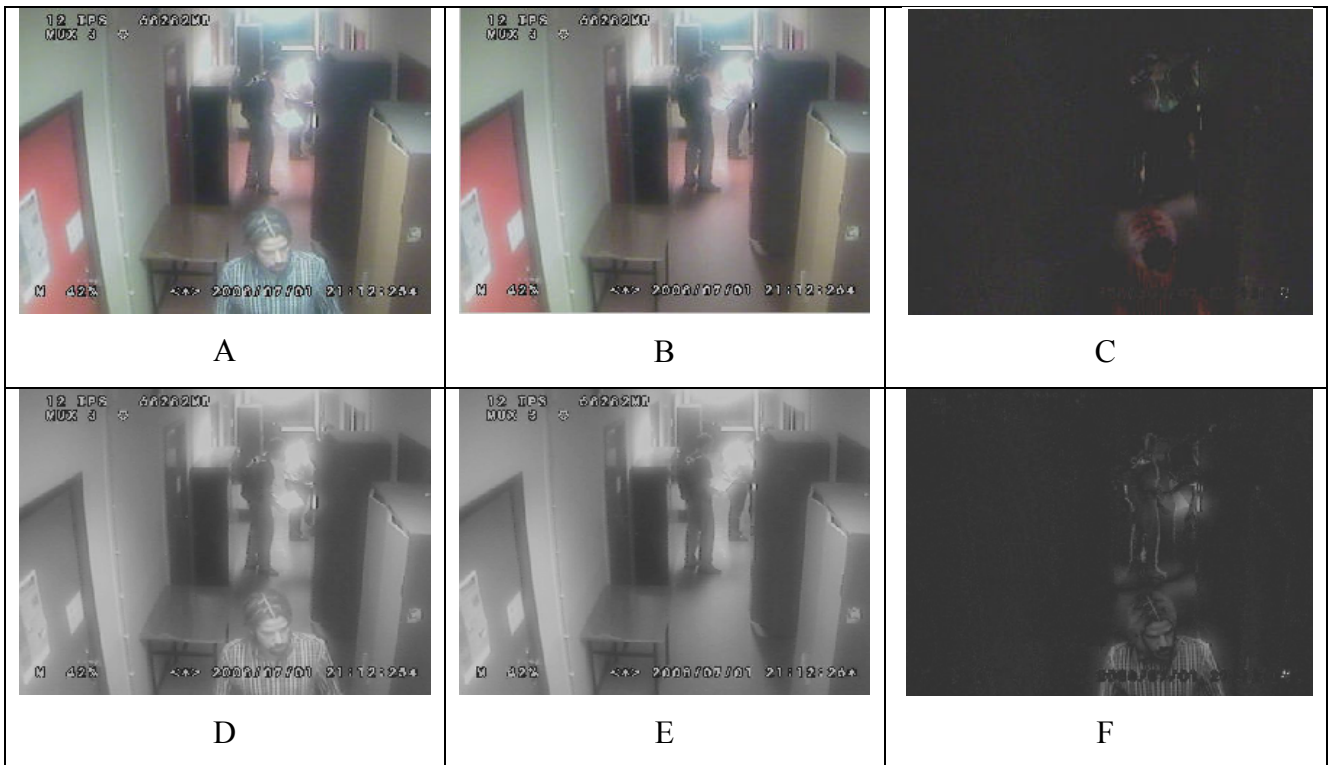


Figura 10 – Módulo da diferença de duas imagens consecutivas (A e B, D e E) da câmara 3, usando as componentes RGB e só com a componente Y

Através da Figura 10-F, é possível observar que usando apenas a componente Y (luminância), a imagem que se obtém contém as características essenciais sendo estas suficientes para se detectar movimento. Assim, opta-se por usar apenas a componente Y por uma razão de simplicidade.

Ainda assim muitas vezes a detecção do movimento apresenta algumas falhas, nomeadamente quando o movimento, de uma ou mais pessoas, entre duas imagens consecutivas é muito reduzido e outras vezes a cor da roupa das pessoas é igual naquele ponto com a cor da imagem anterior, não sendo detectado o movimento. Para tentar colmatar estas falhas na detecção do movimento usou-se a sequência de operações morfológicas, erosão seguida de dilatação. A erosão elimina pontos espúrios enquanto a dilatação amplia o contorno dos elementos existentes na imagem. Estas operações são aplicadas com um elemento estruturante quadrado de 8×8 .

Dado que as características do local são previamente conhecidas e como as três câmaras se encontram fixas, foi identificada, para cada câmara, a região da imagem onde à partida se sabe que não haverá alteração. Em cada imagem, a respectiva região complementar é designada por região de interesse. Ao ser detectado o movimento, é feita uma análise à região de interesse naquela câmara e caso o movimento se encontre fora da região de interesse é ignorado. Sendo detectado movimento na região de interesse é necessário encontrar as coordenadas relativas a esses blocos. Na Figura 11, para cada câmara, a região de interesse considerada é representada a branco.

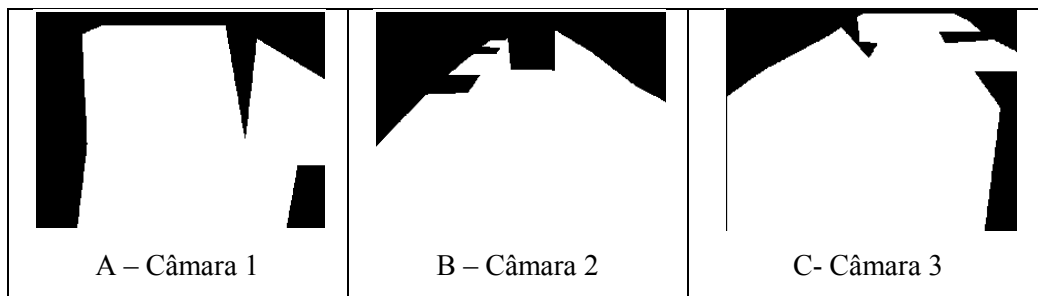


Figura 11 – Região de interesse

A região de interesse funciona como uma “máscara” para delimitar a imagem resultante da detecção de movimento. Deste modo, eventuais movimentos fora da região considerada são eliminados.

3.1.2 Decomposição da imagem

É utilizada a decomposição da imagem em árvore quaternária (*quadtree*) para representação das regiões em que foi detectado movimento. A ideia fundamental é que qualquer imagem quadrada pode ser dividida em quadrantes. Sendo que cada quadrante pode ser dividido novamente em subquadrantes e assim sucessivamente. Para cada imagem quadrada, verifica-se se o valor contido nessa região é superior ao nível de limiar. Em caso positivo esse bloco é novamente dividido e assim sucessivamente até a imagem de cada bloco possuir valores inferiores ao nível de limiar ou até a uma dada dimensão, que poderá ser 1×1 caso a dimensão do bloco seja potência inteira de dois.

Uma vez que se está a trabalhar com imagens rectangulares não é possível aplicar directamente o conceito da *quadtree*. Para tal foi implementado um algoritmo que, fazendo uso do conceito *quadtree*, não está limitado a imagens quadradas. A ideia do algoritmo é definir imagens quadradas dentro da imagem rectangular e fazer a decomposição destas usando árvores quaternárias. Esta divisão nos blocos é efectuada

até estes terem uma dimensão mínima de 8×8 . Feita a decomposição os blocos com dimensão mínima localizam as regiões da imagem onde existe movimento.

Uma vez que as imagens podem ter qualquer tipo de dimensão, para se conseguir definir sempre uma imagem quadrada dentro da imagem não quadrada, é feita uma análise inicial as dimensões da imagem. Caso seja necessário são adicionadas, ao fim da imagem, linhas/colunas até a um máximo de sete salvaguardando deste modo blocos dimensão mínima. Estas linhas/colunas são adicionados com o valor zero, deste modo quando for realizado o módulo da diferença entre duas imagens garante-se que não existirá diferenças entre as linhas/colunas que foram adicionadas.

Na Figura 12-A tem-se o resultado do módulo da diferença de duas imagens consecutivas, com uma resolução de 480×640 , onde não foi detectada nenhuma diferença, daí se obter uma imagem completamente preta, uma vez que ao se subtrair duas imagens iguais fica-se apenas com zeros o que corresponde à cor preta. Na Figura 12-B obtém-se a imagem resultante do módulo da diferença de duas imagens consecutivas onde são detectadas diferenças entre elas. A Figura 12-C mostra a decomposição da imagem quando não é detectado movimento em nenhuma região. Esta divisão da imagem resulta do algoritmo da decomposição da imagem. Na Figura 12-D é possível observar a divisão da imagem quando é detectado movimento (Figura 12-B). Para as imagens A e B da Figura 12 considerou-se um valor de limiar de 5.

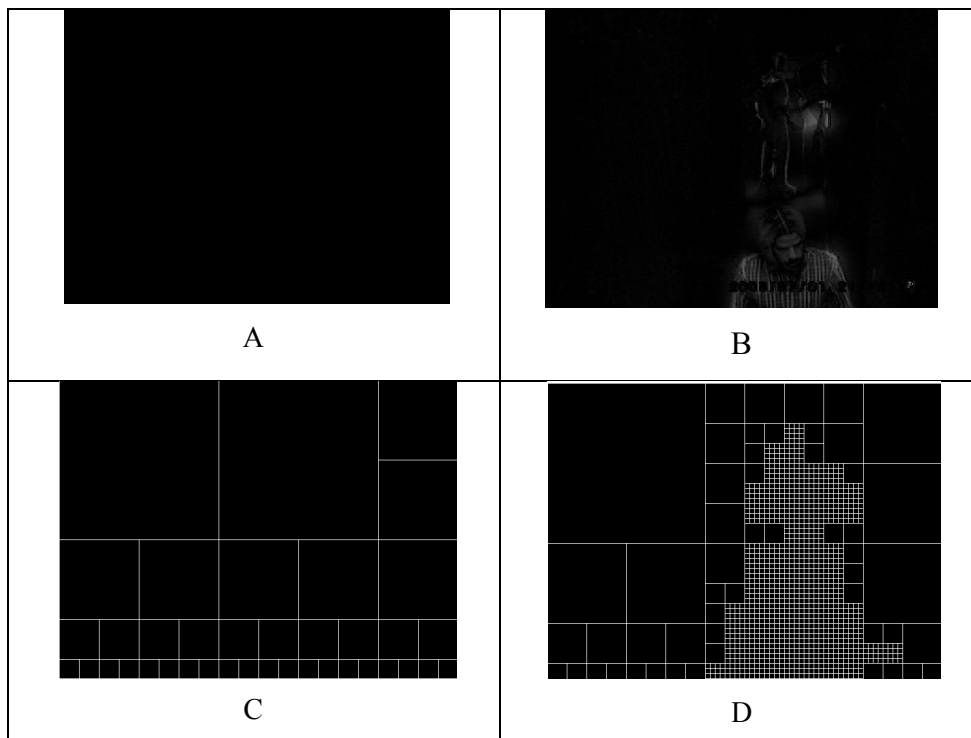


Figura 12 – Divisão da imagem resultante da detecção de movimento

Na Figura 12-B verifica-se que, quando existem diferenças entre duas imagens consecutivas, resulta numa imagem (Figura 12-D) em que as regiões onde se detectou movimento são divididas sucessivamente.

Na imagem resultante da detecção de movimento são analisadas as dimensões dos vários blocos que a constituem, e são guardadas as coordenadas relativas aos blocos que indicam as regiões onde foi detectado movimento. Caso não existam blocos com dimensão 8×8 , significa que não foi detectado movimento sendo esta imagem descartada.

É feito um “recorte”, ilustrado na Figura 13, na imagem actual, de blocos de 8×8 , onde as coordenadas guardadas o indicarem. Estes blocos são guardados uma vez que apenas estes são codificados.

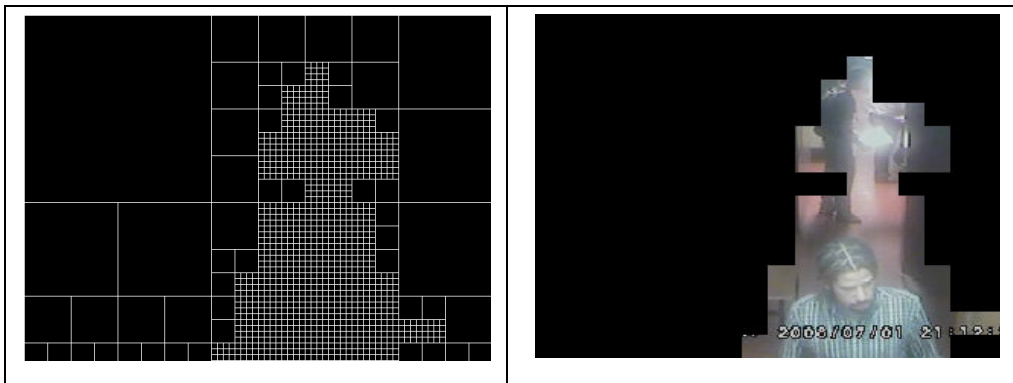


Figura 13 – Sub-imagem da imagem actual

Consoante a câmara é definido um valor de limiar que está associado a essa câmara. Após vários testes experimentais, apresentados no Capítulo 4, considerou-se o valor de 5 para as câmaras 2 e 3, e o valor de 10 para a câmara 1. Esta diferenciação no valor do limiar para a câmara 1 está relacionada com o facto de a zona observada por esta câmara ser a porta do Centro do Cálculo, o que leva a que uma grande parte da imagem tenha variações de luminosidade. Ao se aumentar o limiar nesta câmara pretende-se reduzir o número de blocos que são detectados como movimento aquando da variação da luminosidade. Para as restantes câmaras o valor de limiar é mais baixo uma vez que ao terem como raio de acção o corredor as variações da luminosidade são menos sentidas, como se pode observar na Figura 14-A,B,C. Com este valor é possível garantir que a detecção de movimento das pessoas não é afectada ainda que as sombras e algumas variações da luz sejam consideradas como movimento. A atribuição de um valor de limiar resulta num compromisso com a sensibilidade da detecção de

movimento. Sendo que esta sensibilidade do detector de movimento resultará num ficheiro com maior dimensão, quando o detector for mais sensível, uma vez que existirão mais blocos para codificar.

Na Figura 14 mostra-se a região de interesse de cada câmara.

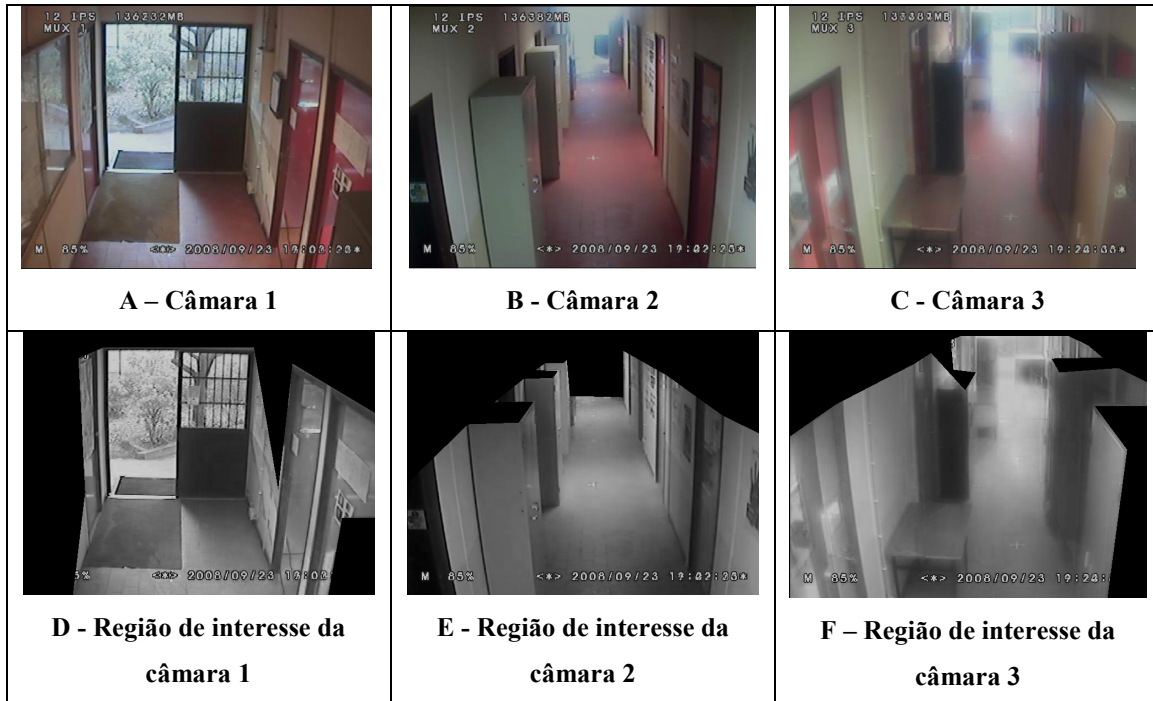


Figura 14 – Imagens obtidas pelas três câmaras e respectivas regiões de interesse

A região de interesse da câmara 1 é praticamente toda a imagem, visto a porta do Centro de Cálculo ser um ponto fundamental para a identificação das pessoas. Uma vez que as entradas/saídas do Centro de Cálculo são captadas através da câmara 1, considerou-se para a câmara 2 que essa zona da imagem pode ser descartada, sendo que deste modo a presença de movimento naquela localização é apenas detectada numa câmara. Para a câmara 3 é considerada como região de interesse todo o corredor.

De salientar o facto de existirem duas portas de gabinetes no Centro de Cálculo que em virtude da disposição das câmaras não se encontram visíveis.

Ao ser detectado movimento, é feita uma análise sobre a percentagem da imagem em que houve alterações de modo a decidir qual o cenário mais adequado, uma vez que para cada cenário está associado um nível de qualidade para a imagem a guardar. O nível de qualidade da imagem determina a taxa de compressão que se pretende para a imagem em questão, ou seja, ao se pretender que uma imagem tenha uma taxa de compressão maior atribui-se um valor alto de qualidade (valores superiores a 1), obtendo-se assim uma imagem que ocupa pouca quantidade de memória para

armazenamento mas que tem uma baixa qualidade para visualização. O valor de qualidade escolhido é multiplicado por cada elemento da tabela de quantização mas actuando na razão inversa, ou seja, quanto maior o valor para a qualidade mais degradada será a imagem resultante e vice-versa. Sendo que deste modo se tem um compromisso taxa de compressão/qualidade. No sistema implementado considerou-se, para valores de qualidade, a gama de valores inteiros de 1 a 6.

3.1.3 Escolha do cenário

Uma vez que os hábitos de utilização do Centro de Cálculo variam ao longo do dia foi necessário definir cenários que têm em conta esses mesmos hábitos.

Para a escolha do cenário são tidos em conta os seguintes aspectos: câmara; número de blocos de dimensão 8×8 , a hora do dia e a data.

Foi criado um cenário que é activado durante o horário nocturno, ou durante o fim-de-semana. Este cenário tem a especificidade de ser igual para todas as câmaras e apresenta um factor de qualidade de 1, independentemente do número de blocos detectados. O valor de limiar é também alterado passando para o valor 1. Deste modo procura-se ter um detector bastante sensível a qualquer movimento uma vez que quando nos encontramos neste cenário não é suposto encontrar-se alguém dentro do Centro de Cálculo, o que significa que qualquer detecção de movimento pode significar estar-se perante uma possível intrusão.

Uma vez que a zona observada por cada câmara é diferente, e tendo em conta a região de interesse que foi definida para cada câmara é necessário saber quantos blocos de 8×8 temos em cada imagem de cada câmara. Na Tabela 1 observa-se o número de pixels, bem como o número de blocos, associados a cada câmara para a sua região de interesse.

	Número de pixels da região de interesse	Número de blocos da região de interesse
Câmara 1	220011	3437
Câmara 2	209336	3271
Câmara 3	244155	3814

Tabela 1 – Número de pixels e de blocos da região de interesse para cada câmara

Foi definido para as 3 câmaras que, caso o número de blocos que indicam a detecção de movimento seja inferior a 1% do número total de blocos existentes na região de interesse, a imagem será descartada. Na Tabela 2 pode-se observar os valores relativos para cada câmara. Estes valores são arredondados por defeito.

Câmara	Valor relativo a 1% dos blocos totais
1	34
2	32
3	38

Tabela 2 – Valor relativo a 1% dos blocos totais para cada câmara

Foram definidos três cenários, para o número de blocos, para cada câmara em que os valores correspondentes a cada cenário foram definidos tendo em conta a especificidade da zona observada cada câmara. Os cenários têm sempre em conta o número de blocos que identificam a região da imagem onde se detectou movimento.

Foram definidos para cada câmara os seguintes cenários:

- Pouco movimento,
- Movimento normal,
- Muito movimento,

Para a câmara 1, os valores relativos a cada cenário foram definidos tal como se apresenta na Tabela 3.

Cenário	Número de blocos	Factor de qualidade
Pouco movimento	[34,300]	6
Movimento normal	[300, 1300]	2
Muito movimento	[1300,3437]	1

Tabela 3 – Valores relativos aos cenários da câmara 1

Para a câmara 1, considera-se pouco movimento quando o número de blocos detectados representa 10% do número total de blocos existentes na região de interesse. Considera-se movimento normal quando o número de blocos detectados representa entre 10% a 40% e considera-se muito movimento quando este representa pelo menos 40% de blocos detectados.

Quando a pessoa entra no Centro de Cálculo, aproxima-se da câmara levando a que mais blocos relativos à detecção de movimento sejam considerados, levando a um aumento na qualidade da imagem. Como se pretende um sistema que vigie a intrusão bem como os actos dessa intrusão, considera-se que a câmara 1 tem como objectivo a identificação da pessoa quando ela entra, sendo que as outras câmaras servirão para seguir os actos dessa mesma pessoa uma vez que ela já foi identificada.

Para a câmara 2, os valores relativos a cada cenário foram definidos tal como se apresenta na Tabela 4.

Cenário	Número de blocos	Factor de qualidade
Pouco movimento	[32,350]	6
Movimento normal	[350, 2000]	3
Muito movimento	[2000,3271]	1

Tabela 4 – Valores relativos aos cenários da câmara 2

Uma vez que a câmara 2 tem como raio de acção o corredor do Centro do Cálculo esta tem uma especificidade diferente da câmara 1. O corredor é não só um local de passagem mas também um local onde normalmente se encontra muita gente parada. Daí se considerar como pouco movimento quando o número de blocos detectados representa 10% do número total de blocos existentes na região de interesse.

Considera-se movimento normal quando o número de blocos detectados representa entre 10 a 60% e considera-se muito movimento quando este representa pelo menos 60% de blocos detectados.

Para a câmara 3, os valores relativos a cada cenário foram definidos com se apresenta na Tabela 5.

Cenário	Número de blocos	Factor de qualidade
Pouco movimento	[34,340]	6
Movimento normal	[300, 2200]	3
Muito movimento	[2200,3814]	1

Tabela 5 – Valores relativos aos cenários da câmara 3

Uma vez que a câmara 3 tem como raio de acção o corredor do Centro do Cálculo esta tem uma especificidade igual ao da câmara 2, daí terem sido considerados os mesmos valores para as percentagens de blocos detectados. Estes valores foram

considerados após vários testes em vídeos referentes ao Centro de Cálculo, através da visualização dos mesmos com diferentes valores de qualidade e de limiar.

3.2 Codificação

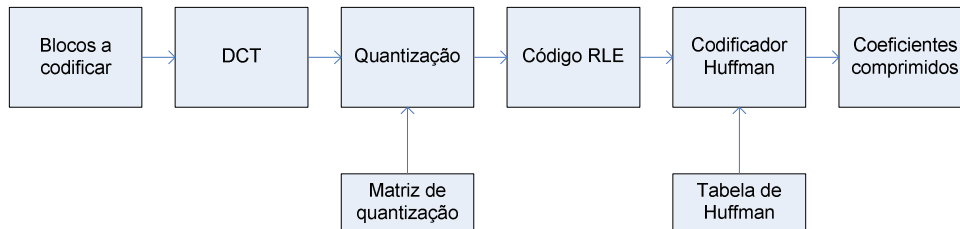


Figura 15 – Processo de codificação

Na codificação, a DCT é aplicada em blocos de 8×8 apenas aos blocos provenientes da detecção de movimento, sendo este processo efectuado a cada um dos blocos. Na quantização a componente DC de cada bloco é guardada e é inserido o valor zero. Todos os blocos, de uma imagem, são transformados em 1×64 e armazenados sequencialmente, deste modo procura-se maximizar as sequências de caracteres iguais. Para todos os valores dos coeficientes são guardados o número de repetições bem como o valor do respectivo coeficiente. Deste modo tem-se um código otimizado para a aplicação do Huffman.

A Figura 16 mostra um exemplo do formato do código RLE.

0	0	0	1	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---

A – Coeficientes

3	0	2	1	5	0
---	---	---	---	---	---

B – Número de repetições de cada coeficiente

Figura 16 – Formato do código RLE

Na Figura 16–A apresentam-se os coeficientes, e na Figura 16–B o número de repetições associado a cada coeficiente e o respectivo coeficiente.

Em seguida é criado um código de Huffman [2][3] diferente para cada imagem e é efectuada a compressão dos dados. O código de Huffman, usa palavras de comprimento uma vez que se adapta às características da imagem, ou seja, às características dos coeficientes a codificar. É feita uma análise ao valor máximo e mínimo existente em todos os blocos, bem como do número de vezes que cada valor se

repete, a partir dos quais se constrói o código de Huffman para cada imagem. A Figura 17 mostra um exemplo do formato da tabela do código de Huffman.

-6	[0 1 1 0 0 0 0 1]
-5	[0 1 1 0 0 0 1]
-4	[0 1 1 0 1 0]
-3	[0 1 1 1 1]
-2	[0 1 0 1]
-1	[1 1]
0	[0 0]
1	[1 0]
2	[0 1 0 0]
3	[0 1 1 1 0]
4	[0 1 1 0 1 1]

Figura 17 – Exemplo do formato da tabela do código de Huffman

A explicação detalhada do bloco da DCT e da Quantização da Figura 15 já foi formulada na secção 2.1

3.3 Armazenamento

O armazenamento de toda a informação relativa a uma imagem é realizado num ficheiro. Como se pode observar na Figura 18, neste ficheiro são armazenadas a data /hora da imagem, as coordenadas relativas aos blocos de dimensão 8×8 que identificam as regiões da imagem onde foi detectado movimento, os coeficientes comprimidos, as componentes DC de cada bloco que foi codificado, a tabela de Huffman respeitante aos blocos, o nível de qualidade com que os blocos foram codificados, a informação relativa às repetições provenientes do código RLE e a câmara a que diz respeito a imagem recolhida.

Data/hora	Tabela de Huffman (dicionário)	Coefficientes comprimidos	Repetições do código RLE	Qualidade	Componentes DC	Coordenadas	Câmara

Figura 18 – Armazenamento da informação relativa a uma imagem

É possível observar na Figura 19, detalhadamente como é realizado o armazenamento:

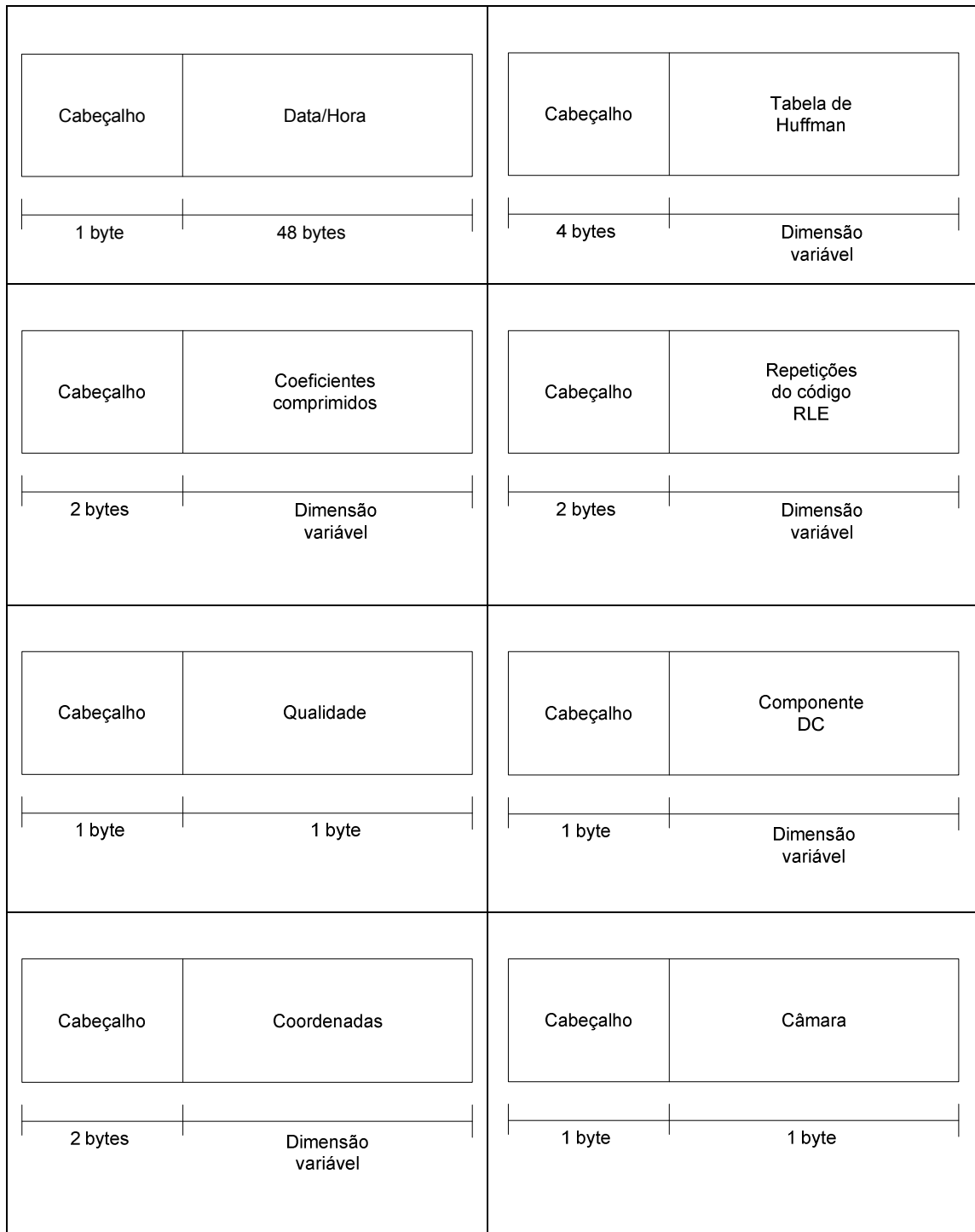


Figura 19 – Armazenamento em detalhe

3.4 Descodificação

A Figura 20 mostra o diagrama de blocos do decodificador.

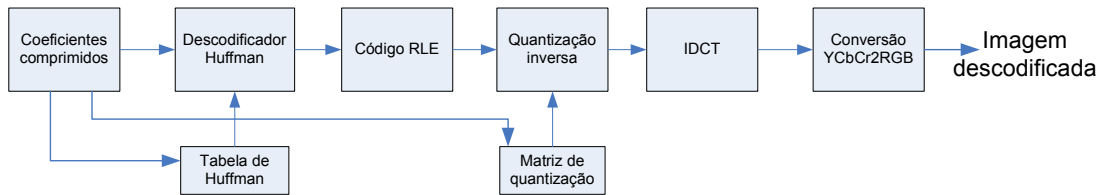


Figura 20 – Diagrama de blocos da decodificação

Na decodificação, é aplicado o decodificador de Huffman com a tabela respeitante aos blocos. Na quantização inversa a matriz de quantização é multiplicada pelo nível de qualidade com que os blocos foram guardados e é inserida a componente DC de cada bloco. As matrizes de quantização utilizadas no processo da decodificação são as mesmas que foram utilizadas para a codificação, e que se encontram representadas na Figura 4, havendo uma matriz de quantização para a componente Y e outra matriz para as componentes Cb e Cr. Em seguida é aplicada a IDCT a cada bloco de 8×8 que foi codificado.

Após a IDCT os blocos são colocados na imagem de fundo respectiva, e na coordenada respectiva. É efectuada uma mudança no espaço de cor da imagem de YCbCr para RGB e obtém-se a imagem decodificada.

Capítulo 4 – Resultados Experimentais

O sistema implementado em MATLAB foi testado com vídeos adquiridos no corredor do Centro de Cálculo pelas 3 câmaras instaladas no local. Estes vídeos têm resolução de 480×640 e com diferentes valores de número de imagens por segundo (1 e 5). Os testes realizados incidiram sobre a detecção de movimento, para escolha do método a utilizar em cada câmara individualmente. O detector de movimento é um bloco crítico no sistema implementado uma vez que é necessário garantir a detecção adequada de intrusões. Realizaram-se testes para determinar o valor de limiar a utilizar para cada uma das câmaras uma vez que se pretende um detector de movimento sensível apenas aos movimentos de pessoas e não a movimentos provocados pela variação da luminosidade, tentando deste modo encontrar o valor óptimo de limiar para cada câmara, o que resultará após a codificação num espaço menor de armazenamento da informação resultante do sistema de vigilância mas garantindo sempre que a identificação e os movimentos das pessoas são detectados. O sistema implementado foi testado comparando-o com outras técnicas de modo a avaliar o desempenho deste. Foi também avaliado o tempo de execução do algoritmo proposto para a codificação/descodificação das imagens. Os resultados experimentais obtidos são apresentados nas próximas secções.

4.1 Detecção de movimento

Testa-se a detecção de movimento numa situação crítica, ou seja quando existe movimento pouco significativo, do detector para o método A, descrito na secção 3.1.1.

A avaliação da detecção de movimento é feita através da imagem D de cada. Na Figura 21–A,B observam-se duas imagens consecutivas onde existe uma pessoa que efectua um movimento pouco significativo entre as duas imagens.

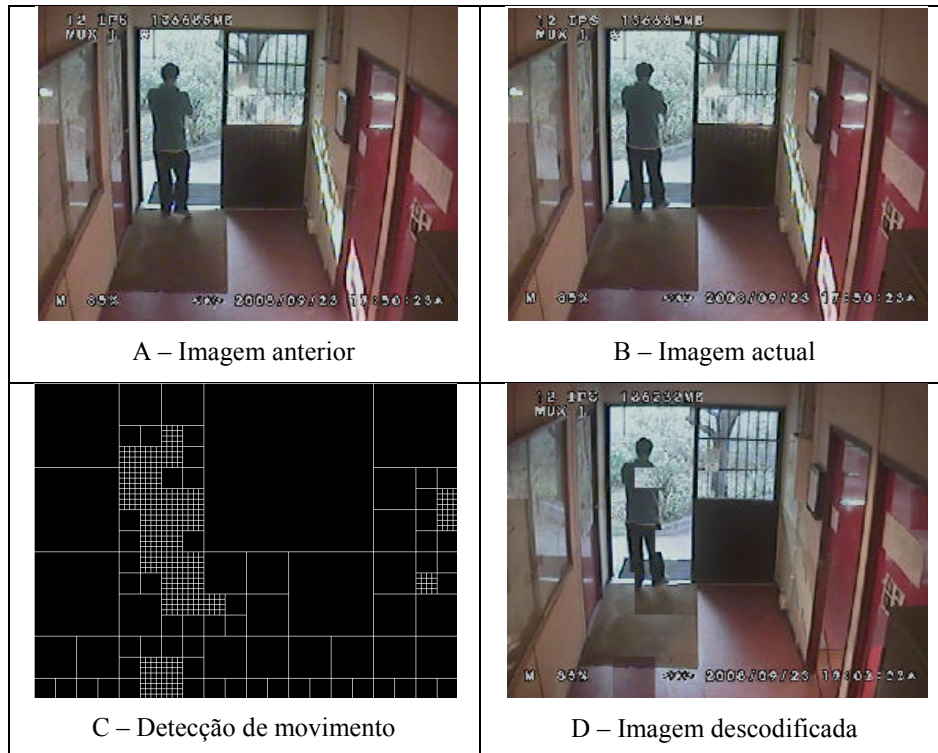


Figura 21 – Detecção de movimento usando o método A e imagem decodificada na câmara 1

Existe uma falha na detecção de movimento. Tal deve-se ao facto de a pessoa ter-se movido lentamente, logo o detector de movimento apresenta lacunas nos pontos onde não foi detectado qualquer movimento, apresentando assim uma imagem decodificada em que não é visível a totalidade da pessoa. De outra forma, parte da imagem da pessoa foi erradamente considerada como imagem de fundo.

Efectua-se o mesmo teste, alterando o detector de movimento para o método B, descrito na secção 3.1.1.

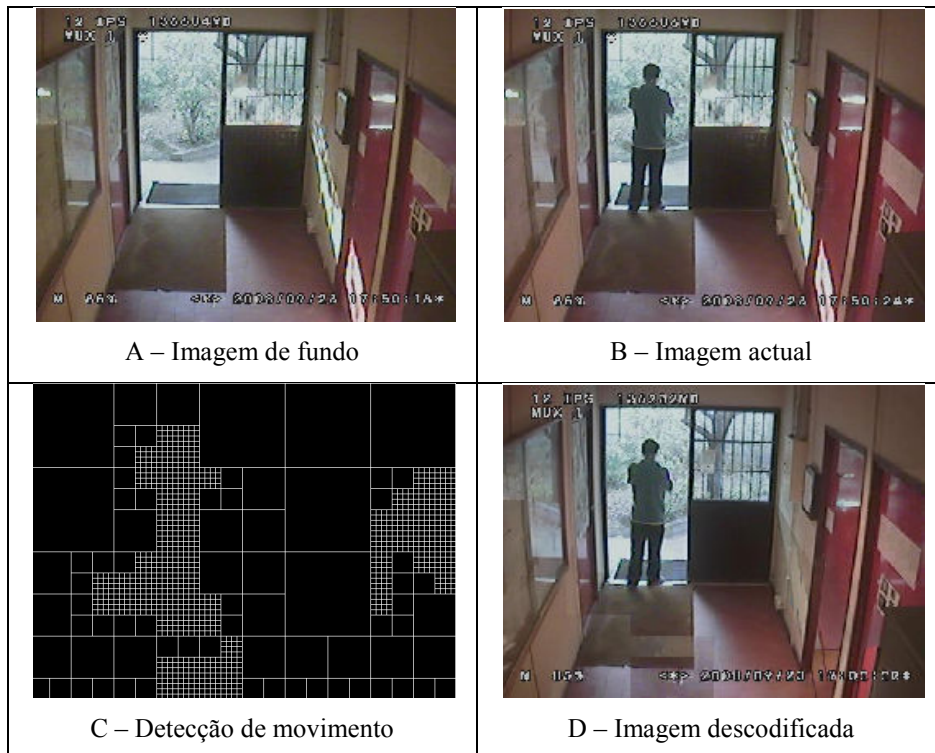


Figura 22 – Deteccção de movimento usando o método B e imagem descodificada na câmara 1

Verifica-se que é detectado a totalidade do movimento. Em comparação com o método anterior, este apresenta um maior número de blocos de dimensão 8×8 ; tal deve-se à detecção do movimento da pessoa e à alteração da luminosidade, dado que a imagem de fundo foi actualizada quando não havia qualquer movimento na imagem.

Efectuou-se ainda outro teste, para a mesma câmara e para os dois métodos de detecção de movimento, em que existe um movimento de passagem pela porta do Centro de Cálculo, tal como se apresenta na Figura 23.

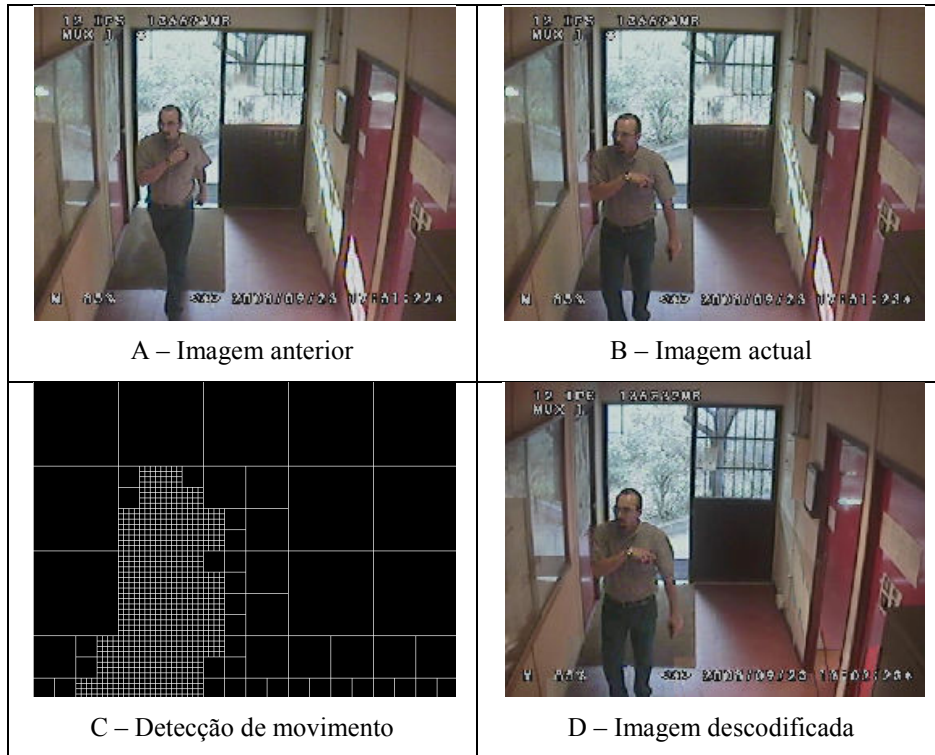


Figura 23 – Detecção de movimento usando o método A e imagem decodificada na câmara 1

Obtém-se uma imagem decodificada em que é visível a totalidade da pessoa, não existindo quaisquer lacunas.

Efectua-se o mesmo teste com o detector de movimento para o método B.

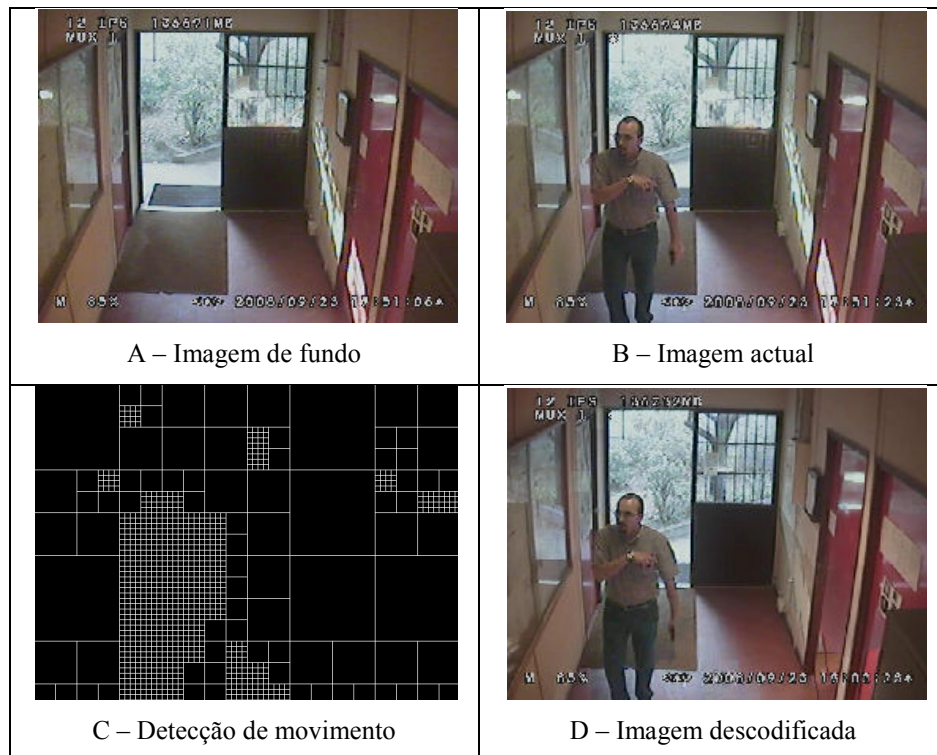


Figura 24 – Detecção de movimento usando o método B e imagem decodificada na câmara 1

É detectada a totalidade do movimento existente, sem lacunas na decodificação da imagem. Comparando os dois métodos é possível observar através da Figura 23 – C e da Figura 24 – C que a detecção de movimento, aquando de um movimento de passagem, apresenta resultados bastantes semelhantes para os dois métodos.

Para a câmara 1 é extremamente importante que seja possível, após a decodificação da imagem, a identificação de todos os movimentos e intervenientes. Desse modo optou-se pelo método B para a câmara 1, uma vez que não apresenta falhas aquando da existência de pouco ou nenhum movimento, em detrimento do espaço necessário para o armazenamento dos coeficientes, e apresenta resultados semelhantes ao método A quando existe um movimento de passagem pela porta do Centro de Cálculo.

Efectua-se o mesmo tipo de teste mas desta vez para a câmara 2.

Na Figura 25–A,B temos duas imagens consecutivas onde o movimento das duas pessoas presentes é pouco significativo.

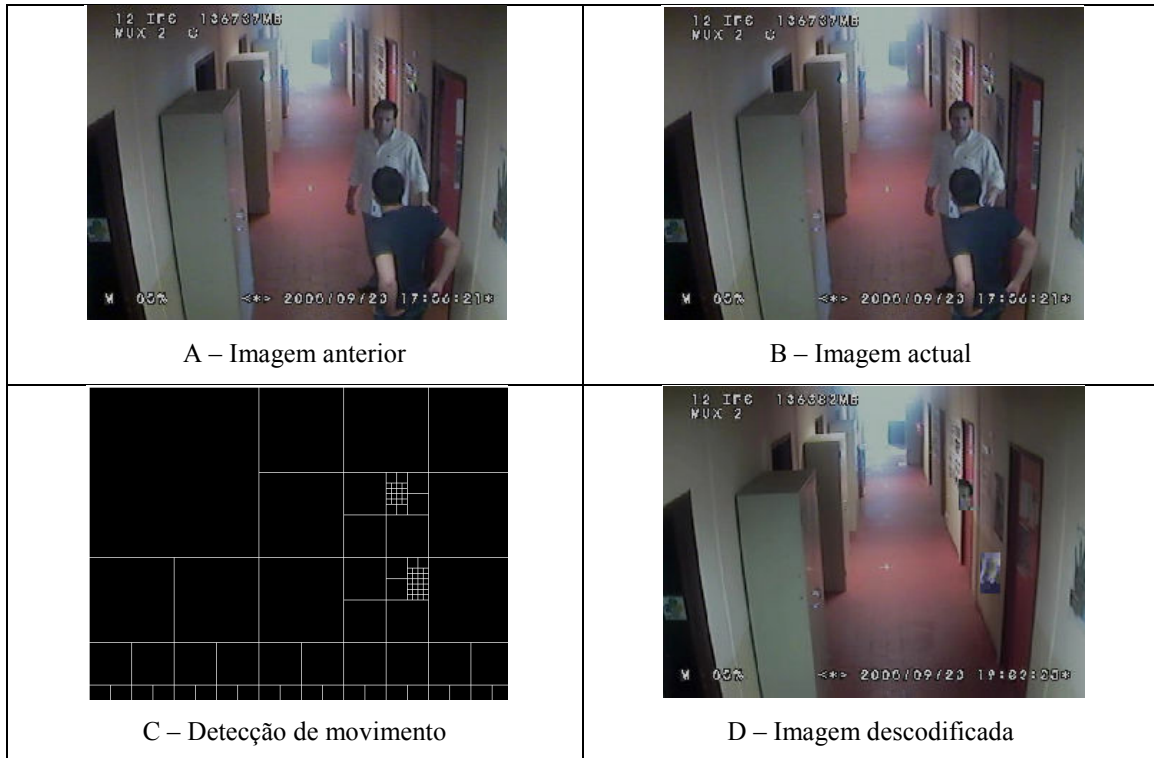


Figura 25 – Detecção de movimento usando o método A e imagem descodificada na câmara 2

Existe uma falha na detecção de movimento, tal deve-se ao facto de as pessoas que se encontram junto à porta pouco ou nada se terem movido, logo como não é detectado movimento quando as pessoas se encontram paradas, estas desaparecem da imagem.

Efectua-se o mesmo teste mas alterando o detector de movimento para o método

B.

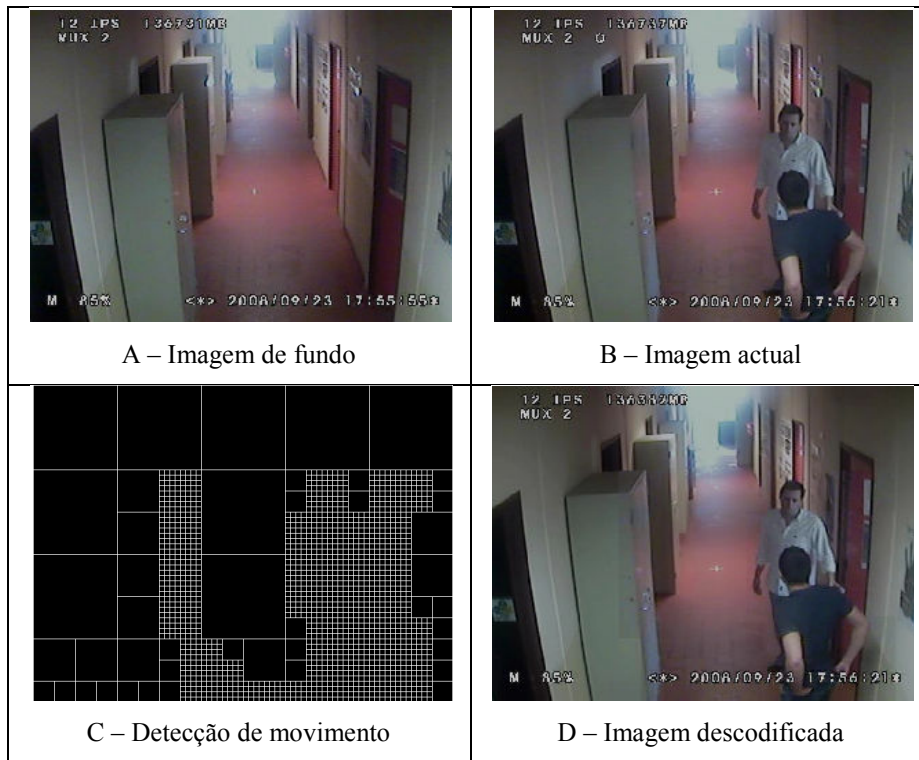


Figura 26 – Detecção de movimento usando o método B e imagem decodificada na câmara 2

Verifica-se que a totalidade do movimento é detectada. Neste método obtém-se um maior número de blocos de dimensão 8×8 em comparação com o método anterior.

Efectuou-se outro teste, para a mesma câmara e para os dois métodos de detecção de movimento, em que existe um movimento de passagem pelo corredor.

Na Figura 27–A,B mostram-se duas imagens consecutivas onde existe movimento significativo ao longo do corredor.

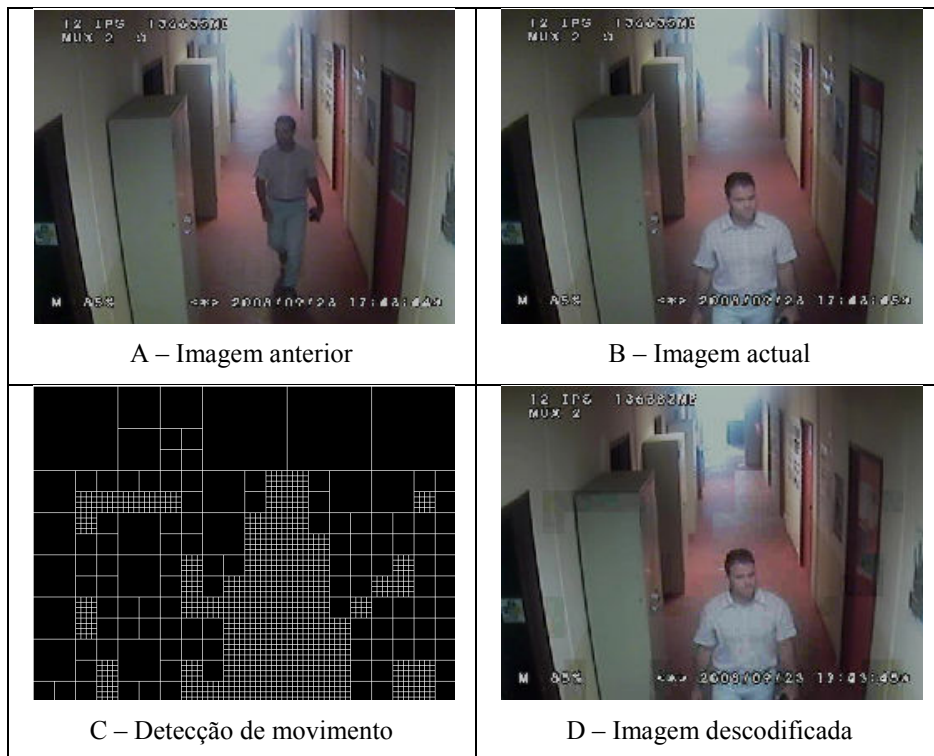


Figura 27 – Detecção de movimento usando o método A e imagem decodificada na câmara 2

Como se verifica na Figura 27, é detectado o movimento da pessoa ao longo do corredor, ao contrário da Figura 25, uma vez que existiu movimento significativo entre as imagens a detecção de movimento apresenta bons resultados.

Efectuando o mesmo teste mas para o método B, apresentam-se os resultados na Figura 28.

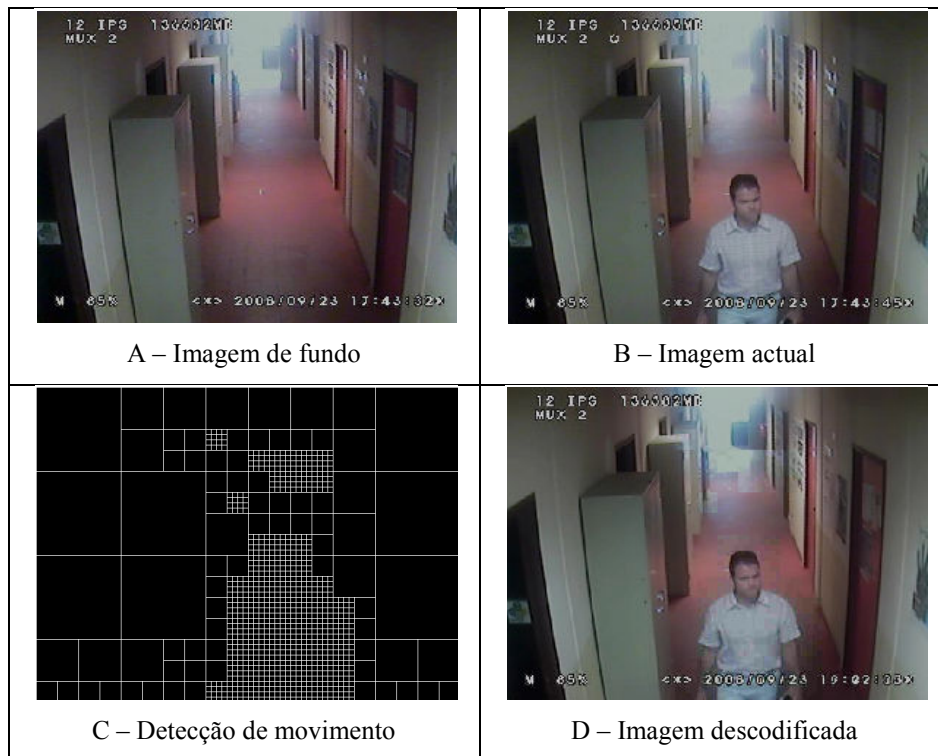


Figura 28 – Detecção de movimento usando o método B e imagem decodificada na câmara 2

É detectada a totalidade do movimento existente. Em comparação com o outro método este apresenta um menor número de blocos de dimensão 8×8 . Ainda assim a escolha para a detecção de movimento, na câmara 2, recaiu sobre o método A uma vez que a zona observada pela câmara é o corredor do Centro de Cálculo pretende-se detectar apenas os movimentos existentes, para deste modo vigiar o corredor, e não quando as pessoas se encontram paradas no corredor. Deste modo é possível reduzir no armazenamento uma vez que apenas aquando da existência de movimento esta informação é guardada. Por último, efectua-se o mesmo tipo de teste para a câmara 3.

Na Figura 29–A,B observam-se duas imagens consecutivas com movimento de passagem ao longo do corredor.

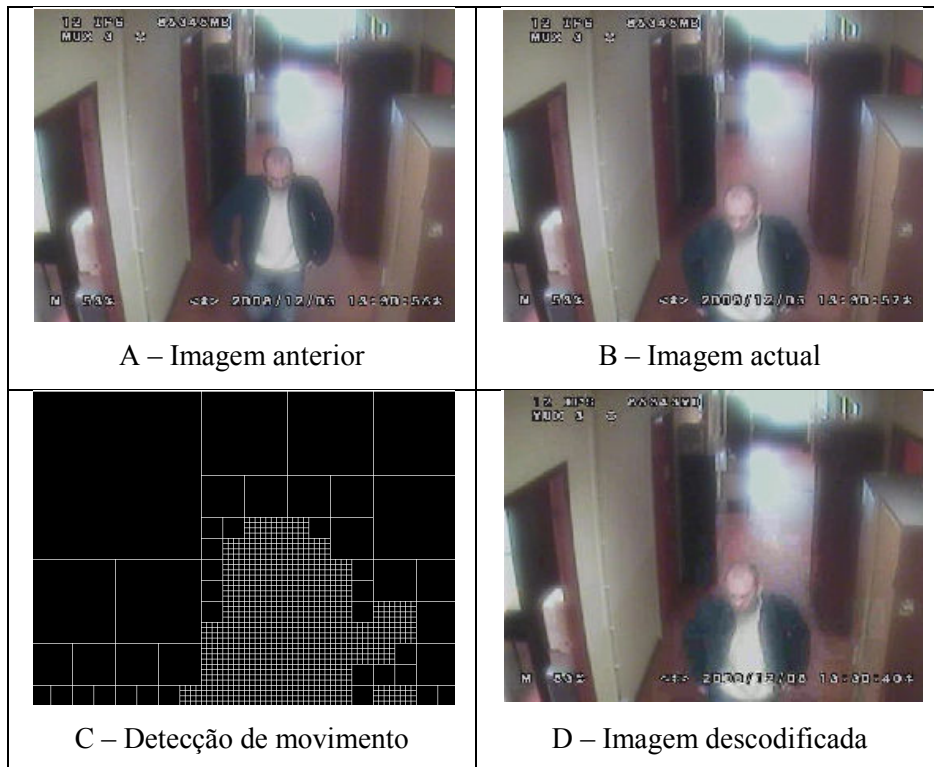


Figura 29 – Detecção de movimento usando o método A e imagem descodificada na câmara 3

É detectado todo o movimento existente entre as duas imagens consecutivas. Uma vez que existe movimento de passagem, e este é significativo, o movimento da pessoa é todo detectado permitindo deste modo seguir o movimento desta, tal como a sua identificação.

Efectua-se o mesmo teste mas alterando o detector de movimento para o método B.

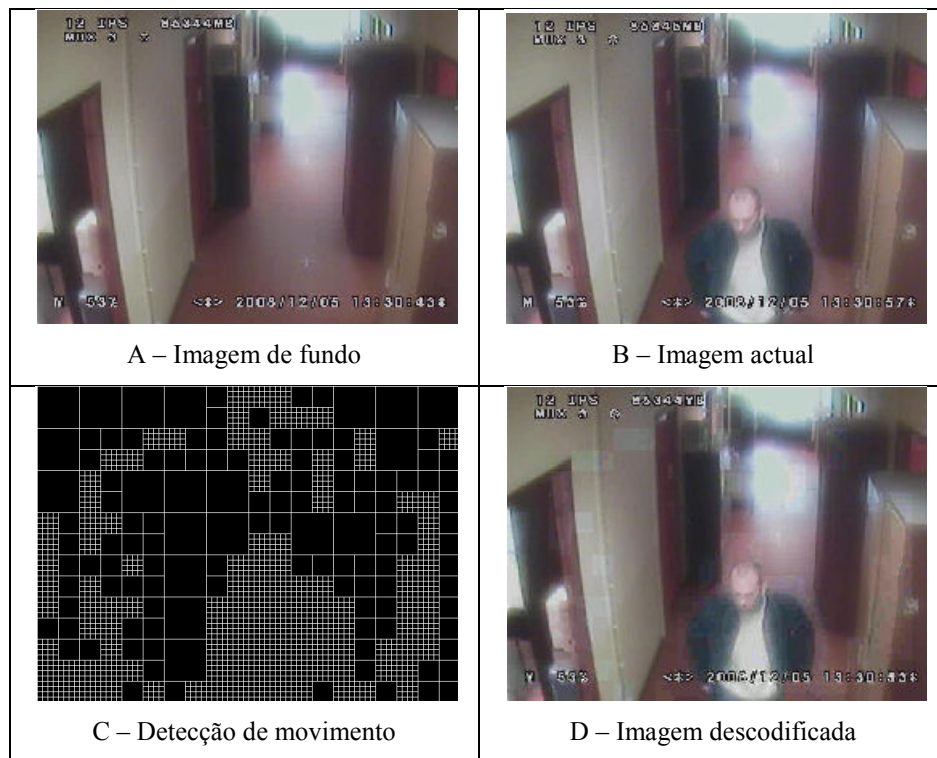


Figura 30 – Detecção de movimento usando o método B e imagem descodificada na câmara 3

Na Figura 30-C existe um maior número de blocos detectados como movimento. Tal deve-se ao facto de a imagem de fundo ter sido actualizada quando não havia movimento na imagem, mas entretanto já se passou algum tempo e as condições de luminosidade alteraram-se. Tal como para a câmara 2, optou-se pela detecção de movimento usando o método A pelas razões já enunciadas na câmara 2.

4.2 Influência de diferentes valores para o limiar

Para avaliação da influência do valor de limiar, estando este definido na secção 3.2.1, efectuam-se alguns testes para valores do limiar diferentes de 5 (valor considerado anteriormente para a câmara 2 e 3).

Considerando as imagens da Figura 31 respeitantes à câmara 2, obtêm-se os resultados apresentados na Figura 32.



Figura 31 – Imagens consecutivas para detecção de movimento usando diferentes valores do limiar

Valor do limiar	Imagem resultante da detecção de movimento	Imagem decodificada
1		
5		

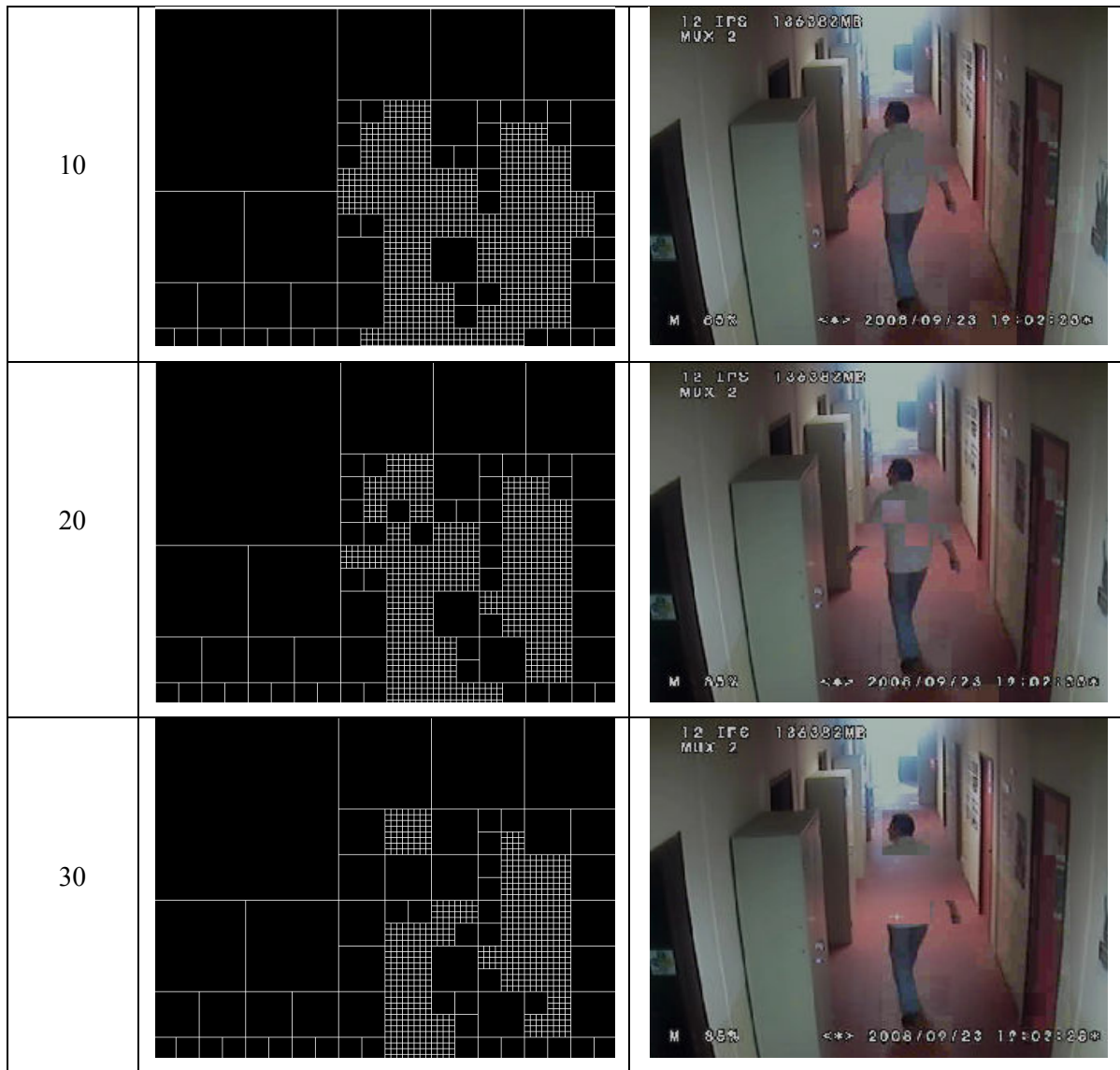


Figura 32 – Detecção de movimento para diferentes valores do limiar para a câmara 2

À medida que o valor do limiar aumenta o detector de movimento torna-se menos sensível. Para valores superiores a 10 existe uma degradação acentuada na detecção de movimento, e a imagem descodificada apresenta bastantes lacunas, podendo impossibilitar a identificação da pessoa bem como o seu movimento. Para valores inferiores a 5, a detecção de movimento é muito sensível aos movimentos existentes entre as duas imagens, o que resulta num maior número de blocos de dimensão 8×8 . Para o valor de 5 temos uma boa detecção de movimento e, comparando com a imagem da detecção de movimento para o valor 1, observa-se uma redução elevada do número de blocos enquanto na imagem descodificada para estes dois valores as diferenças não são muito significativas.

Os valores abaixo de 5 apresentam visualmente uma boa imagem, mas em que devido ao elevado número de blocos detectados, o tempo de processamento necessário para codificar os blocos aumenta e a taxa de compressão diminui pois existem mais blocos a ser armazenados. Os valores acima de 10 apresentam uma imagem que visualmente apresenta muitas falhas. A escolha para o valor de limiar recai sobre o valor 5 uma vez que se obtém uma imagem onde são perceptíveis os movimentos existentes, e a identificação da pessoa. Com este valor, consegue-se diminuir o tempo de processamento e aumentar a taxa de compressão e obter imagens com qualidade aceitável.

4.3 Comparação do sistema implementado com outras técnicas

Apresenta-se uma comparação do sistema implementado com imagens em formato JPEG e em BITMAP. Para se efectuar a comparação com diferentes formatos, calculou-se a SNR (*Signal-to-Noise ratio*) e a PSNR (*Peak Signal-to-Noise ratio*), de modo a avaliar a distorção nas imagens decodificadas. A SNR é definida como

$$SNR = 10 * \log_{10} \left(\frac{P_S}{P_N} \right) \quad (10)$$

em que

P_S é a potência da imagem original,

P_N é a potência do ruído, sendo este definido como

$$P_N = \frac{1}{m * n * 3} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2$$

em que

$I(i, j)$ representa a imagem original,

$K(i, j)$ representa a imagem obtida após a codificação e decodificação,

$$PSNR = 10 * \log_{10} \left(\frac{MAX^2_{Img}}{P_N} \right) \quad (11)$$

em que

$MAX = 255$, uma vez que nos encontramos a representar a 8 *bits* cada componente.

Calculou-se a percentagem removida entre o tamanho da imagem original e o da codificada.

$$Taxa_{compress\tilde{a}o} = \left(1 - \frac{Dim_img_final}{Dim_img_inicial}\right) * 100. \quad (12)$$

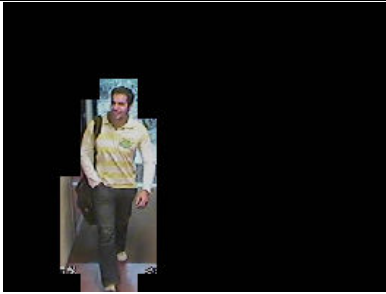
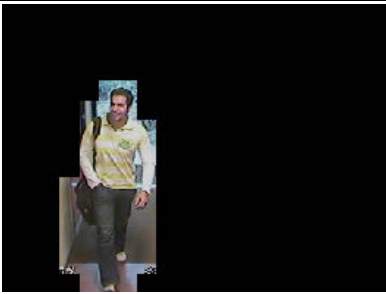
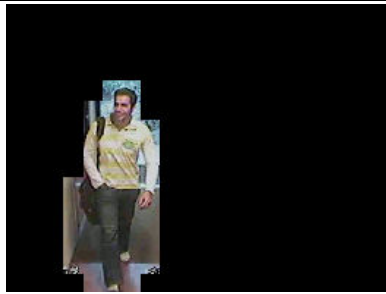








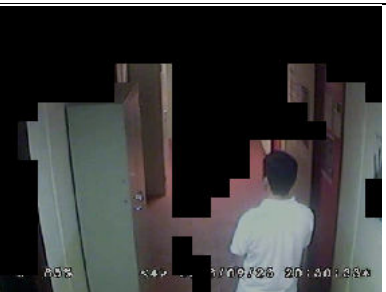
Formato	BITMAP	JPEG	Sistema implementado
	 A	 B	 C
	 A	 B	 C
	 A	 B	 C
	 A	 B	 C

Figura 33 – Sub-Imagens obtidas usando 3 técnicas diferentes

Na Figura 33 tem-se a mesma sub-imagem usando três técnicas diferentes. As sub-imagens dizem respeito à detecção de movimento entre duas imagens, sendo a detecção de movimento efectuada consoante a câmara, data/hora e o cenário. Para a detecção de movimento na linha 1 é utilizado o método B, na linha 2 e 3 o método A e na linha 4, em virtude de ser considerado horário nocturno, é utilizado o método B.

Na Figura 33 (linhas 1 e 2) está-se perante o cenário em que existe movimento significativo levando a que na imagem descodificada (Figura 33, linha 1-C) o movimento seja todo detectado, permitindo a identificação da pessoa e o seu movimento.

Ainda na mesma figura (linha 3), está-se perante o cenário em que apesar de haver muitas pessoas na imagem o movimento dessas é bastante reduzido, levando a que se obtenha uma imagem descodificada (Figura 33, linha 3-C) em que a detecção de movimento apresenta lacunas levando ao desaparecimento de pessoas que estão na imagem. Na Figura 33 (linha 4) está-se no cenário nocturno, e como é possível observar (Figura 33, linha 4-C) o movimento é totalmente detectado e a imagem descodificada apresenta-se visualmente boa. Uma vez que se encontra no horário nocturno a detecção de movimento é realizada através do método B, daí que exista detecção de movimento em regiões que aparentemente não registam quaisquer tipos de movimento; tal deve-se ao facto de a imagem de fundo ter sido actualizada quando não foi detectado movimento, não estando totalmente actualizada daí resultarem diferenças na imagem actual com a imagem de fundo que são consideradas como movimento. Para se aferir sobre o tamanho das imagens obtidas pelo sistema implementado este foi dividido em duas fases: I e II. A Fase I representa a dimensão total das variáveis antes de estas serem armazenadas em memória, e a Fase II representa a dimensão total do ficheiro armazenado em memória. Esta divisão em duas fases resulta do facto de ao se efectuar o armazenamento do ficheiro em memória este ser alvo de uma compressão adicional, tirando-se assim partido de uma opção que é disponibilizada pelo MATLAB, tendo deste modo a Fase I e a Fase II dimensões diferentes.

BITMAP	JPEG	Sistema Implementado	
		Fase I	Fase II
135KB	10,1KB	15,8KB	7,37KB
204KB	10,4KB	13,3KB	4,56KB
225KB	9,60KB	13,5 KB	3,99KB
486KB	20,2KB	42,7KB	18,4KB

Tabela 6 – Tamanho das imagens com 3 técnicas diferentes

Analisando a Tabela 6 é possível concluir sobre os diferentes tipos de tamanho de imagem que se obtêm. Os tamanhos apresentados dizem respeito apenas aos blocos onde foi detectado movimento e não ao tamanho das imagens apresentadas na Figura 33 uma vez que estas apenas servem para visualização. Para se obter os tamanhos dos blocos com o BITMAP e o JPEG, os blocos provenientes da detecção de movimento foram armazenados e guardados com a respectiva extensão.

Comparando o tamanho da imagem BITMAP em relação ao JPEG e ao Sistema Implementado Fase II. Como se observa existe uma diferença bastante significativa do BITMAP para JPEG. Em relação ao tamanho das imagens para o Sistema Implementado Fase II este apresenta menores valores, comparando com as outras duas técnicas, mas como se observa na imagem obtida no Sistema Implementado Fase II, correspondente à linha 3, muitas vezes esta redução no tamanho da imagem leva a que esta se apresente visualmente bastante degradada. Observa-se que a diferença entre a Fase I e a Fase II do Sistema Implementado apresenta uma redução significativa, tal deve-se à compressão dos dados aquando do seu armazenamento em memória.

JPEG	Sistema Implementado	
	Fase I	Fase II
BITMAP→JPEG: 92,51%	BITMAP→Si: 89,03% JPEG→Si: -56,43%	BITMAP→Si: 94,54% JPEG→Si: 27,02% Fase I →Fase II: 53,35%
BITMAP→JPEG: 95,09%	BITMAP→Si: 93,97% JPEG→Si: -27,88%	BITMAP→Si: 97,76% JPEG→Si: 56,15% Fase I →Fase II: 65,71%
BITMAP→JPEG: 95,73%	BITMAP→Si: 94% JPEG→Si: -40,62%	BITMAP→Si: 98,22% JPEG→Si: 58,43% Fase I →Fase II: 28,88%
BITMAP→JPEG: 95,84%	BITMAP→Si: 91,21% JPEG→Si: -111,18%	BITMAP→Si: 96,21% JPEG→Si: 8,91% Fase I →Fase II: 56,90%

Tabela 7 – Taxa de compressão para as 3 técnicas diferentes

Analisando Tabela 7 é possível concluir sobre a taxa de compressão que se obtém, comparando o tamanho da imagem original, BITMAP, em relação ao JPEG e em relação ao Sistema Implementado. Como se observa o Sistema Implementado Fase II obtém uma elevada taxa de compressão em relação à imagem em formato BITMAP, sempre acima dos 94%, sendo superior às taxas de compressão obtidas de BITMAP para JPEG, para os casos testados. Em relação ao JPEG o Sistema Implementado Fase II apresenta uma taxa de compressão acima dos 50% para as imagens que se referem ao corredor. Para a imagem referente à porta do Centro de Cálculo a taxa de compressão é menor uma vez que a imagem foi codificada com factor de qualidade melhor. Para a imagem referente ao cenário da noite observa-se no Sistema Implementado Fase II a taxa de compressão, comparando com o JPEG, é reduzida uma vez que são considerados um elevado número de blocos e estes são codificados com menor perda de informação, ou seja, com o valor 1 de qualidade. Utilizando as imagens apresentadas na

Figura 33 calcula-se o SNR e o PSNR, tendo em conta apenas os blocos que foram codificados.

JPEG	Sistema Implementado
BITMAP→JPEG: 18,85dB	BITMAP→Si: 10,46dB JPEG→Si: 9,79dB
BITMAP→JPEG: 20,23dB	BITMAP→Si: 11,18dB JPEG→Si: 10,93dB
BITMAP→JPEG: 21,00dB	BITMAP→Si: 12,14dB JPEG→Si: 11,93dB
BITMAP→JPEG: 20,91dB	BITMAP→Si: 16,62dB JPEG→Si: 15,03 dB

Tabela 8 – Cálculo da SNR para as 3 técnicas diferentes

JPEG	Sistema Implementado
BITMAP→JPEG: 51,18dB	BITMAP→Si: 42,79dB JPEG→Si: 42,12dB
BITMAP→JPEG: 50,73dB	BITMAP→Si: 41,70dB JPEG→Si: 41,46dB
BITMAP→JPEG: 51,09dB	BITMAP→Si: 42,23dB JPEG→Si: 42,02dB
BITMAP→JPEG: 47,71dB	BITMAP→Si: 42,41dB JPEG→Si: 41,83dB

Tabela 9 – Cálculo da PSNR para as 3 técnicas diferentes

Através da análise da Tabela 8 e Tabela 9, demonstra-se que as imagens 2-C e 3-C da Figura 33 são aquelas que apresentam maior distorção em relação à imagem original, uma vez que são imagens referentes ao corredor logo foram codificadas com pior factor de qualidade. Verifica-se nos resultados obtidos pela imagem 4-C na Figura 33 que esta é a que obtém menos distorção em relação à original, ou seja, tem melhor qualidade. Tal deve-se ao facto de ter sido codificada em horário nocturno o que permitiu obter uma imagem de boa qualidade.

Analisando um vídeo captado pela câmara 3 com resolução de 480×640 , composto por 17 imagens, com uma imagem por segundo e ocupa em memória 7,47MB. Neste vídeo existe sempre movimento ainda que reduzido uma vez que são duas pessoas que se encontram a falar no corredor. Este vídeo ao ser codificado pelo Sistema Implementado passa a ocupar em memória 26,8KB, acrescido de uma imagem de fundo, que é necessária para a visualização das imagens descodificadas, e que ocupa em memória 24,2KB. O espaço total ocupado em memória é de 51KB obtendo-se uma taxa de compressão de 99,31%.

Na Figura 34 mostra-se algumas imagens, não consecutivas, decodificadas do vídeo.



Figura 34 – Imagens decodificadas do vídeo da câmara 3

Uma vez que existe pouco movimento, torna-se mais uma vez perceptível o desaparecimento das pessoas na imagem.

Analisando outro vídeo captado pela câmara 2 com resolução de 480×640 , composto por 74 imagens, com uma imagem por segundo e ocupa em memória 32,5MB. Neste vídeo não existe praticamente movimento à exceção de uma pessoa que se desloca ao longo do corredor até desaparecer. O vídeo ao ser codificado pelo sistema implementado passa a ocupar em memória 26KB, acrescido de uma imagem de fundo, que é necessária para a visualização das imagens decodificadas, e que ocupa em memória 30,8KB. O espaço total ocupado em memória é de 56,8KB obtendo-se uma taxa de compressão de de 99,82%.



Figura 35 – Imagens decodificadas do vídeo da câmara 2

Quando a pessoa se encontra ao fundo do corredor a imagem decodificada encontra-se bastante degradada. À medida que a pessoa se vai deslocando e aproximando da câmara a qualidade da imagem aumenta.

Apresenta-se uma sequência de imagens de modo a se comparar as imagens decodificadas com as imagens obtidas pela câmara.

A Figura 36 apresenta uma sequência de imagens consecutivas captadas pela câmara 1, começando numa imagem onde não existe praticamente movimento até à entrada de uma pessoa no Centro de Cálculo. É um vídeo AVI com uma resolução de 480×640 e foram analisadas 18 imagens desse vídeo. Trata-se de um vídeo com 5 imagens por segundo, mas em que na Figura 36 são apresentadas apenas as imagens onde existe movimento.





Figura 36 – Sequência de imagens consecutivas captadas pela câmara 1

As imagens decodificadas detectam o movimento existente na porta do Centro de Cálculo e permitem uma identificação da pessoa. As imagens originais ocupam em memória 832KB enquanto as imagens codificadas ocupam em memória 76,7KB sendo necessária uma imagem de fundo que ocupa 45,6KB.

4.4 Tempo de execução do algoritmo

Analisando a complexidade de cada módulo do codificador, em relação ao tempo de execução, apresentam-se os resultados na Tabela 10. Os testes foram realizados sobre um processador a 1,7 GHz numa máquina com 480 MB de RAM.

	Detecção divisão imagem	Codificador Linhas/columnas	Recorte dos blocos movimento	Escolhe cenário	Codificação		Total
Pouco movimento	3,110s	0,271s	0,306s	0,013s	2,481s		6,181s
Movimento normal	3,187s	2,106s	2,130s	0,019s	22,618s		27,954s
Muito movimento	3,858s	6,076s	8,027s	0,057s	128,027s		146,045s

Tabela 10 – Tempo de execução de cada módulo do codificador tendo em conta o cenário

Verifica-se que à medida que o movimento aumenta, o tempo gasto para a execução de todo o algoritmo aumenta. Grande parte do tempo é gasta no bloco codificação; tal deve-se ao elevado número de blocos a codificar. Estes valores referem-se apenas a uma imagem referente a cada cenário. As imagens foram captadas pela câmara 1, em que para o pouco movimento foram codificados 96 blocos, para o movimento normal 832 blocos e para o muito movimento 2448 blocos.

Analisando a complexidade de cada módulo do decodificador, em relação ao tempo de execução, apresentam-se os resultados na Tabela 11.

	Decodificador Coordenadas	Decodificador Huffman	Código RLE	Descodificação		Total
Pouco movimento	0,008s	0,530s	0,011s	1,677s		2,226s
Movimento normal	0,023s	8,429	0,104s	9,554s		18,11s
Muito movimento	0,162s	84,406s	0,234s	22,132s		106,934s

Tabela 11 – Tempo de execução de cada módulo do decodificador tendo em conta o cenário

À medida que o movimento aumenta, existem mais blocos para decodificar e o tempo de execução aumenta. Os tempos referem-se às mesmas imagens que foram consideradas para a Tabela 10.

Capítulo 5 – Conclusões

Neste trabalho apresentou-se um algoritmo de codificação inteligente para o sistema de vídeo vigilância instalado no Centro de Cálculo. No algoritmo tira-se partido do conhecimento prévio das características físicas do local bem como dos hábitos de utilização do mesmo. É usado um detector de movimento, consoante a câmara, hora e dia, para identificar as regiões onde existiu movimento. Apenas essas regiões são alvo de codificação e armazenamento. A qualidade com que estas regiões são codificadas depende do número de blocos das regiões onde existiu movimento, tendo sempre em os objectivos da vigilância assegurando qualidade suficiente para posterior identificação das pessoas, e dos seus movimentos ao longo do corredor.

O algoritmo foi implementado em MATLAB e avaliado usando vídeos recolhidos, pelas 3 câmaras instaladas no Centro de Cálculo, a várias horas do dia e em dias diferentes para testar diversas situações de movimento e de intensidade luminosa.

O codificador apresenta bons resultados na compressão dos vídeos. Embora havendo degradação da qualidade das imagens, são detectadas todas as entradas e saídas, permitindo a identificação das pessoas, bem como os movimentos no corredor do Centro de Cálculo, conseguindo que a quantidade de memória necessária para o armazenamento dessa informação seja bastante reduzida comparando com o sistema actual, e em que as imagens descodificadas apresentam ainda assim qualidade suficiente na generalidade das situações. Esta redução na quantidade de memória necessária para o armazenamento é conseguida uma vez que o algoritmo de compressão implementado está relacionado com as condições para onde foi optimizado, tirando partido assim do conhecimento das suas características. Caso se pretenda aplicar o algoritmo noutras circunstâncias, em geral, haverá degradação do desempenho. Para especialização para outros casos terá que ser feito um estudo prévio das características do local e das condições, bem como alterar alguns parâmetros do algoritmo para, desse modo, tentar

que o algoritmo mantenha as boas taxas de compressão e uma boa qualidade nas imagens armazenadas.

O algoritmo apresenta algumas falhas, nomeadamente na detecção de movimento, uma vez que caso este seja muito reduzido entre duas imagens é apenas codificado esse mesmo movimento fazendo com que a pessoa em causa desapareça da imagem enquanto não existir um movimento significativo, levando a que nestas situações a imagem descodificada apresente uma degradação visível.

Verifica-se que o desempenho do algoritmo, em termos de processamento e de qualidade nas imagens descodificadas, depende muito do detector de movimento e dos valores de limiar usados para a detecção de movimento, uma vez que se o detector de movimento estiver muito sensível obtêm-se muitos blocos para codificar o que leva a que o tempo de processamento desses mesmos blocos seja bastante elevado. Por outro lado, com um detector de movimento menos sensível, obtêm-se menos blocos para codificar mas em que a detecção de movimento deixa de apresentar bons resultados levando a imagens descodificadas visualmente degradadas. A escolha dos valores do limiar depende também da intensidade luminosa existente no raio de acção da câmara em causa.

Como trabalho futuro sugere-se:

- a optimização do código implementado ou implementação do sistema em linguagem JAVA, de modo a assegurar portabilidade.
- deverão ser testadas outras formas de detecção de movimento de modo a que este detecte apenas as pessoas descartando o resto, ou que inclusivé ao detectar uma pessoa siga o seu movimento, tendo deste modo uma imagem da pessoa mas em que apenas é codificado o seu movimento.
- a implementação de mais codificadores, formando uma família de codificadores, de modo a tentar que o sistema se torne mais robusto combinando diferentes codificadores com múltiplos cenários.

Referências

- [1] CCITT, Information Technology-Digital Compression and Coding of Continuous-Tone Still Images-Requirements and Guidelines. Recommendation T.81, 1992
- [2] William B. Pennebaker, Joan L. Mitchell, JPEG: Still Image Data Compression Standard Springer, Edition 2, 1993
- [3] G.K. Wallace, The JPEG Still Picture Compression Standard, Communications of the ACM, ACM, Vol. 34, No. 4, 1991, pp. 31-44.
- [4] K. Sayood, Introduction to Data Compression, Morgan Kaufmann Publishers Inc, 1996.
- [5] M. Oliver. J. Monti, “Surveillance IP camera design with intelligent encoding for reduced bandwidth, higher quality”, disponível em <http://www.videsignline.com/howto/205200047>, acessado em 26 de Abril de 2008, 21 Dezembro, 2007