

Simara Sonaglio

Detecção de movimento através de sinais de vídeo: Estudo e implementação de um sistema

São José – SC
março / 2009

Simara Sonaglio

Detecção de movimento através de sinais de vídeo: Estudo e implementação de um sistema

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina para a obtenção do diploma de Tecnólogo em Sistemas de Telecomunicações.

Orientador:
Prof. Ms. Marcos Moecke

CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES
INSTITUTO FEDERAL DE SANTA CATARINA

São José – SC
março / 2009

Monografia sob o título “Detecção de movimento através de sinais de vídeo: Estudo e implementação de um sistema”, defendida por Simara Sonaglio e aprovada em 10 de março de 2009, em São José, Santa Catarina, pela banca examinadora assim constituída:

Prof. Ms. Marcos Moecke
Orientador

Prof. Dr. Evandro Cantú
IFSC

Prof. Dra. Elen Macedo Lobato Merlin
IFSC

Quer você pense que pode ou não fazer algo, você está certo.

Henry Ford

Agradecimentos

À minha família e namorado pelo incentivo não somente neste momento, mas no decorrer de todo o curso.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo fornecimento de bolsa de pesquisa PIBITI durante a primeira etapa deste trabalho.

Ao IFSC pela disponibilidade de espaço físico para a realização deste trabalho.

Aos colegas de curso e especialmente aos que conviveram comigo no Laboratório de Projeto Final.

Aos professores do IFSC pelo auxílio e incentivo ao longo do curso.

E ao meu orientador, Marcos Moecke, pelo acompanhamento e auxílio no decorrer deste trabalho.

Resumo

Este trabalho apresenta um sistema de detecção de movimento baseado apenas em sinais de vídeo. A informação de vídeo a ser processada é capturada de uma câmera de baixo custo do tipo webcam. Neste sistema apenas são armazenados os quadros com movimento, facilitando a eventual análise por humanos. O padrão de compressão adotado no trabalho é o MPEG4 através do codec de XVID. Um texto é sobreposto a cada imagem, de modo a marcar o instante no qual ocorreu cada evento. Com o objetivo de facilitar a identificação do objeto causador do movimento, é realizada a segmentação do objeto em movimento, o qual é demarcado com uma moldura no vídeo armazenado. Os resultados obtidos mostram que o sistema é eficiente na detecção de movimentos, evitando alarmes falsos devido, entre outros fatores, à variação da iluminação, variação de sombras, alteração de fundo e movimento causado pelo vento.

Palavras Chave: Detecção de movimento, processamento de vídeo, monitoramento eletrônico de ambientes.

Abstract

This work presents a new system for motion detection by analyzing only the video signal. The video is captured by low cost webcam. Only scenes with motion are recorded on digital media, in order to allow a further analysis by human being, The video coding standard used is MPEG4 via codec XVID. A text is over imposed to each image to mark the time that each event has occurred. Aiming to improve the identification of the moving object, a frame is set around it. Results show that the system is efficient for motion detection, avoiding false alarms due to lighting variation, shadows, image background change.

Keywords: Motion detection, video processing, electronic surveillance.

Sumário

1	Introdução	13
2	Fundamentação teórica	16
2.1	Pixel.....	16
2.2	Imagem digital	17
2.3	Operações morfológicas.....	18
2.4	Espaço de cor	19
2.5	Vídeo digital.....	22
2.6	Compressão de vídeo	23
2.7	Modelagem do ambiente	24
2.8	Segmentação do objeto	25
2.9	Detecção de movimento.....	26
2.10	Detecção de não movimento	26
2.11	Tamanho da imagem de um objeto em relação à distância de uma lente	27
3	Detecção de movimento através de sinais de vídeo.....	29
3.1	Material utilizado	30
3.2	Aquisição do vídeo.....	31
3.3	Detecção do movimento.....	31
3.4	Detecção de alterações no fundo.....	43
3.5	Gravação dos quadros com movimento	45
3.6	Ajuste da exposição à luz da câmera.....	48
4	Resultados.....	49
4.1	Detecção de movimento em ambiente interno	49
4.2	Detecção de movimento em ambiente externo	50

4.3	Comparação dos métodos DMFMed e DMFMorf.....	52
4.4	Comparação dos algoritmos DFI e DFB.....	54
4.5	Comparação entre IMU e IMM.....	55
4.6	Gravação de imagens com o codec Indeo5 e XVID.....	55
5	Conclusões e Trabalhos Futuros	57
	Lista de Abreviaturas e Siglas.....	60
	Referências Bibliográficas	61

Lista de Figuras

Figura 1 - Vizinhança de um pixel.....	17
Figura 2 - Representação da imagem digital.....	17
Figura 3 - Exemplo de operação morfológica.....	18
Figura 4 - Espectro de luz visível.	19
Figura 5 - Espaço de cor RGB.	20
Figura 6 - Representação de uma sequência de vídeo digital.	22
Figura 7 - Esquemático do funcionamento de lentes convergentes.	27
Figura 8 - Tamanho da imagem.	28
Figura 9 - Diagrama de blocos da detecção de movimento por subtração de fundo do DMFMed.	32
Figura 10 - Exemplo da segmentação dos objetos com o DMFMed.	33
Figura 11 - Diagrama de blocos da detecção de movimento por subtração de fundo do DMFMorf.....	34
Figura 12 - Exemplo das operações morfológicas empregadas no DMFMorf.....	36
Figura 13 - Diagrama de blocos da contagem de objetos na imagem.	37
Figura 14 - Número de objetos detectados em uma sequência de vídeo, usando o RGB.....	38
Figura 15 - Número de objetos detectados em uma sequência de vídeo, usando o YCbCr.	38
Figura 16 - Número de objetos detectados em uma sequência de vídeo, usando o YIQ.....	39
Figura 17 - Número de objetos detectados em uma sequência de vídeo, usando o YCbCr.	39
Figura 18 - Imagens utilizadas para a determinação do limiar de ruído.....	40
Figura 19 - Determinação do limiar de ruído para a luminância. Utilizado como referência uma imagem de fundo (IMG1) e uma imagem com movimento (IMG2).....	41
Figura 20 - Determinação do limiar de ruído para a cromaticidade. Utilizado como referência uma imagem de fundo (IMG1) e uma imagem com movimento (IMG2).	41
Figura 21 - Determinação do tamanho do elemento estruturante utilizado para a filtragem morfológica. Utilizado como referência uma imagem de fundo (IMG1) e uma imagem com movimento (IMG2).	43
Figura 22 - Exemplo de alteração de cenário da imagem de fundo.....	44
Figura 23 - Diagrama de blocos do método de detecção de mudança de fundo pela diferença temporal entre quadros.....	44
Figura 24 - Exemplo de inserção de moldura.	46
Figura 25 - Exemplo de movimento captado em ambiente interno.	50
Figura 26 - Exemplo de movimento captado em ambiente externo com o uso do VEDAM1 e VEDAM2.	51

Figura 27 - Comparação de desempenho entre o método DMFMed e DMFMorf.	53
Figura 28 - Objeto segmentado com o GIMP.	54
Figura 29 - Conseqüência da atualização da imagem de fundo com os algoritmos DFI e DFB.	55

Lista de Tabelas

Tabela 1 -	Comparação entre diferentes tipos de codecs de compressão de vídeo.....	47
------------	---	----

1 Introdução

A informação visual é provavelmente a melhor fonte usada pelo ser humano para monitoramento de ambientes. Na primeira geração de sistemas de monitoramento por vídeo, esta informação, que tem muita redundância espacial e temporal, era processada pelo cérebro humano do vigilante. Este sistema conhecido como circuito fechado de TV mostrou ser de pouca confiança ao longo do tempo, pois o desempenho para a detecção de movimento decaiu rapidamente quando o vigilante necessita observar continuamente um conjunto de câmeras (LOPEZ et al., 2006). Para superar este problema surgiram sistemas de monitoramento equipados com sensores infra-vermelho, nos quais a detecção do movimento passou a ser automática (SAKAI et al., 1995).

Nos últimos anos houve uma grande expansão no mercado de equipamentos com a capacidade de detectar o movimento através de sensores externos, ou pela própria imagem, possibilitando assim o monitoramento automático e inteligente da cena. Nestes sistemas a observação e interpretação humana foram substituídas por algoritmos computacionais automatizados e inteligentes (WANG et al., 2004) (CEDRAS; SHAH, 1995). A detecção de movimento e a segmentação da imagem são áreas do processamento de sinais que tem forte apelo de uso seja em sistemas de vigilância eletrônica ou em outras aplicações como controle de acesso, visão robótica e sistemas de auxílio à direção de veículos (WANG et al., 2004).

A maioria desses sistemas segmenta a imagem, separando o objeto em movimento do fundo estático. Esta segmentação é realizada com o objetivo de detectar o movimento na cena. O acompanhamento da trajetória do objeto segmentado e a gravação das imagens para posterior análise são opções que também são frequentemente implementadas. O processo de detecção do movimento normalmente envolve tanto a modelagem do ambiente, que consiste na representação da imagem de fundo, como a segmentação do objeto em movimento e a sua classificação (WANG et al., 2004).

Durante o processo de detecção de movimento podem ocorrer falsos alarmes de detecção fazendo com que movimentos que não aconteceram sejam detectados e que movimentos existentes na cena passem não detectados. A produção destes falsos alarmes

depende das condições de aquisição da imagem, resolução da câmera, taxa de quadros, ruído de aquisição, e das condições do ambiente monitorado como iluminação, sombras, vento, reflexos (TOTH et al., 2000).

Neste trabalho foram desenvolvidos dois métodos de detecção de movimento baseado apenas em sinais de vídeo, adquirido através de câmeras de baixo custo do tipo webcam. O primeiro método faz uso de filtragem mediana para a eliminação de ruídos (DMFMed) enquanto o segundo utiliza para isto tanto a filtragem mediana como a filtragem morfológica (DMFMorf). Neste trabalho também foram desenvolvidos dois algoritmos para a detecção de alteração na imagem de fundo, através da alteração da imagem de fundo inteira (DFI) e pela atualização em blocos desse fundo (DFB). Para possibilitar a posterior análise de eventos, as imagens com movimento detectadas são armazenadas em mídia digital com informação de data e hora sobrepostas a ela. Para facilitar a identificação do objeto causador do movimento, uma moldura é inserida em torno deste objeto. Além disto, estas sequências de imagens são comprimidas para reduzir o seu tamanho em bit.

Para a realização dos testes de avaliação foram construídos dois sistemas de vigilância eletrônica de ambientes. O primeiro sistema é denominado “Vigilância Eletrônica por Detecção Automática de Movimento 1” (VEDAM1). Este sistema utiliza o método DMFMed para a detecção de movimento e o algoritmo DFI para detecção de alteração no fundo. A moldura para identificação do objeto causador do movimento é feita através do algoritmo (IMU) e a gravação das imagens em mídia digital é feita como o codec de vídeo Indeo5. O segundo sistema é denominado “Vigilância Eletrônica por Detecção Automática de Movimento 2” (VEDAM2). Este sistema é constituído do método de detecção de movimento DMFMorf e a sua atualização de fundo é feita pelo algoritmo DFB. A inserção de moldura é realizada pelo algoritmo IMM e a gravação das imagens em mídia digital é feita no padrão de compressão MPEG4 através do codec XVID.

A estrutura do restante deste trabalho é apresentada de modo que o leitor que não está familiarizado com a área de processamento de imagens possa entender a proposta elaborada. Ao longo do texto, diversas imagens são utilizadas para ilustrar o processo, recomenda-se que o leitor utilize a versão eletrônica deste documento, ou uma impressão em cores. No capítulo 2 são descritos os fundamentos básicos relacionados ao processamento digital de vídeo e imagem, bem como conceitos mais específicos relacionados à detecção de movimento que precisam ser conhecidos para a correta compreensão deste trabalho. Uma breve descrição do processo de formação da imagem em uma câmera também é apresentada. No capítulo 3 é descrita o sistema desenvolvido para a detecção de movimento através de sinais de vídeo.

Neste capítulo são descritos os dois métodos de detecção desenvolvidos, incluindo algoritmos para alteração de fundo e os algoritmos de inserção de moldura. No capítulo 4 são apresentados os resultados mais significativos alcançados com estes métodos desenvolvidos, usando dois tipos de cenário distinto: ambiente interno, ambiente externo. Este capítulo também compara o desempenho dos algoritmos desenvolvidos para detecção de movimento, detecção de alteração do fundo da imagem e inserção de moldura. No capítulo 5 são apresentadas as conclusões mais relevantes, comentários finais e apresentadas alguns possíveis trabalhos futuros.

2 *Fundamentação teórica*

Neste capítulo são apresentados alguns fundamentos básicos necessários ao entendimento deste trabalho. Nas seções que se seguem são apresentados alguns conceitos relacionados ao processamento digital de imagens, tais como, o *pixel*, a imagem digital, as operações morfológicas, os espaços de cores, o vídeo digital, a compressão de vídeo. Também são apresentados alguns conceitos mais específicos relacionados à detecção de movimento, tais como, a modelagem de ambientes, a segmentação do objeto, a detecção de movimento e a detecção de não movimento. Finalmente é discutida a questão da formação da imagem em uma câmera, envolvendo as distâncias do objeto em relação à distância de uma câmera e a distância focal.

2.1 Pixel

O *pixel* é o menor elemento que constitui uma imagem digital. O *pixel* geralmente tem forma quadrada ou retangular, sendo que a cada ponto é atribuído um valor que representa uma determinada cor. Quanto mais *pixels* forem utilizados para formar uma imagem digital maior será a resolução da mesma, ou seja, haverá mais detalhes nesta imagem acarretando em uma maior qualidade (SCURI, 1999).

Cada *pixel* pode ser representado com um determinado número de bits sendo que quanto maior este número maior a quantidade de cores possíveis para um mesmo ponto, por exemplo, um *pixel* representado com 4 bits pode ter 16 cores diferentes enquanto um com 24 bits pode ter 16,7 milhões de cores. Em imagens coloridas cada *pixel* é representado por três valores correspondentes a cada componente de cor da imagem, sendo que cada um destes pontos é normalmente representado por 8 bits.

Em algoritmos de preenchimento e determinação de continuidade de objetos na imagem, outro ponto importante a ser analisado na estrutura dos *pixels* é o tipo de vizinhança. A Figura 1 ilustra as possíveis vizinhanças de um *pixel*, a qual pode ser do tipo 4 ou 8

conectada. Na vizinhança 4 conectada todos os *pixels* conectados são vizinhos de borda, sendo todos equidistantes, enquanto que na 8 conectada nem todos os vizinhos são equidistantes, podendo ser tanto vizinhos de borda como vizinhos na diagonal. Normalmente vizinhanças do tipo 4 conectadas são utilizadas para os objetos em uma imagem e as 8 conectadas são utilizadas para o fundo (SCURI, 1999).



Figura 1 - Vizinhança de um pixel.

2.2 Imagem digital

No processamento digital de sinais, uma matriz de *pixels* representa um quadro ou uma imagem digital estática. Esta matriz pode ser bidimensional, representando apenas a luminância (preto e branco), ou tridimensional (imagens coloridas). Cada plano bidimensional representa uma das cores do modelo de cor aditivo RGB, conforme mostrado na Figura 2, ou uma das componentes dos espaços de cores YCbCr, YIQ ou HSV. Na imagem, cada pixel RGB é normalmente representado por 24 bits, ou seja, 8 bits para cada componente de cor, sendo que o valor do *pixel* em cada componente RGB pode variar entre 0 (preto) e 255 (branco). A resolução da imagem corresponde ao número de colunas e de linhas de pixels contidas em sua matriz (PRATT, 2001).

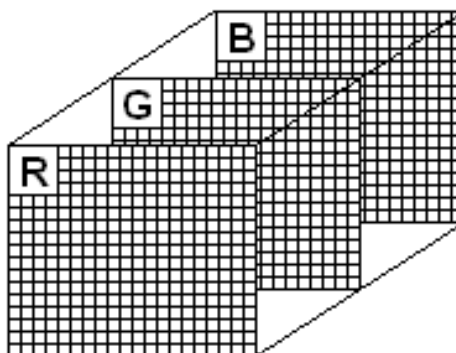


Figura 2 - Representação da imagem digital.

2.3 Operações morfológicas

As operações morfológicas são usadas para extrair partes da imagem que são utilizados para representação e descrição da forma de uma região. As técnicas morfológicas também podem ser utilizadas para pré ou pós processamento, como por exemplo, filtragem morfológica. As operações morfológicas podem ser aplicadas tanto em imagens binárias como em imagens em escala de cinza (GONZALEZ et. al., 2002).

Neste trabalho são aplicados dois tipos de operações morfológicas em imagem binária, sendo elas a erosão e a dilatação. A erosão é a redução do objeto A por um elemento estruturante B a partir da borda do objeto, enquanto que a dilatação faz o contrário, expandindo um objeto A através de um elemento estruturante B. O elemento estruturante usado nestas operações consiste normalmente de uma matriz binária que pode ter diferentes formatos (quadrado, retângulo ou disco) e tamanhos. O tamanho da erosão ou da dilatação é determinado pelo raio deste elemento (PRATT, 2001).

Na Figura 3 é mostrado um exemplo da aplicação das operações morfológicas de erosão e dilatação na filtragem de ruídos. Neste caso, tanto na erosão quanto na dilatação foi utilizado um elemento estruturante com formato de disco e raio equivalente a 5 *pixels*. A Figura 3(a) mostra a imagem original, onde é possível ver a ocorrência de pequenos ruídos. Na Figura 3(b) é apresentado o resultado da erosão aplicada sobre a imagem (a). O processo de erosão fez com que os ruídos da imagem fossem todos eliminados, porém o objeto detectado também sofreu perdas. Na Figura 3(c) é possível notar que as perdas provocadas pela erosão foram recuperadas com o processo de dilatação.

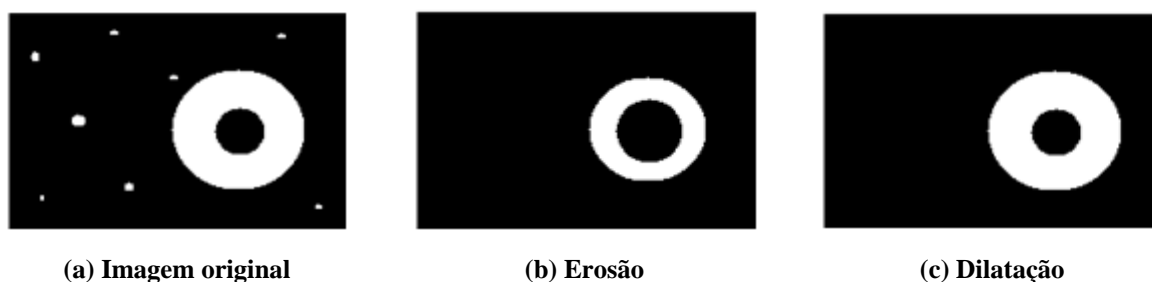


Figura 3 - Exemplo de operação morfológica.

2.4 Espaço de cor

Espaço de cor é um sistema para representação numérica de cores, podendo ser formado por modelos de cor aditivos ou subtrativos. Os modelos de cor aditivos são formados por cores primárias que quando somados os seus comprimentos de onda dão origem a outras cores, já nos modelos de cor subtrativos as cores são obtidas através da subtração entre o comprimento de onda dominante e a luz branca (MACHADO, 2006). Normalmente os espaços de cor são formados por três ou quatro componentes de cores como, por exemplo, o espaço de cor RGB (vermelho, verde e azul) e CMYK (ciano, magenta, amarelo e preto).

No sistema desenvolvido foram testados quatro tipos de espaço de cor diferentes com o propósito de determinar qual seria o melhor para se trabalhar com a detecção de movimento, sendo eles o RGB, YCbCr, YIQ e HSV. A imagem capturada da câmera inicialmente está no formato RGB, necessitando ser convertida para os demais formatos.

2.4.1 RGB

O espaço de cor RGB é formado pelas cores primárias aditivas: vermelho, verde e azul. Estas três cores são ditas como cores primárias, pois as cores vistas pelo olho humano são formadas pela combinação dos seus comprimentos de onda. Na Figura 4 é apresentado o espectro de luz visível ao olho humano, este espectro varia entre 400 e 780 nm, sendo que o comprimento de onda da cor azul é 435,8 nm, do verde é 546,1 nm e do vermelho é 700 nm (GONZALEZ et. al., 2002).

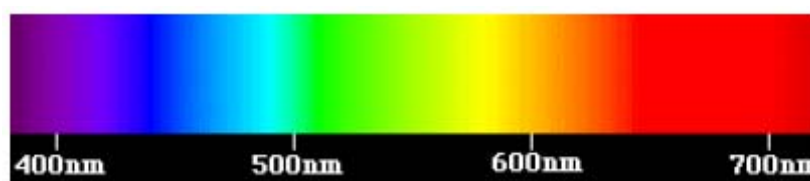


Figura 4 - Espectro de luz visível.
FONTE: SOUTO

Com estas três cores é possível gerar muitas outras, apenas variando o valor de cada componente RGB entre 0 e 255 (ver Figura 5). Por exemplo: a cor amarela é obtida pela interação entre a cor vermelha e a verde, a cor magenta é formada pela cor vermelha e azul, e a cor branca pela interação entre os valores máximos das três cores primárias.

Este espaço de cor é utilizado principalmente em dispositivos de exibição de imagens, como monitores e televisores, também podendo ser utilizado em dispositivos para captura de imagens, como câmeras de vídeo (MACHADO, 2006).

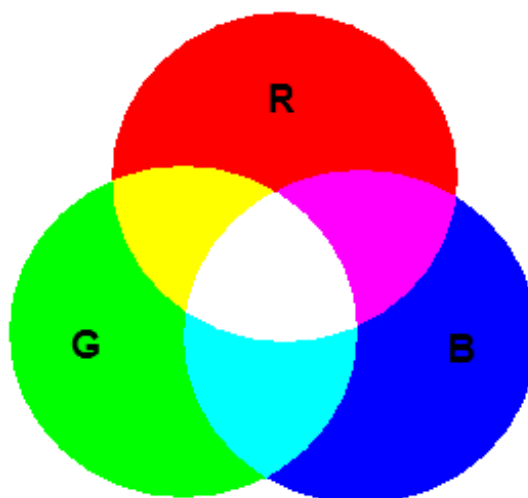


Figura 5 - Espaço de cor RGB.

2.4.2 YCbCr

O espaço de cor YCbCr é utilizado em sistemas de vídeo digital, sendo formado pela componente Y, que representa a luminância da imagem (tons de cinza), e as componentes Cb e Cr, que representam os desvios para as cores azul/amarelo e vermelho/verde, respectivamente. Os valores de Y, Cb e Cr podem variar entre 0 de 255 (PRATT, 2001). A equação de conversão do espaço de cores RGB para o YCbCr é apresentada a seguir:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} + \begin{pmatrix} 65,481 & 128,553 & 24,966 \\ -37,797 & -74,203 & 112,000 \\ 112,000 & -93,786 & -18,214 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (1)$$

2.4.3 YIQ

O YIQ é um espaço de cor desenvolvido pelo NTSC (*National Television System Committee*). Este espaço de cor é composto por três componentes, sendo eles o Y que corresponde a luminância e o I e Q que correspondem a crominância e descrevem os atributos de matiz e saturação da imagem. O matiz de uma imagem está associado ao comprimento de

onda dominante entre todos os comprimentos de onda que formam uma cor (PRATT, 2001). A equação de conversão do espaço de cores RGB para o YIQ é apresentada a seguir:

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ 0,595716 & -0,274453 & -0,321263 \\ 0,211456 & -0,522591 & 0,311135 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (2)$$

2.4.4 HSV

O HSV é um espaço de cor formado por uma componente H que define a matiz da imagem (*Hue*), uma componente S para definir a saturação da imagem (*Saturation*) e uma componente V para definir o valor da intensidade em tons de cinza da imagem (*Value*). Este espaço de cor é utilizado principalmente em algoritmos de processamento de imagem onde é necessária uma descrição de cores natural e intuitiva aos seres humanos (GONZALEZ et. al., 2002). As equações de conversão do espaço de cor RGB para o HSV são apresentadas a seguir.

A equação de conversão da componente que representa a matiz é dada por:

$$H = \begin{cases} \theta & \text{se } B \leq G \\ 360 - \theta & \text{se } B > G \end{cases} \quad (3)$$

Sendo:

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G) + (R-B)]}{\left[\frac{1}{2}[(R-G)^2 + (R-B)(G-B)] \right]^{\frac{1}{2}}} \right\} \quad (4)$$

A equação de conversão da componente de saturação da imagem é dada a seguir:

$$S = 1 - \frac{3}{(R+G+B)} [\min(R, G, B)] \quad (5)$$

A equação de conversão da componente que representa a intensidade da imagem é apresentada a baixo:

$$V = \frac{1}{3}(R + G + B) \quad (6)$$

2.5 Vídeo digital

O sinal de vídeo digital consiste basicamente em uma seqüência de imagens digitais reproduzidas com uma determinada taxa de quadros para representar cenas em movimento (ver Figura 6). Esta taxa é medida em quadros por segundo e quanto maior, melhor será a percepção pelo sistema visual humano como sendo uma seqüência contínua no tempo. A taxa de quadros mínima para gerar em seres humanos a sensação de continuidade no tempo é de 15 quadros por segundo (PRATT, 2001).

Outro parâmetro para medir a qualidade de um vídeo digital é a taxa de bits. Esta taxa é medida em bits por segundo e quanto maior o seu valor melhor será a qualidade do vídeo. Ela pode ser de dois tipos: taxa constante de bits e taxa variável de bits. Na taxa constante de bits, como o próprio nome sugere, o valor da taxa se mantém sempre igual em todo o vídeo, esta técnica é utilizada principalmente em aplicações onde a largura de banda é fixa. Já na taxa variável de bits o valor varia conforme o tipo de informação presente no vídeo, em trechos do vídeo onde há variações muito rápidas das cenas é utilizado uma taxa maior, em trechos em que as variações ocorridas são pequenas, a taxa utilizada é menor.

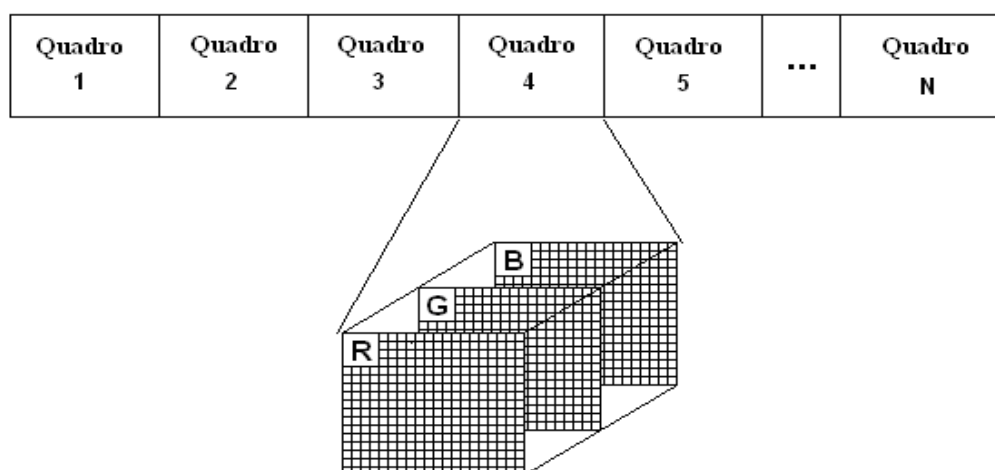


Figura 6 - Representação de uma seqüência de vídeo digital.

2.6 Compressão de vídeo

Para reduzir a taxa de bits das sequências de vídeo são utilizadas várias técnicas de compressão de vídeo. Essa redução visa tanto à redução da banda necessária na transmissão em redes de computadores como no tamanho usado na armazenagem do vídeo em mídia digital. Esta redução de tamanho é possível diminuindo as redundâncias temporais e espaciais entre as imagens dos quadros sucessivos. Por exemplo, em um vídeo com uma rua (como fundo estático) e um carro ou pessoa se movendo, somente é necessário que fundo seja representado no primeiro quadro, enquanto que nos demais apenas é necessário armazenar a diferença entre o quadro anterior.

As técnicas de compressão de vídeo podem ser sem perdas e com perdas. A compressão sem perda é aquela em que o sinal original é comprimido de tal forma que o sinal reconstruído é exatamente igual ao original. Neste caso normalmente a taxa de compressão é baixa. Por outro lado, na compressão com perda são obtidas taxas de compressão mais elevadas, porém o sinal reconstruído não é idêntico ao sinal original acarretando em perda de qualidade (CARVALHO, C.A. et. al.).

A seguir serão apresentadas algumas das técnicas de compressão implementadas em codecs de vídeo. Existe uma grande variedade de codecs, neste trabalho foram apenas avaliados os seguintes: XVID, Indeo3, Indeo5, Cinepak, IYUV e DIVX.

2.6.1 XVID

O XVID é uma implementação de fonte aberta para codificador/decodificador do padrão de compressão de vídeo MPEG-4. O XVID remove as informações do vídeo que não são importantes para a percepção humana, obtendo assim altas taxas de compressão e mantendo uma boa qualidade visual da imagem. O codec XVID pode ser executado em qualquer sistema operacional (XVID, 2009).

2.6.2 Indeo

O Indeo Video é um codec que foi desenvolvido pela Intel em 1992 e posteriormente vendido a Ligos Corporation. Esse codec teve seu maior uso em meados dos anos 90, sendo depois superado por codecs baseados nos padrões MPEG. Apesar disso, o Indeo ainda é utilizado em alguns tipos de jogos (LIGOS CORPORATION, 2009).

Na terceira versão do Indeo Vídeo (Indeo3), eram utilizadas técnicas como amostragem de cor, quantização de vetor e RLE (*Run Length Encoding*) (A FERGUSON, 2001). Na quinta versão (Indeo5) o codec passou a codificar separadamente o fundo e os objetos de cada quadro do vídeo (LIGOS CORPORATION, 2007).

2.6.3 Cinepak

O codec de vídeo Cinepak foi desenvolvido pela empresa SuperMatch, sendo projetado para codificar vídeos com resoluções de 320x240 para taxas de transferência de 150 kbps. O Cinepak é baseado na quantização vetorial, possibilitando baixa complexidade computacional. No Cinepak cada quadro é segmentado em blocos de tamanho 4x4 e cada um destes blocos pode ser codificado usando 1 ou 4 vetores de quantização (B FERGUSON, 2001).

2.6.4 IYUV

O IYUV é o código de 4 caracteres (FourCC – *Four Character Code*) que representa o codec de vídeo Intel YUV. Este codec utiliza a técnica da subamostragem das componentes de cor para reduzir o tamanho do vídeo, ou seja, ele reduz a resolução das componentes de cor (FOURCC, 2009).

2.6.5 DIVX

O DIVX é um codec de vídeo desenvolvido pela empresa Divx, Inc. O DVIX utiliza o algoritmo de compressão H.264, apresentando boa capacidade de compressão e qualidade do vídeo, porém exige grande capacidade de processamento. Este codec é compatível com os sistemas operacionais Windows, Linux, Solaris e Mac OS X (DIVX, 2009).

2.7 Modelagem do ambiente

A modelagem do ambiente é utilizada para representar um ambiente tri-dimensional em uma imagem bi-dimensional. Esta imagem é denominada de fundo e deve preferencialmente corresponder a uma cena estática. A adequada detecção de movimento depende de uma correta modelagem do ambiente. Modelando de modo eficiente o ambiente evita-se comparar

uma imagem com um fundo incorreto, assim evitando fazer com que movimentos que deveriam ser detectados não sejam e que movimentos que na verdade não existem sejam detectados. O modelo do ambiente é utilizado na detecção de movimento através da técnica de subtração de fundo.

Em um sistema de vigilância como o proposto neste trabalho, a imagem de fundo usada necessita ser constantemente atualizada, devido às variações que ocorrem na imagem capturada pela câmera, seja em função das variações na iluminação, movimentos da câmera, ou alterações da cena. Estas últimas podem ser produzidas pela mudança de objetos que são movidos temporariamente, e que uma vez tornados estáticos não devem mais ser considerados como movimento pelo algoritmo de detecção (BROFFERIO et al., 1990).

2.8 Segmentação do objeto

Segmentar uma imagem significa, de modo geral, isolar regiões da mesma que possuam propriedades comuns entre si. A imagem pode ser segmentada analisando-se propriedades como luminosidade, cor, textura e movimento. É comum chamar de objeto da imagem o conjunto de *pixels* no qual se está focando o interesse e de fundo os *pixels* que podem ser desprezados.

A segmentação normalmente é baseada na descontinuidade ou na similaridade entre os *pixels* de uma região. Na descontinuidade entre *pixels* a segmentação se baseia da detecção de mudanças abruptas na luminância ou crominância da imagem. A técnica baseada na descontinuidade mais utilizada é a detecção de bordas.

Na similaridade entre *pixels* a imagem é dividida em regiões que são similares entre si de acordo com alguma característica predefinida. A binarização é uma das técnicas baseada neste conceito muito utilizada para segmentar a imagem. Esta técnica consiste em determinar um limiar de corte para separar o objeto e o fundo, sendo que os *pixels* que tem valor acima do limiar são considerados como sendo o objeto, representados por “1”, e o demais são considerado como fundo e representados por “0” (GONZALEZ et. al., 2002). A imagem obtida constituída de apenas dois níveis “0” e “1” é denominada de imagem binarizada.

2.9 Detecção de movimento

A detecção de movimento através de sinais de vídeo tem como objetivo básico detectar o movimento em ambientes com predominância de cenas estáticas, ou seja, em ambientes em que normalmente não ocorre movimentação (JONES, 1995) (PINHEIRO; LINS, 2007). A implementação da detecção de movimento pode ser feita principalmente através de duas técnicas: 1) a subtração do fundo, que consiste na diferença entre o modelo do ambiente e a imagem atual captada; ou 2) através da diferença temporal entre quadros sucessivos (WANG et al., 2004).

Neste trabalho, de acordo com a situação, foram utilizadas as duas técnicas para a detecção do movimento. A subtração do fundo foi utilizada para a detecção de movimento propriamente dita e a diferença temporal entre quadros sucessivos foi utilizada para a alteração de fundo.

2.10 Detecção de não movimento

Ao contrário da detecção de movimento, a detecção de não movimento tem como objetivo detectar a ausência de movimentação de um determinado objeto em ambientes onde há predomínio de movimento. Neste tipo de ambiente a detecção de movimento seria inviável, pois a movimentação é constante (JONES, 1995).

Um exemplo que descreve bem a idéia da detecção de não movimento é apresentado por JONES, 1995. Considere uma estação ferroviária na qual há predomínio de cenas com movimento. Caso uma pessoa abandonar uma mala no chão, está se tornando um objeto sem movimento e, se deixada nesta situação por mais que um tempo predeterminado, poderá detectada.

Neste trabalho a detecção de não movimento foi utilizada com uma função diferente da descrita acima. Ela é usada para detectar que uma imagem captada em ambientes predominantemente sem movimento corresponde a uma cena estática e portando o fundo da imagem pode ser substituído. A detecção de não movimento neste caso é implementada através da técnica de diferença temporal entre quadros sucessivos.

2.11 Tamanho da imagem de um objeto em relação à distância de uma lente

Câmeras de vídeo são dispositivos para captura de imagens em tempo real. Estes dispositivos podem ou não utilizar lentes para a captura de imagens. Os dispositivos que não possuem lentes são conhecidos como *pin hole* (do inglês, buraco de alfinete). Estes dispositivos são constituídos de um ambiente completamente vedado à luz, onde um pequeno buraco permite que raios luminosos projetados por um objeto entrem, formando a imagem deste objeto no material fotosensível no interior do ambiente (YOUNG, 1989).

As câmeras de vídeo convencionais utilizam lentes esféricas biconvexas para capturar imagens. As lentes biconvexas são lentes convergentes que fazem com que os raios luminosos que incidem paralelamente nelas sejam convergidos para um ponto sobre o seu eixo.

Para determinar a imagem de um objeto refletido precisamos conhecer apenas dois raios luminosos (r_1 e r_2) que incidem na lente. Como mostra a Figura 7, o r_1 é um raio luminoso que incide na lente passando pelo segundo foco da mesma (F_2) e é refratado paralelamente ao seu eixo. O r_2 é o raio que incide paralelamente a lente e é refratado passando pelo primeiro foco da mesma (F_1) (ALVARENGA et. al., 2000).

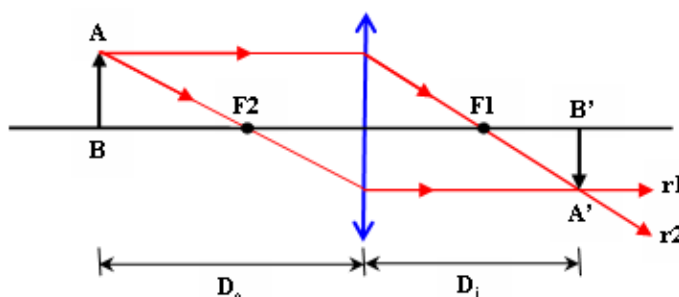


Figura 7 - Esquemático do funcionamento de lentes convergentes.

Para determinar o aumento produzido por uma lente é utilizada a equação abaixo:

$$\frac{A'B'}{AB} = \frac{D_i}{D_o} \quad (7)$$

onde $A'B'$ é o tamanho da imagem, AB é o tamanho do objeto, D_i é a distância entre a lente

e a imagem do objeto e D_o é a distância entre a lente e o objeto.

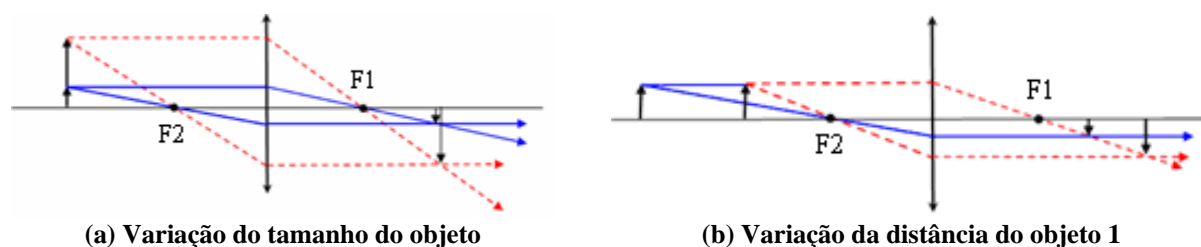
A distância em que a imagem se forma D_i também está relacionada com a distância do objeto D_o e a distância focal da lente f , através da equação de Gauss:

$$\frac{1}{f} = \frac{1}{D_i} + \frac{1}{D_o} \quad (8)$$

Outro parâmetro relacionado às lentes é o aumento linear transversal A , o qual relaciona o tamanho do objeto com sua imagem:

$$A = \frac{A'B'}{AB} \quad (9)$$

Na Figura 8 é exemplificada a variação do tamanho da imagem de um objeto. Através desta imagem é possível ver que o tamanho da imagem de um objeto pode variar de acordo com o tamanho do próprio objeto [Figura 8(a)], ou variar devido à distância do objeto em relação à lente [Figura 8(b)]. Neste último caso, quanto mais longe da lente estiver o objeto, menor será o tamanho da sua imagem.



(a) Variação do tamanho do objeto

(b) Variação da distância do objeto 1

Figura 8 - Tamanho da imagem.

3 Detecção de movimento através de sinais de vídeo

O objetivo desse trabalho foi desenvolver um sistema de detecção de movimento baseado apenas em sinais de vídeo. Este método deve ser capaz de eliminar a informação irrelevante causada por problemas na aquisição da imagem, resolução da câmera, taxa de quadros, ruído de aquisição, e condições do ambiente como iluminação, sombras, vento, reflexos (TOTH et al., 2000). Desta forma, mantendo o máximo de detalhe na imagem, permitindo, assim, uma melhor detecção do movimento e maior redução de alarmes falsos. Obtendo-se um método de detecção de movimento automático e confiável.

Neste trabalho foram desenvolvidos dois sistemas de vigilância eletrônica de detecção de ambientes. O primeiro sistema foi desenvolvido durante o biênio 2007-2008 como projeto de pesquisa e é chamado de “Vigilância Eletrônica por Detecção Automática de Movimento 1” (VEDAM1). O VEDAM1 faz a detecção de movimento através do método DMFMed e a detecção de alteração na imagem de fundo é feita pelo algoritmo DFI. A moldura no objeto é inserida pelo algoritmo IMU. Neste sistema as imagens são armazenadas com o codec de vídeo Indeo5. O segundo sistema foi desenvolvido durante o período do trabalho de conclusão de curso e é chamado de “Vigilância Eletrônica por Detecção Automática de Movimento 2” (VEDAM2). O VEDAM2 detecta movimento através do método DMFMorf e a detecção de alteração no fundo é realizada pelo algoritmo DFB. Neste sistema a inserção de moldura é feita com o algoritmo IMM. No VEDAM2, as imagens são armazenadas com o padrão de compressão de vídeo MPEG4 através do codec XVID.

Nas seções que seguem são apresentados os métodos utilizados nos sistemas VEDAM1 e VEDAM2. Na Seção 3.3 são discutidos os métodos de detecção de movimento desenvolvidos DMFMed e DMFMorf. Na Seção 3.4 são apresentados os algoritmos de detecção de alteração de fundo DFI e DFB, e na Seção 3.5 é discutida a forma de gravação das imagens detectadas com movimento, os algoritmos de inserção de moldura IMU e IMM e a determinação do tipo de codec de vídeo a ser utilizado pelo VEDAM2. Finalmente, na Seção

3.6 é apresentado o método utilizado pelo VEDAM2 para ajuste da exposição à luz da câmera (VEL).

3.1 Material utilizado

Para o desenvolvimento do projeto foram utilizados basicamente componentes de hardware e de software. O hardware utilizado foi uma câmera de vídeo para a captura da imagem e um computador do tipo Desktop para o processamento das imagens. A aquisição das imagens foi realizada com câmeras de vídeo do tipo webcam. A primeira câmera utilizada foi um modelo GOTEC DROP CAM 100 da Leadership com resolução de 160×120 até 640×480 *pixels* e taxa de 10 a 30 quadros por segundo (CAM1). A segunda foi uma câmera modelo PK-333MB da A4TECH, com resolução entre 160×120 e 1280×960 e taxa de 6 a 30 quadros por segundo (CAM2). Estas câmeras por utilizarem sensores CMOS de baixo custo apresentam um forte ruído de aquisição, principalmente quando o ambiente é pouco iluminado. Para reduzir este ruído, foi utilizada uma filtragem mediana no DMFMorf para evitar a detecção de falsos movimentos. O uso de duas câmeras nos experimentos visou principalmente comparar o seu desempenho no monitoramento de ambientes internos e externos e foram utilizadas apenas no desenvolvimento do DMFMed.

Na abordagem proposta as câmeras foram mantidas fixas, predefinindo-se também a distância focal, contraste, cor e brilho. No ambiente externo foi usada apenas a CAM2, que possibilita uma melhor fixação do equipamento para a captura das imagens, além de possuir uma maior resolução espacial permitindo a realização de zoom digital. Na câmera CAM1, foi desabilitado o auto-ajuste de branco, enquanto que na CAM2 foi mantido selecionado o auto-ajuste de branco, para compensar as maiores variações de luminosidade no ambiente externo.

Para o processamento das imagens foi utilizado um computador com processador Intel Pentium Dual Core de 1,6GHz, com memória de 2GB, sistema operacional Windows XP e monitor de LCD de 17" com resolução de 1280×1024 *pixels*.

Os algoritmos de detecção de movimento, segmentação da imagem e gravação do vídeo foram desenvolvidos em linguagem m com o software MATLAB 2007b.

3.2 Aquisição do vídeo

A captura da informação de vídeo é realizada com uma câmera de baixa resolução do tipo webcam. Para que esta webcam possa se comunicar com o computador, são configurados alguns parâmetros como o tipo de adaptador de vídeo, taxa de quadros e tipo de gatilho (*trigger*) utilizado. Este último parâmetro é utilizado para marcar o momento em que a câmera deverá adquirir uma nova imagem. Ele é configurado neste trabalho como manual para evitar que haja falta de sincronismo entre a captura e o processamento. Também são configurados outros parâmetros da imagem tais como: contraste, exposição à luz, brilho e saturação.

No algoritmo de aquisição do vídeo, a captura de imagens ocorre continuamente através de um laço infinito. Para dar início a este laço é necessário que seja iniciado o objeto de captura de vídeo. A cada nova iteração é preciso que seja feito novo gatilho para que a câmera possa capturar uma outra imagem.

3.3 Detecção do movimento

A detecção do movimento em sequências de imagens foi realizada utilizando os métodos o DMFMed e o DMFMorf. O DMFMed apresentou algumas deficiências quanto a segmentação do objeto. Devido ao alto limiar de ruído que foi aplicado, em alguns casos, partes do objeto não são detectadas. Estas partes são confundidas com o fundo da imagem, resultando em uma segmentação falha, onde um mesmo objeto pode ser separado em vários objetos de menor tamanho. O DMFMorf busca corrigir estes problemas, através da alteração do tipo de filtragem aplicada e da redução dos valores dos limiares de ruído.

A detecção de alteração do fundo da imagem foi realizada através do método DFI e DFB. O DFI faz a atualização de fundo inteiro e é utilizado em conjunto com o método de detecção de movimento DMFMed. O DFB faz a atualização da imagem de fundo dividindo-a em blocos menores. Este método é utilizado junto com o DMFMorf.

Nas próximas seções cada método será descrito em detalhes. Na Subseção 3.3.3, a escolha do espaço de cores YCbCr entre os demais espaços é justificada. A Subseção 3.3.4 demonstra como os limiares de ruído utilizados para o espaço de cor YCbCr são obtidos e a

Subsecção 3.3.5 explica como se determina o tamanho do elemento estruturante utilizado na filtragem morfológica.

3.3.1 Detecção de Movimento por Filtragem Mediana (DMFMed)

O método de detecção do movimento usado no DMFMed é descrito na forma de um diagrama de blocos na Figura 9. O processamento foi realizado no espaço de cor YCbCr. Os limiares de ruído utilizados para a componente de luminância e as componentes de crominância foram de 30 e 20, respectivamente.

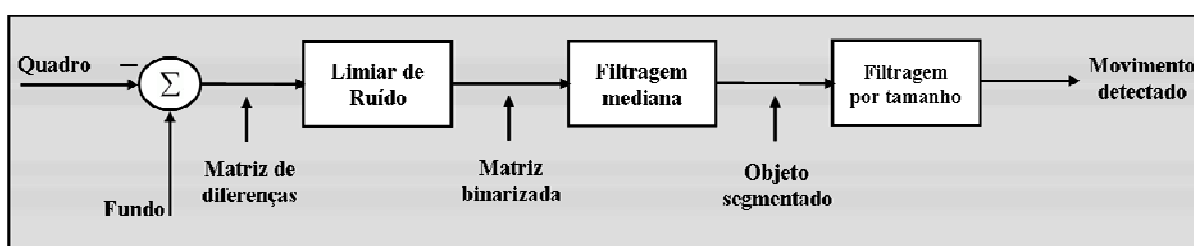


Figura 9 - Diagrama de blocos da detecção de movimento por subtração de fundo do DMFMed.

a) PASSO 1

O primeiro passo é a subtração do quadro de fundo [Figura 10(a)] pela imagem atual [Figura 10(b)], obtendo-se uma matriz de diferenças entre estas imagens. Os valores de cada ponto desta matriz podem variar entre -255 e 255. Tanto a imagem de fundo quanto a imagem atual são convertidas do espaço de cor RGB para o YCbCr antes do início da detecção de movimento.

b) PASSO 2

No segundo passo a matriz de diferenças é comparada *pixel-a-pixel* com o limiar de ruído determinado para a câmera, resultando em uma matriz binária de diferenças na qual os valores acima do limiar são representados por 1 e abaixo por 0 [Figura 10(c)]. Os *pixels* representados por 1 são candidatos a pertencerem a um objeto em movimento.

c) PASSO 3

No terceiro passo é realizada uma filtragem mediana para zerar *pixels* isolados da matriz binária. Esta filtragem é utilizada para eliminar pequenos ruídos da imagem, desta forma diminuindo muitas informações irrelevantes. Após esta etapa temos o objeto segmentado, ou

seja, temos separados o fundo em preto (0) e os objetos em movimento em branco (1) [Figura 10(d)]. A filtragem mediana é realizada em blocos de 3×3 *pixels* com de uma imagem binarizada. A equação que representa esta filtragem é apresentada a seguir:

$$x[k,l] = \frac{1}{9} \sum_{j=k-1}^{k+1} \sum_{i=l-1}^{l+1} x[i,j] \quad (7)$$

onde $x[i,j]$ é o *pixel* na posição horizontal i e vertical j .

d) PASSO 4

No quarto e último passo é realizada uma filtragem por tamanho dos objetos segmentados, na qual os objetos são analisados quanto a sua área total em *pixels*. Objetos menores que 100 *pixels* são descartados, finalizando o processo de detecção de movimento.

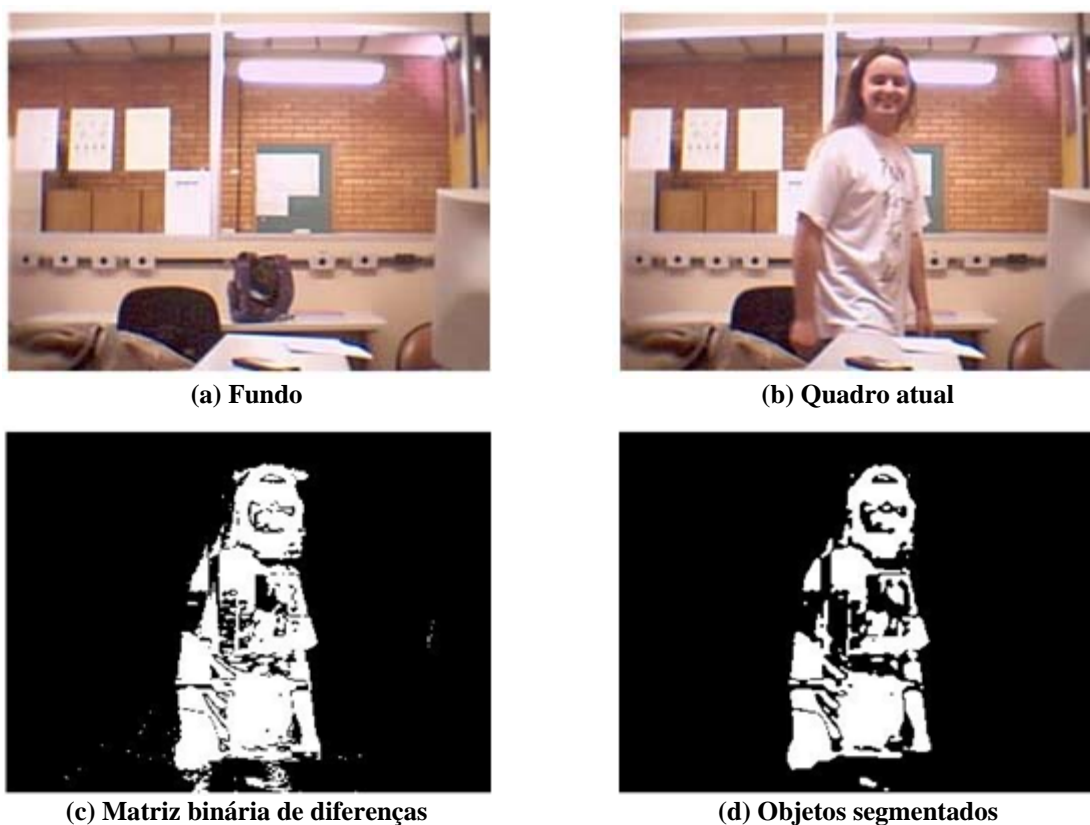


Figura 10 - Exemplo da segmentação dos objetos com o DMFMed.

3.3.2 Detecção de Movimento por Filtragem Morfológica (DMFMorf)

A detecção de movimento do DMFMorf também é feita através da subtração de fundo,

porém a filtragem foi alterada, como mostra o diagrama de blocos na Figura 11.

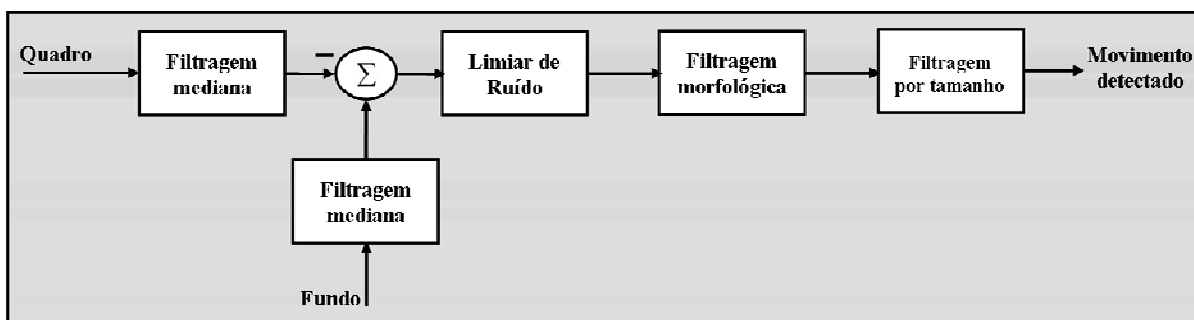


Figura 11 - Diagrama de blocos da detecção de movimento por subtração de fundo do DMFMorf.

a) PASSO 1

No primeiro passo, a imagem convertida do espaço de cor RGB para o YCbCr é filtrada. A filtragem realizada é uma filtragem mediana feita tanto na imagem atual quanto na imagem de fundo, nas componentes Y, Cb e Cr. Esta filtragem é realizada para compensar o forte ruído de aquisição de imagens que câmeras do tipo webcam apresentam. A equação da filtragem mediana do DMFMorf é a mesma que a apresentada em (10), diferindo desta apenas no intervalo de valores que os *pixels* da imagem podem assumir, sendo este intervalo de 0 e 255.

b) PASSO 2

No segundo passo, é realizada a subtração da imagem de fundo pela imagem atual, resultando em uma matriz de diferenças.

c) PASSO 3

No terceiro passo, os *pixels* da matriz de diferenças são comparados com os limiares de ruído determinados. Para a componente de luminância foi usado um limiar de ruído de 30 e para a crominância de 5. Após a aplicação do limiar de ruído é obtida uma imagem binarizada, na qual os *pixels* iguais a “1” indicam que os mesmos sofreram alteração superior ao limiar e, portanto podem pertencer a um objeto em movimento.

Os limiares de ruído foram reduzidos em relação ao algoritmo anterior devido principalmente a alteração na forma de filtragem. A filtragem mediana realizada nas componentes de cor YCbCr faz com que o ruído de aquisição da câmera seja diminuído antes da operação de subtração de fundo. Quando a subtração da imagem de fundo pela imagem atual é realizada no DMFMorf, a matriz de diferenças apresenta menos ruído que a matriz de diferença do DMFMed. A filtragem morfológica também influencia neste resultado, pois

permite a eliminação dos ruídos com maior tamanho.

O limiar para a luminância é maior que o da crominância por ser nesta componente que se encontram as maiores variações na imagem, por exemplo, variações de iluminação, sombras projetadas por objetos e reflexos de luz. O limiar para a crominância foi diminuído de 20 para 5 para permitir uma melhor segmentação do objeto causador do movimento. Esta alteração foi possível porque estas componentes da imagem apresentam menos variação.

d) PASSO 4

O quarto passo é a filtragem morfológica utilizada para eliminação de ruídos na imagem binarizada [Figura 12(a)]. Um exemplo deste procedimento é apresentado na Figura 12. A filtragem morfológica é feita através de operações de erosão e dilatação da imagem binarizada. Na primeira etapa da filtragem morfológica é realizada uma erosão [Figura 12(b)] nesta imagem. Esta erosão tem a finalidade de eliminar os elementos da imagem correspondentes a ruídos. Nesta etapa, são eliminados os elementos da imagem binarizada correspondentes a ruídos menores que o elemento estruturante usado. Alguns elementos de ruídos que são maiores que o dobro do raio do elemento estruturante não são eliminados. Como consequência da erosão ocorre a diminuição do tamanho do objeto causador do movimento. A segunda etapa da filtragem morfológica é a aplicação de uma dilatação [Figura 12(c)] com o mesmo elemento estruturante para recuperar o tamanho e forma similar a do objeto principal. Em uma terceira etapa, é realizado um procedimento de preenchimento dos contornos do objeto que eventualmente não ficaram bem segmentados [Figura 12(d) e (e)]. Note que em (c) é possível ver as falhas de segmentação no contorno do objeto na imagem segmentada após a filtragem morfológica, enquanto que em (d) estas falhas de contorno foram quase completamente corrigidas através da operação de dilatação do objeto. Como o tamanho do objeto aumentou e sua forma mudou, é realizada em seguida uma operação de erosão para restabelecer a forma e tamanho original do objeto (e).

e) PASSO 5

No quinto passo é realizada a filtragem por tamanho do objeto segmentado. Os objetos menores do que 100 *pixels*, ou seja, objetos que representam menos de 0,13% da imagem são eliminados.

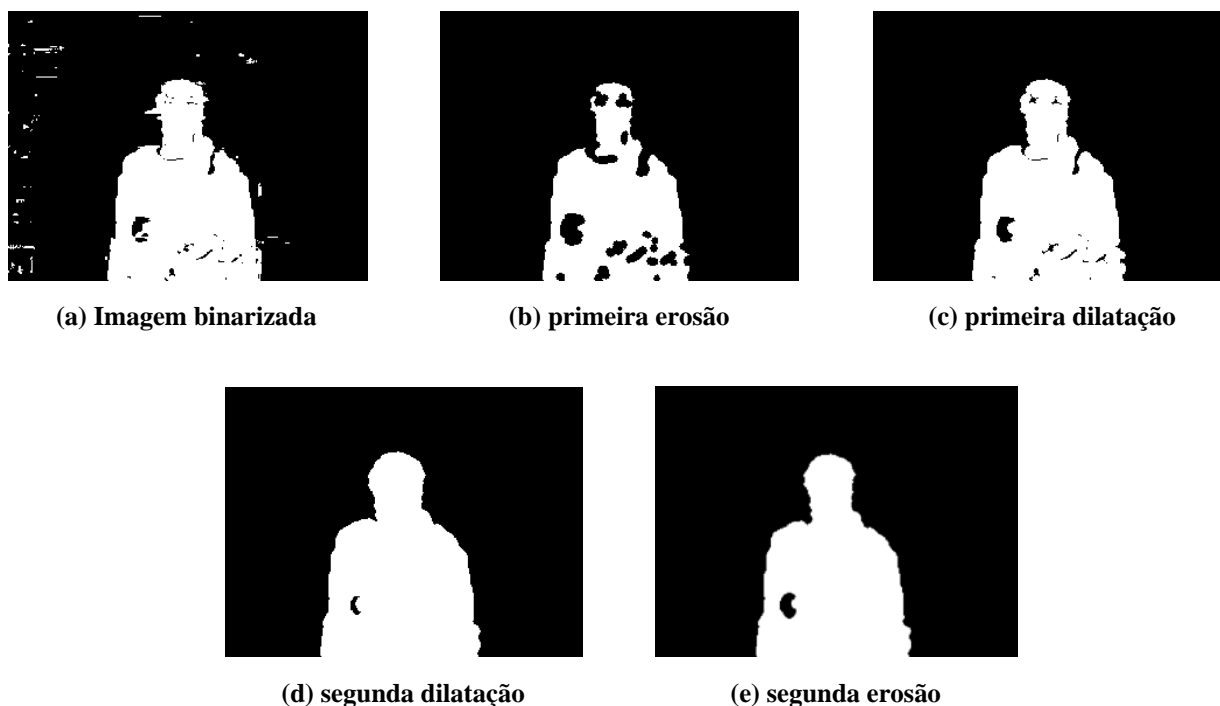


Figura 12 - Exemplo das operações morfológicas empregadas no DMFMorf.

O processamento dos dois métodos é realizado simultaneamente nas 3 componentes de cor Y, Cb e Cr, sendo ao final ainda obtida a imagem segmentada através da combinação dessas componentes com uma operação lógica OU *pixel-a-pixel*. A imagem obtida após esta operação representa o objeto em movimento. A área contígua ocupada em número de *pixels* iguais a 1 indica a posição estimada deste objeto. Este objeto pode apresentar ao final do processo de detecção de movimento problemas como ruídos que não puderam ser eliminados, má formação do contorno do objeto, vazios dentro do objeto e dimensão errada.

3.3.3 Espaços de cores para a detecção de movimento

Para determinar qual espaço de cor seria utilizado na detecção de movimento foram avaliados os espaços de cores RGB, YCbCr, YIQ e HSV. O critério para escolha do melhor espaço de cor foi a verificação de qual apresentava melhor segmentação do objeto e maior imunidade a ruídos. Para cada espaço de cor foi necessário estabelecer um limiar de ruído a ser aplicado para cada componente de cor. Este limiar determina quanto o valor de um pixel da imagem atual pode variar em relação à imagem de fundo sem que seja detectada a alteração. O valor deste limiar foi obtido variando o seu valor entre 0 e 50. Para cada valor de limiar foi calculado o número de objetos em uma cena. As imagens utilizadas nesta avaliação foram: 1) uma imagem representando o fundo estático e 2) uma imagem com um único objeto causando movimento neste fundo. Em cada caso o limiar escolhido foi aquele que

apresentou o número de objetos encontrados na imagem mais próximo do valor real (um objeto na avaliação feita).

Em seguida foi usada a sequência de teste denominada SEQ_TESTE (disponível em: http://www.sj.cefetsc.edu.br/~simara/SEQ_TESTE.mat). Esta sequência é constituída de 51 quadros usados para a análise dos espaços de cor, sendo o primeiro quadro desta sequência uma imagem estática para representar o fundo e os outros 50, cenas que apresentam um único objeto em movimento. Com esta sequência foi aplicado o procedimento apresentado na Figura 13. O procedimento é o mesmo empregado na detecção de movimento desde a filtragem mediana até a filtragem morfológica, sendo realizada após essa etapa a contagem do número de objetos binarizados na cena. O espaço de cor a ser utilizado no DMFMorf foi escolhido com base no número de objetos encontrados em cada imagem, na sequência utilizada o valor correto é um. As variações do número de objetos se devem a dois fatores principais, a não eliminação de ruídos pelo algoritmo de filtragem e a falha na segmentação do objeto. Em todas as filtrações morfológicas foi usado um elemento estruturante com raio 3 em formato de disco.

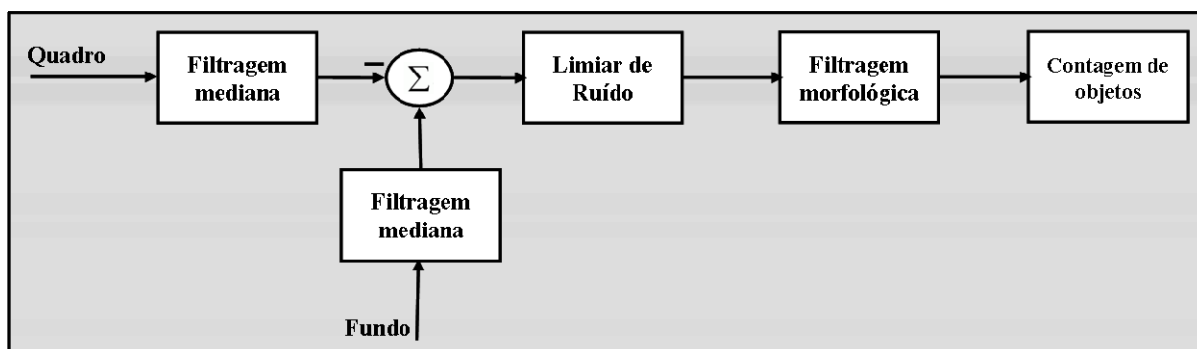


Figura 13 - Diagrama de blocos da contagem de objetos na imagem.

No espaço de cor RGB foi aplicado limiar de ruído igual a 30 para as três componentes de cor, pois estas apresentam variações de luminosidade e reflexos de luz, uma vez que, cada componente tem misturadas as informações de luminância e cor. A Figura 14 apresenta os resultados obtidos para o número de objetos encontrados para cada quadro da sequência de vídeo analisada, sendo possível ver que o número máximo de objetos encontrado foi dois e que esse evento ocorreu em vários quadros da sequência.

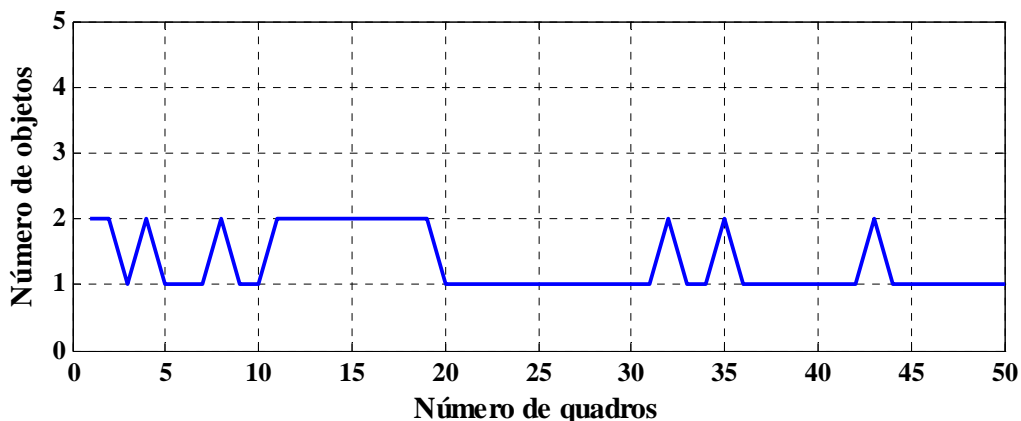


Figura 14 - Número de objetos detectados em uma sequência de vídeo, usando o RGB.

No espaço de cor YCbCr, foi usado um limiar 30 para a componente de luminância Y e um limiar de 5 para as componentes de crominância Cb e Cr. A Figura 15 mostra o número de objetos encontrados, sendo possível ver que o número máximo de objetos encontrados novamente foi dois, mas neste caso apenas em quatro quadros da sequência.

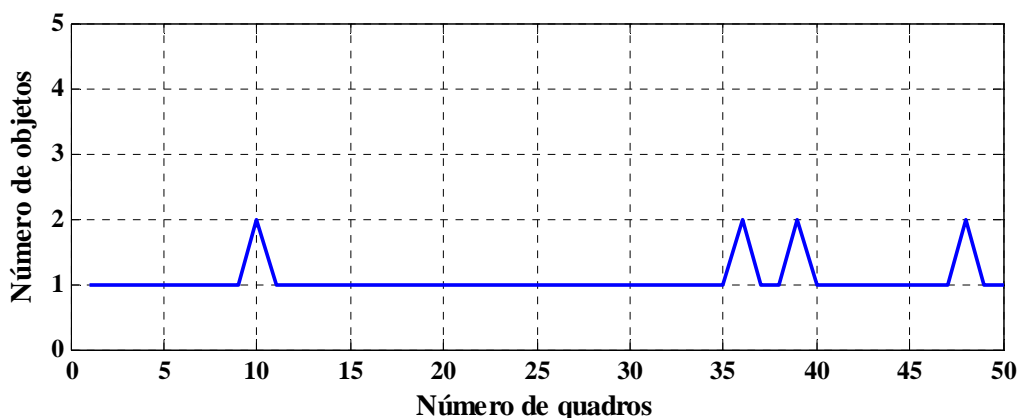


Figura 15 - Número de objetos detectados em uma sequência de vídeo, usando o YCbCr.

No espaço de cor YIQ, foram usados os mesmos limiares que os do YCbCr, pois este também separa a imagem em luminância e duas crominâncias. Para a componente Y foi usado um limiar de 30 e para as componentes I e Q um limiar de 5. Como mostra a Figura 16, o número máximo de objetos encontrados nas imagens da sequência de vídeo também foi dois, ocorrendo em vários quadros da sequência.

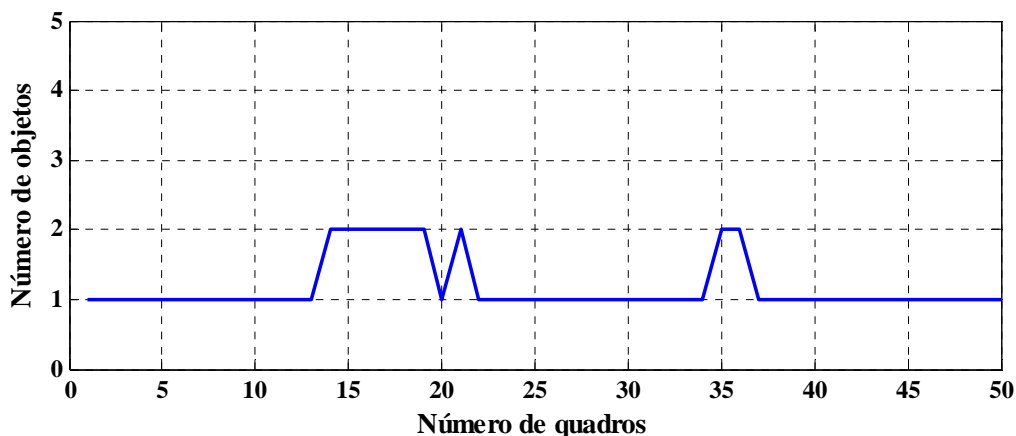


Figura 16 - Número de objetos detectados em uma sequência de vídeo, usando o YIQ.

Nos testes com o espaço de cor HSV, foi usado o mesmo limiar de 30 para todas as componentes. Neste caso foram obtidos os piores resultados, pois o número de objetos encontrados em cada quadro da sequência foi muito elevado (ver Figura 17). As causas desse elevado número de segmentos encontrados são o valor alto usado no limiar para a eliminação dos ruídos, que por sua vez resultou na segmentação do objeto em movimento em vários objetos menores.

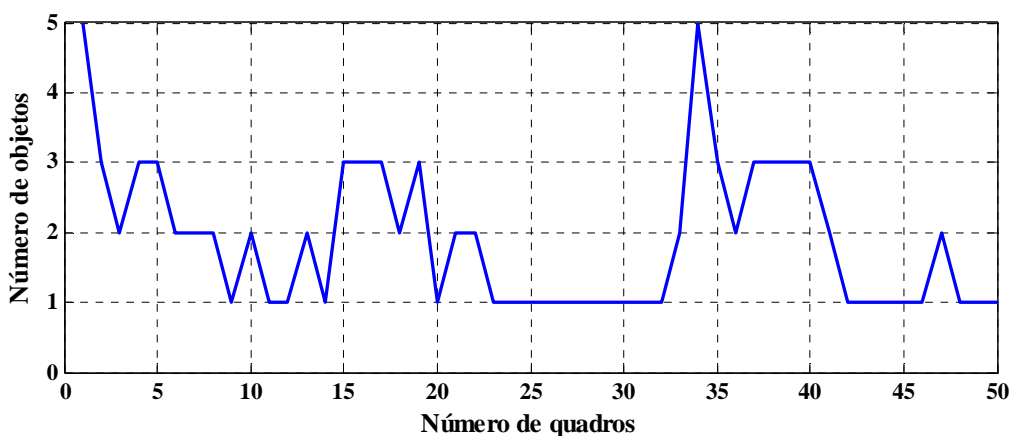


Figura 17 - Número de objetos detectados em uma sequência de vídeo, usando o YCbCr.

Para se evitar falsas detecções ou perda de partes ou até mesmo de todo o objeto em movimento é necessário se utilizar um bom espaço de cor no processo de detecção de movimento. Com base nos resultados obtidos, o espaço de cor YCbCr se mostrou o mais indicado para realizar a detecção de movimento. Este espaço de cor foi escolhido por ter sido o que apresentou menor variação do número de objetos encontrados na sequência de vídeo, detectando na maioria das vezes o número ideal de objetos.

3.3.4 Limiares de ruído

Para determinar o limiar de ruído que será utilizado no algoritmo DMFMorf foi avaliado o número de objetos em uma cena variando o valor do limiar entre 0 e 50. Para fins de comparação de limiar utilizou-se uma imagem de fundo (Figura 18(a) - IMG1) e uma imagem com um novo objeto neste fundo estático (Figura 18(b) - IMG2). Foram determinados os limiares de ruído para os espaços de cores RGB, YCbCr, YIQ e HSV, porém nesta subseção, somente será tratado a determinação dos limiares do espaço de cor YCbCr, escolhido para realizar a detecção de movimento. Os limiares de ruído da luminância e crominância foram obtidos separadamente, uma vez que estes valores precisam ser diferentes devido às características de cada componente.



(a) Imagem de fundo (IMG1)



(b) Imagem atual (IMG2)

Figura 18 - Imagens utilizadas para a determinação do limiar de ruído.

O número de objetos encontrados em uma cena varia muito com relação ao limiar de ruído aplicado, conforme é mostrado pela Figura 19. Nota-se que para limiares baixos o número de objetos encontrados é alto, isso se deve a grande quantidade de ruídos que não foram filtrados. Conforme o limiar cresce o ruído diminui e consequentemente o número de objetos encontrados também. Porém quando o limiar é muito alto o número de objetos binarizados cresce e a segmentação do objeto tornasse ineficiente, pois um mesmo objeto pode ser separado em vários pedaços. Os limiares de ruído altos também fazem com que a área do objeto segmentado diminua, pois quanto maior o limiar maior será a variação na dos *pixels* tolerada pelo algoritmo, em consequência maior será a probabilidade do um objeto em movimento ser confundido como fundo da imagem. Nos piores casos, é possível que limiares muito elevados resultem na não detecção de objetos em movimento em uma cena.

Para o limiar de ruído da luminância, os melhores resultados obtidos para à eliminação de ruído e qualidade da segmentação do objeto estão na faixa de 24 a 31 (ver Figura 19).

Como na luminância se encontram grandes variações de iluminação, projeção de sombra e reflexos, o limiar adotado foi de 30.

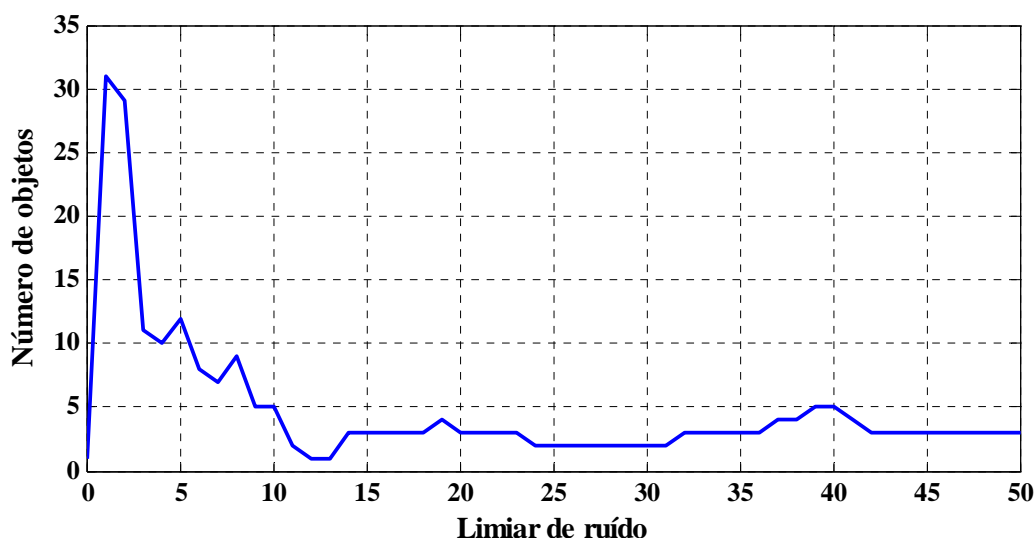


Figura 19 - Determinação do limiar de ruído para a luminância. Utilizado como referência uma imagem de fundo (IMG1) e uma imagem com movimento (IMG2).

Para a crominância os melhores resultados de limiar de ruído ficaram na faixa de 4 a 7 (ver Figura 20). Como é possível observar na Figura 20, a partir do limiar 25 não há identificação de nenhum objeto. Isso se deve ao fato de a crominância variar muito menos que a luminância, fazendo com que para limiares muito altos de ruído o algoritmo não identifique os objetos causadores de movimento nas componentes Cb e Cr que representam a crominância da imagem. Considerando a melhor segmentação do objeto, utilizou-se um limiar para a crominância de valor 5.

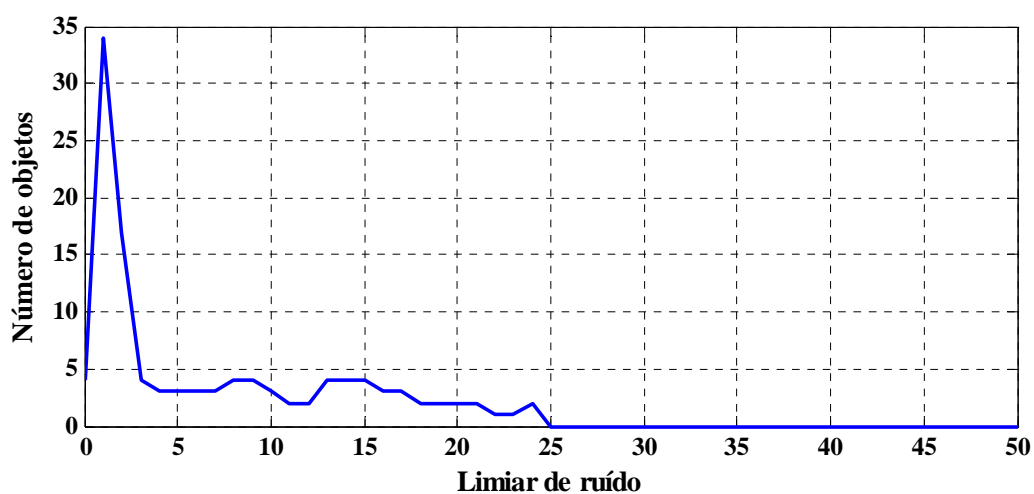


Figura 20 - Determinação do limiar de ruído para a crominância. Utilizado como referência uma imagem de fundo (IMG1) e uma imagem com movimento (IMG2).

3.3.5 Tamanho do elemento estruturante utilizado na filtragem morfológica

O elemento estruturante usado na filtragem morfológica tem formato de disco. O raio deste elemento é que vai determinar o resultado das operações de erosão e dilatação responsáveis pela filtragem morfológica, sendo este um fator fundamental no desempenho do algoritmo DMFMorf que utiliza esse tipo de filtragem. O tamanho do elemento estruturante interfere diretamente no desempenho da filtragem morfológica quando aplicado na redução do ruído como na preservação da coesão do objeto em movimento. Com o objetivo de determinar o tamanho mais adequado do elemento estruturante, foi realizado um estudo do número de objetos encontrados em uma cena em relação ao tamanho do raio deste elemento. Neste estudo utilizou-se uma imagem de fundo [Figura 18(a) - IMG1] como referência para a detecção de movimento do DMFMorf e uma imagem com um novo objeto neste fundo estático [Figura 18(b) – IMG2]. O raio do elemento estruturante foi determinado para os espaços de cores RGB, YCbCr, YIQ e HSV, porém nesta subseção somente será descrito a determinação do raio para o YCbCr. Os limiares de ruído utilizados (30 - luminância e 5 – crominância) foram aqueles obtidos na etapa anterior. As imagens binarizadas das 3 componentes de cor Y, Cb e Cr foram combinadas através de uma operação lógica OU *pixel-a-pixel*. Na imagem resultante desta operação foi aplicada a filtragem morfológica com os diferentes raios do elemento estruturante.

Conforme é mostrado na Figura 21, um elemento estruturante com raio mínimo (raio igual a 1) não é suficiente para filtrar os ruídos existentes na imagem binária. Raios maiores podem causar falha na segmentação do objeto, fazendo com que um único objeto seja entendido pelo algoritmo como se fosse dois ou mais objetos, conforme ilustrado na Figura 21 para um raio de 6 *pixels*. Visando uma adequada redução de ruído e segmentação do objeto, foi adotado um elemento estruturante circular com raio de 3 *pixels*.

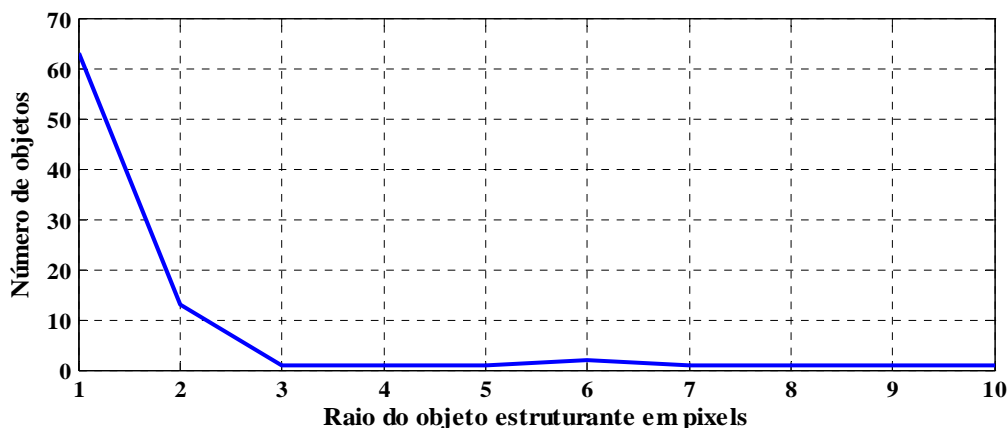


Figura 21 - Determinação do tamanho do elemento estruturante utilizado para a filtragem morfológica. Utilizado como referência uma imagem de fundo (IMG1) e uma imagem com movimento (IMG2).

3.4 Detecção de alterações no fundo

O fundo é uma imagem estática utilizada como referência para a detecção de movimento. A partir desta imagem será subtraída a imagem atual para detectar, através da técnica de subtração de fundo, o movimento em uma cena. A adequada detecção de movimento depende de uma imagem de fundo correta.

A Figura 22 ilustra um exemplo de alteração de cenário, na qual um objeto estático que estava incorporado ao ambiente foi removido do cenário 1. Se esta mudança de ambiente se prolongar por algum tempo, é necessário que ela seja adequadamente tratada para evitar que seja considerada como movimento e assim resultar na gravação de sucessivos quadros numa cena que é estática. Para tal, apenas o momento da retirada deve ser gravado, sendo em seguida assumido um novo modelo de ambiente (cenário 2), pois existe uma diferença entre o primeiro e o segundo cenário.

No DMFMed o algoritmo que implementa a detecção de alteração de fundo é chamado de Detecção de Fundo Inteiro (DFI). Este algoritmo detecta a ausência de movimento na imagem inteira, atualizando toda a imagem de fundo. Porém, no decorrer das avaliações foi percebido que ocorrem às vezes alterações de objetos estáticos em cenas com movimento, que não eram adequadamente tratadas pelo algoritmo DFI. Essa deficiência é devido à existência de movimentos em uma cena impedindo que o DFI altere o fundo. Não alterando a imagem de fundo o algoritmo de detecção de movimento vai detectar erroneamente um objeto estático como sendo um objeto em movimento. Por este motivo, o algoritmo utilizado no DMFMorf divide esta imagem em blocos menores, detectando a ausência de movimento e

atualizando o fundo em cada bloco. Esse algoritmo é denominado de Detecção de Fundo por Blocos (DFB).



Figura 22 - Exemplo de alteração de cenário da imagem de fundo.

3.4.1 Detecção de Fundo Inteiro (DFI)

Para detectar a necessidade de mudança de fundo foi utilizada a técnica da diferença temporal entre imagens apresentada no diagrama de blocos da Figura 23. Através desta técnica é possível detectar a inexistência de movimento. Este método é semelhante ao usado na detecção do movimento do DMFMorf, diferindo deste na primeira etapa do processo, na qual a subtração é entre o quadro atual e o quadro anterior da imagem. Esta técnica permite detectar de modo eficiente que o fundo foi alterado.

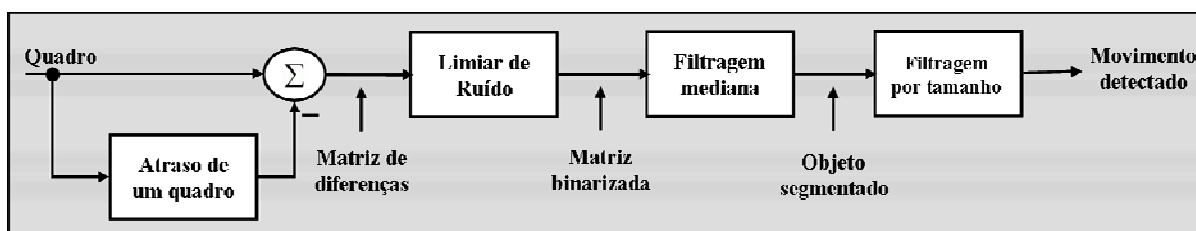


Figura 23 - Diagrama de blocos do método de detecção de mudança de fundo pela diferença temporal entre quadros.

Esse algoritmo de detectar alterações no fundo consiste das seguintes etapas: 1) é feita a subtração entre a imagem atual e a imagem anterior a ela, 2) é aplicado um limiar de ruído, 3) é feita uma filtragem mediana, 4) é aplicada uma filtragem pelo tamanho do objeto, na qual se a imagem binarizada apresentar variação menor ou igual a 10 *pixels* em relação ao quadro anterior considera-se que não há movimento na cena. Se for detectada a ausência de movimento em cinco quadros consecutivos, é realizada a atualização do fundo. Isto faz com que mudanças na cena que duram mais que 2 segundos sejam detectadas e incorporadas ao

fundo estático. Estas mudanças podem ser tanto mudanças de objetos estáticos no cenário como variações de iluminação.

3.4.2 Detecção de Fundo por Blocos (DFB)

A detecção do fundo por blocos é realizada dividindo-se a imagem em 16 blocos (4 colunas e 4 linhas de blocos). Estes blocos são analisados separadamente através da função ALTERA_FUNDO. Esta função também utiliza a diferença temporal entre imagens. Quando detectado a ausência de movimentos em um ou mais blocos o fundo correspondente ao mesmo é atualizado. Essa característica permite que mesmo em cenas nas quais haja movimento em alguns dos blocos, os demais possam ser atualizados, evitando detectar erroneamente um objeto estático como sendo um objeto em movimento.

3.5 Gravação dos quadros com movimento

É importante gravar em mídia digital as imagens nas quais foram detectados movimentos significativos devido à necessidade posterior de análise de eventos. Esta gravação deve ser apenas dos eventos isolados para facilitar a visualização e análise das imagens por operadores humanos, reduzindo assim a fadiga dos vigilantes encarregados de observar diversos monitores de imagem. A análise humana das imagens gravadas ou transmitidas não foi dispensada no sistema proposto, uma vez que a observação humana ainda é o processo mais confiável para identificação do tipo de objeto visual.

Uma vez determinado que um quadro de imagem deve ser gravado, a informação de data e hora é sobreposta a esta imagem de modo a permitir posteriormente saber quando o evento ocorreu. Para facilitar a análise das imagens armazenadas por observadores humanos, utiliza-se uma moldura em torno dos objetos em movimento conforme mostrado na Figura 24(b). Para facilitar a identificação dos arquivos, o seu nome tem a data e hora em que foi armazenado o primeiro quadro, e através da organização das pastas, os arquivos de vídeo são separados por mês e ano.

Os sistemas VEDAM1 e VEDAM2 além de salvar os quadros detectados com movimento, também salvam um quadro anterior e um posterior ao movimento, de modo que se pode sempre verificar as alterações que ocorreram na cena. Todas as mudanças de imagem de fundo também são gravadas.

3.5.1 Inserção de moldura

No VEDAM1 a inserção de moldura é feita de tal forma que apenas é possível identificar através de moldura um objeto causador de movimento por imagem. Este algoritmo foi denominado Inserção de Moldura Única (IMU). No IMU se existir mais de um objeto na cena, a moldura será inserida em torno de um retângulo mínimo que todos os objetos juntos ocupam [ver Figura 24(a)].

Para solucionar este problema, foi desenvolvido para o VEDAM2 um algoritmo para inserção de moldura em vários objetos por imagem. Este algoritmo é denominado Inserção de Moldura Múltipla (IMM). A moldura neste algoritmo é inserida no retângulo mínimo que cada objeto detectado ocupa [ver Figura 24(b)]. A inserção de moldura distinguindo cada objeto facilita ainda mais a observação e identificação dos objetos causadores de movimento por observadores humanos.

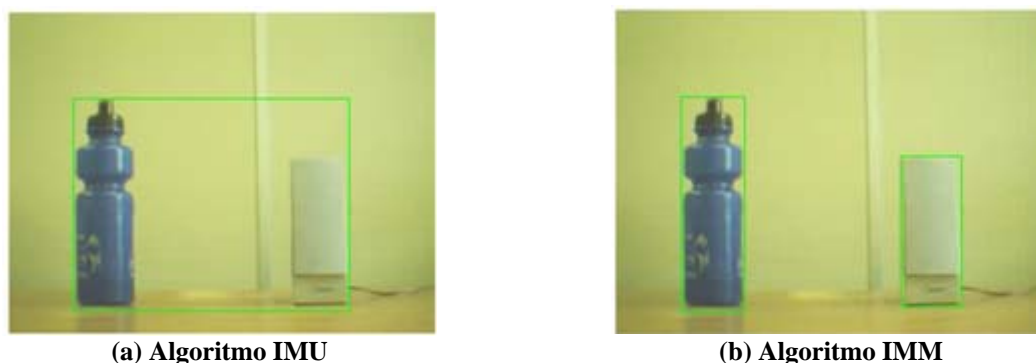


Figura 24 - Exemplo de inserção de moldura.

3.5.2 Compressão de vídeo

No VEDAM1 a seqüência de quadros com movimento é armazenada em arquivo com formato de container de vídeo AVI (*Audio and Video Interleave*) usando o codec de compressão de vídeo Indeo5. Este codec de vídeo apresenta uma taxa de compressão de vídeo baixa.

A informação de vídeo normalmente necessita de uma grande taxa de bits para ser armazenada. O sistema desenvolvido necessita armazenar em mídia digital todas as cenas detectadas com movimento. Desta forma, faz-se necessário a utilização de um codec de compressão de vídeo mais eficiente, que possibilite um armazenamento mais compacto do vídeo. Neste trabalho foram testados e comparados diversos codecs de vídeo. Para possibilitar a comparação entre os codecs de vídeo, foi realizada a aquisição de uma seqüência de vídeo

com movimento totalizando 368 quadros. Esta sequência foi gravada em uma taxa de 3 quadros por segundo, sendo equivalente a 120 segundos de gravação em ambiente interno. Depois de realizada a detecção de movimento, a sequência a ser gravada se resume a 82 segundos. Esta grande quantidade de quadros se deve a predominância de movimentação nos quadros desta sequência especialmente criada para teste da compressão. A resolução da imagem é de 240x320 e o tipo de container utilizado é o AVI. A Tabela 1 mostra os resultados de taxa de compressão e tamanho do arquivo obtido com cada tipo de codecs de vídeo. Foram avaliados o arquivo sem compressão, e os codecs XVID, Indeo5, Indeo3, Cinepak, IYUV e DIVX. A taxa de compressão de todos os codec foi obtida comparando seu tamanho em relação ao vídeo sem compressão.

Tabela 1 - Comparação entre diferentes tipos de codecs de compressão de vídeo

Tipo de codec	Tamanho (MB)	Taxa de compressão
Sem compressão	54,5	1:1
XVID	0,655	83,2:1
Indeo5	3,1	17,6:1
Indeo3	2,47	22,1:1
Cinepak	5,41	10,1:1
IYUV	27,2	2:1
DIVX	3,43	15,9:1

Analisando a Tabela 1 é possível ver que o codec Indeo5, utilizado no DMFMed, necessita de 3,1 MByte para armazenar a sequência em questão, apresentando uma taxa de compressão de aproximadamente 17:1. O codec que obteve o melhor resultado em termos de redução do tamanho dos arquivos foi o XVID que apresentou taxa de compressão de aproximadamente 83:1, usando 655 kbytes.

Devido a esta significativa redução de do tamanho dos arquivos das sequências de vídeo, o XVID foi escolhido como o codec de compressão de vídeo a ser utilizado no VEDAM2.

Quanto à qualidade visual das sequências armazenadas, o Indeo5 e o XVID apresentam resultados bem próximos, sendo que em ambos os casos é possível distinguir bem, após a compressão, as informações de data e moldura inseridas na imagem.

3.6 Ajuste da exposição à luz da câmera

A exposição à luz de uma câmera diz respeito à quantidade de luz que esta deixa entrar pela abertura de seu diafragma até o seu sensor. Câmeras do tipo webcam são de uso preferencial em ambientes internos que possuam iluminação controlada, apresentando grande sensibilidade à luz em ambientes externos. Este tipo de câmera apresenta problemas quando a exposição à luz da mesma é muito baixa (ambientes escuros) e quando a exposição é muito alta (ambientes muito claros).

Para os testes, a webcam utilizada (CAM2) permite controlar a exposição à luz entre 1 e 100. Este parâmetro permite ajustar a entrada de luz no sensor da câmera. No sistema VEDAM1 este valor é predeterminado antes do início da captura de imagens e é mantido constante independentemente de alterações que ocorressem na iluminação do ambiente. No sistema VEDAM2, para resolver os problemas de exposição à luz da câmera, foi desenvolvido um algoritmo que possibilita detectar e alterar a exposição à luz da câmera em qualquer momento da captura. Este algoritmo é denominado “Variação da Exposição à Luz” (VEL). Com este algoritmo se for detectado que o ambiente está pouco iluminado, a exposição à luz é aumentada gradativamente até ser obtida uma boa qualidade de imagem ou até ser atingido o valor máximo possível. Para ambientes muito iluminados o processo é inverso, se for detectado excesso de iluminação, a exposição à luz é diminuída até ser obtida uma boa qualidade de imagem ou até atingir o valor mínimo possível. Para determinar a qualidade da imagem em relação à exposição à luz da câmera, o VEL calcula o valor médio de cada *pixel*, considerando como ideal os valores médios entre a faixa de 100 a 200. Para valores médios abaixo de 100 é considerado que a imagem de modo geral está muito escura e a exposição à luz precisa ser aumentada. Para valores médios acima de 200 é considerado que a imagem está muito clara e a exposição à luz precisa ser diminuída.

4 Resultados

Neste capítulo serão apresentados os resultados mais significativos alcançados com os sistemas descritos no Capítulo 3. Estes sistemas foram analisados quanto ao seu desempenho em ambientes internos e externos, considerando como critérios de análise a qualidade da segmentação do objeto e a quantidade de ruído não filtrado. Para possibilitar a comparação entre os dois algoritmos, foram capturadas sequências de vídeo as quais foram processadas tanto com o VEDAM1 quanto com o VEDAM2.

4.1 Detecção de movimento em ambiente interno

Em ambientes internos foram obtidos bons resultados quanto à redução de alarmes falsos tanto para o VEDAM1 como para o VEDAM2 (ver Figura 25). Neste caso, devido à iluminação ser constante e ser geralmente proveniente do teto, a projeção de sombras tende a ser menor, havendo também menor variação de luminosidade. Apesar disso, devem ser consideradas algumas situações que podem causar falsas detecções de movimento, como por exemplo, a mudança brusca na iluminação do local causada pelo acendimento ou desligamento de uma lâmpada, ou a abertura de alguma porta ou janela. Ao longo do tempo também podem ocorrer modificações no ambiente devido à remoção ou inserção de objetos no ambiente, nestes casos estas alterações são detectadas pelo sistema e incorporadas ao modelo do ambiente, através da troca da imagem de fundo.

O VEDAM1 apresentou resultados satisfatórios para este tipo de ambiente quanto a eliminação de ruídos que poderiam ocasionar alarmes de detecção falsos. Porém devido ao seu alto valor de limiar de ruído os objetos não são corretamente segmentados (ver Figura 25).

No caso do VEDAM2 os resultados obtidos para ambiente interno são melhores que os alcançados com o VEDAM1 tanto na segmentação dos objetos como na eliminação de ruídos como pode ser visto na Figura 25.



Figura 25 - Exemplo de movimento captado em ambiente interno.

4.2 Detecção de movimento em ambiente externo

Nos ambientes externos também foram obtidos resultados satisfatórios (ver Figura 26), porém neste tipo de ambiente é necessária uma maior preocupação com os alarmes falsos. As variações de luminosidade devido à hora do dia ou até mesmo variação das condições climáticas, tais com a presença de nuvens reduzindo a iluminação natural podem afetar o desempenho do sistema. Neste caso também se está sujeito à ação do vento que pode resultar, em movimentação da vegetação, que são movimentos que o sistema não deseja detectar, a ocorrência de reflexos de luz em superfícies lisas que podem muitas vezes ser detectadas como movimento e o aumento da projeção de sombras. Também se devem levar em conta as alterações de objetos visuais tais como: móveis, abertura de janelas e portas, veículos, os quais produzem variações que devem ser incorporadas ao modelo de fundo, tornando a

alteração da imagem de fundo mais frequente.

O tipo de câmera de vídeo usada também passa a ser um fator importante. As câmeras do tipo webcam são de uso preferencial para ambientes internos, podendo apresentar em ambientes externos uma sensibilidade excessiva à luz devido a abertura de seu diafragma que permite uma grande entrada de luz até o sensor. Outro problema com relação à luminosidade captada pela câmera é a utilização da mesma em ambientes pouco iluminados. Este problema não diz somente respeito a ambientes externos, mas está mais associado a este que ao interno devido a maior dificuldade de controle de iluminação deste ambiente em relação ao outro. No sistema VEDAM2 através do algoritmo VEL consegue-se contornar este problema, variando a exposição à luz da câmera durante a captura das imagens.



(a) Fundo



(b) Imagem atual



(c) Objeto segmentado com o VEDAM1



(d) Objeto segmentado com o VEDAM2



(e) Imagem armazenada com o VEDAM1



(f) Imagem armazenada com o VEDAM2

Figura 26 - Exemplo de movimento captado em ambiente externo com o uso do VEDAM1 e VEDAM2.

Conforme pode ser visualizado na Figura 26, o sistema VEDAM2 apresenta melhor resultado que o VEDAM1 para ambientes externos. Este sistema segmenta melhor os objetos causadores do movimento e também elimina melhor os ruídos. Outro ponto positivo do VEDAM2 é a inserção de moldura para identificação do objeto causador do movimento [ver Figura 26(e) e 22(f)], sendo possível neste sistema envolver com moldura isoladamente mais de um objeto por imagem enquanto no VEDAM1 isso não é possível.

4.3 Comparação dos métodos DMFMed e DMFMorf

Os métodos DMFMed e DMFMorf foram comparados quanto a segmentação do objeto causador do movimento e a eliminação de ruídos.

Quanto a segmentação do objeto causador de movimento, o DMFMorf apresenta desempenho superior ao DMFMed, como é possível verificar na Figura 27. Nota-se na imagem (c) que o DMFMed apresentou muita falha na segmentação, não conseguindo definir o contorno do objeto corretamente enquanto o DMFMorf conseguiu definir todo o contorno deste objeto, apresentando apenas uma pequena falha de segmentação dentro do mesmo [ver Figura 27(d)].

O DMFMed apresenta resultados satisfatórios quanto a segmentação do objeto, porém devido ao seu alto valor de limiar de ruído, quando o objeto causador do movimento tem características de luminosidade e cor semelhantes ao fundo a segmentação é bastante falha. Isto faz com que boa parte do objeto não seja detectado [ver Figura 27(c) e (e)].

A segmentação do DMFMorf apresenta bons resultados [ver Figura 27(d) e (f)]. Com o DMFMorf são obtidos resultados satisfatório quanto a segmentação do objeto mesmo quando fundo e objeto têm características de luminância e cor muito próximas. Isso se deve a melhor filtragem das imagens que possibilitou a redução dos limiares de ruído.

As falhas na segmentação do objeto tanto da Figura 27(c) como na da Figura 27(d) são causadas principalmente pelo limiar de ruído que precisa ser aplicado. Quanto maior este limiar, maior será a variação dos *pixels* da imagem atual em relação ao fundo que o algoritmo irá tolerar.

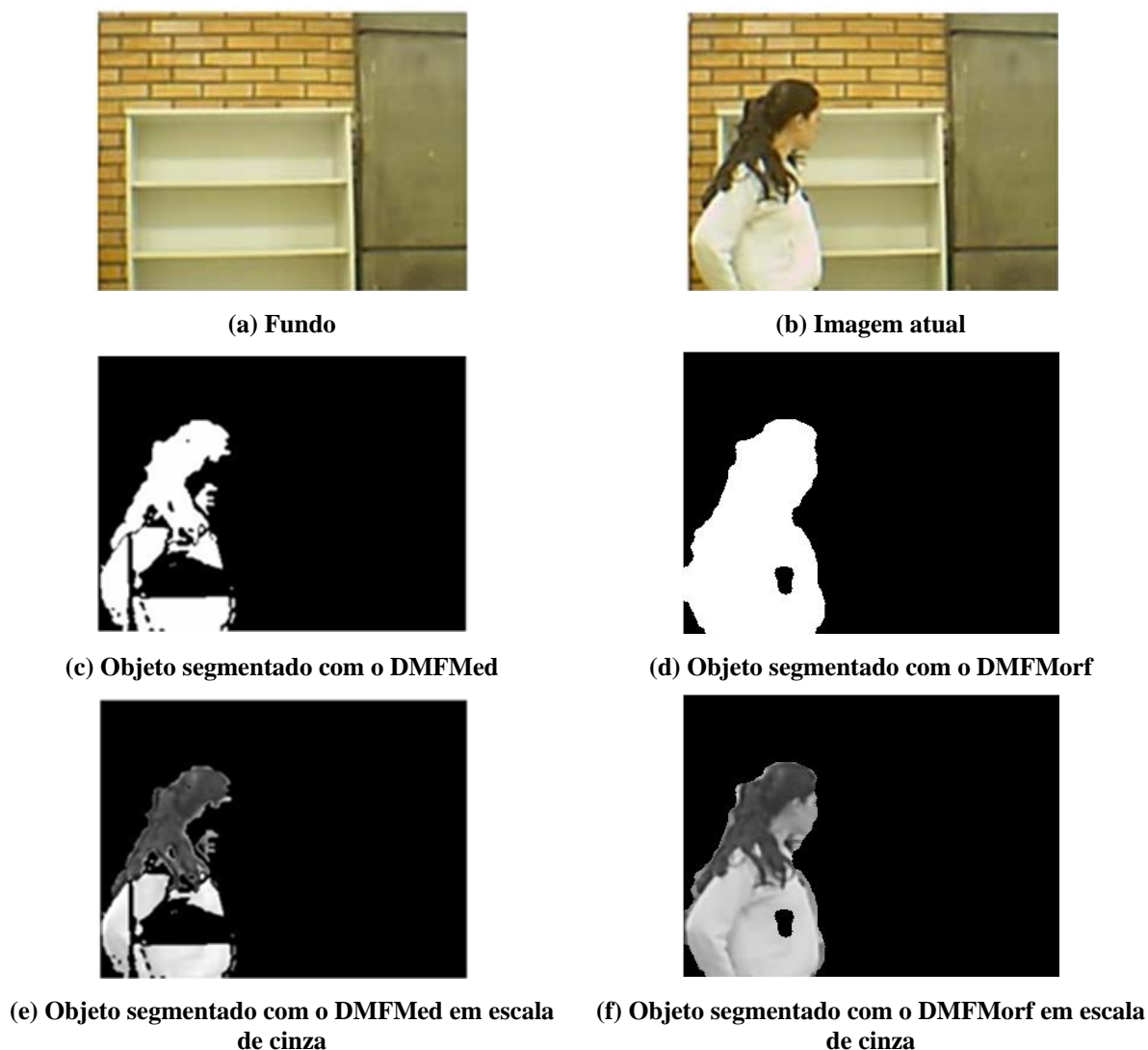


Figura 27 - Comparação de desempenho entre o método DMFMed e DMFMorf.

Para fins comparativos, o objeto causador de movimento apresentado na Figura 27(b) foi segmentado manualmente no *software* de edição de imagens GIMP (*GNU Image Manipulation Program*). Na Figura 28 é possível o objeto segmentado com este *software*. A área do objeto segmentado com o GIMP é de exatamente 15203 *pixels* enquanto a área do mesmo objeto segmentado como DMFMed é de 10319 *pixels* e como o DMFMorf é de 15473 *pixels*. A área do objeto segmentado com o DMFMed é consideravelmente menor que as outras duas, pois o objeto possui áreas internas que foram confundidas como o fundo estático. Com o DMFMorf a área do objeto segmentado ficou maior que a área da imagem segmentada como o GIMP devido as operações morfológicas que o mesmo foi submetido. Quando uma operação de dilatação é realizada é possível que regiões do contorno do objeto que estão separadas por uma distância menor que o raio do objeto sejam fechados. Como o

elemento estruturante usado é um disco, o contorno destas partes do objeto tende a ficar mais arredondado.



(a) Objeto segmentado



(b) Objeto segmentado em escala de cinza

Figura 28 - Objeto segmentado com o GIMP.

4.4 Comparação dos algoritmos DFI e DFB

Na Figura 29 são apresentados os resultados com os algoritmos DFI e DFB. Na imagem (a) é apresentada a imagem de fundo obtida com o DFI, nesta imagem é possível notar que um objeto que já não existe mais na imagem atual (a) ainda é considerado como parte do fundo estático. Nesta imagem atual (c) há apenas a presença de um objeto em movimento, sendo este localizado no lado esquerdo da imagem. A presença deste objeto faz com que seja detectado o movimento nesta região, como pode ser visto na imagem (d). Na imagem (b) é apresentada a imagem de fundo do DFB, na qual foi atualizada a imagem de fundo da direita onde estava o objeto que havia sido retirado. Em consequência, na detecção de movimento somente o objeto da imagem (c) é detectado [ver Figura 29(e)].

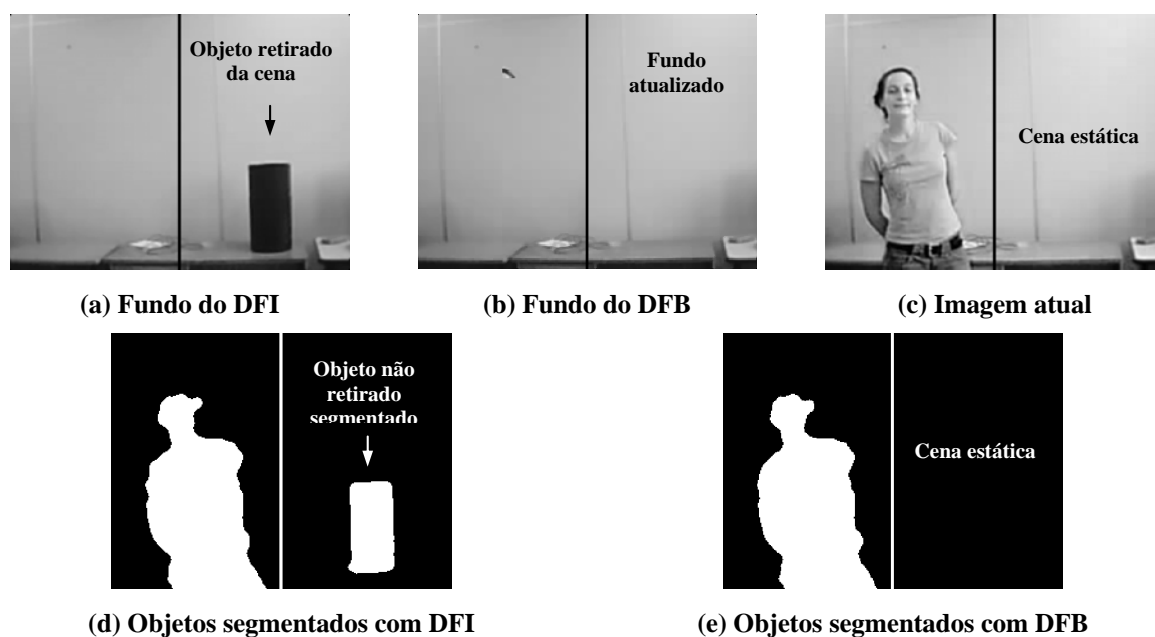


Figura 29 - Conseqüência da atualização da imagem de fundo com os algoritmos DFI e DFB.

4.5 Comparação entre IMU e IMM

Os algoritmos de inserção de moldura visam facilitar a identificação dos objetos causadores de movimento em uma cena. O algoritmo IMM resultou em uma melhor visualização dos objetos em movimento facilitando a sua identificação por observadores humanos, principalmente quando existem muitos objetos na cena, ou aparecem objetos pequenos. Isto ocorre porque o IMM permite identificar com moldura individual cada objeto em uma cena, enquanto com o IMU é possível fazer a inserção de apenas uma moldura por cena. Neste caso quando existe mais que um objeto na cena, a moldura inserida corresponde ao retângulo mínimo que todos os objetos juntos ocupam.

4.6 Gravação de imagens com o codec Indeo5 e XVID

Quanto ao armazenamento das imagens, com o sistema VEDAM1 que utiliza o codec de compressão de vídeo Indeo5, o sistema necessita de 13 kBytes em média para armazenar cada imagem com resolução de 320×240 pixels. São armazenadas as mudanças de fundo e os quadros com movimento a uma taxa máxima de 3 quadros por segundo. Cada arquivo de

imagem contém no máximo 200 quadros, sendo nomeado pela data e hora do primeiro quadro.

No VEDAM2 foi utilizado o codec de vídeo XVID, pois o mesmo apresenta significativa redução do tamanho dos arquivos das sequências de vídeo em relação ao Indeo5. Com o XVID, a taxa de compressão em relação ao vídeo sem compressão é de aproximadamente 83:1, diminuindo o tamanho do arquivo de um vídeo de 54,5 MBytes para 655 kBytes. Já o codec utilizado no VEDAM1, o Indeo5, apresenta uma taxa de compressão em relação ao vídeo original de aproximadamente 17:1. A taxa de compressão do XVID em relação ao Indeo5 é de aproximadamente 5:1.

No XVID um quadro de 320×240 *pixels* usa aproximadamente 2,6 kBytes no arquivo. Cada arquivo armazena no máximo 900 quadros a uma taxa de quadros por segundo igual a 3, gerando arquivos de vídeo de no máximo 5 minutos de duração. Para armazenar estes 5 minutos de vídeo são necessários 2,4 MBytes utilizando o XVID e 11,7 MBytes utilizando o Indeo5.

5 Conclusões e Trabalhos Futuros

Neste trabalho foi descrita a implementação de sistemas de vigilância eletrônica automática com detecção de movimento baseado apenas no sinal de vídeo. Nos dois sistemas foram usadas imagens adquiridas através de câmeras de baixo custo, com o objetivo de viabilizar o desenvolvimento futuro de um produto barato que possa ser utilizado em ambientes públicos, principalmente escolas municipais e estaduais.

O primeiro sistema é denominado VEDAM1 e foi desenvolvido durante o biênio 2007-2008 como projeto de pesquisa. Este sistema faz a detecção de movimento através de filtragem mediana, utilizando o método DMFMed. Este sistema foi aperfeiçoado durante o período de trabalho de conclusão de curso e deu origem ao VEDAM2. O VEDAM2 detecta o movimento através de filtragem morfológica, utilizando o método DMFMorf. Os dois sistemas conseguem detectar de modo eficiente o movimento, porém o VEDAM2 apresenta melhores resultados que o VEDAM1 quanto à eliminação de ruídos e segmentação do objeto causador do movimento.

Estes sistemas também necessitam substituir frequentemente a sua imagem de fundo, que é utilizada como referência na detecção de movimento, de modo que a adequada detecção de movimento depende de uma imagem de fundo correta. O VEDAM1 utiliza o algoritmo DFI, fazendo a alteração da imagem de fundo inteira. O VEDAM2 usa o algoritmo DFB, fazendo a alteração da imagem de fundo em blocos menores. Esta característica permite que mesmo em cenas nas quais haja movimento em alguns dos blocos, os demais possam ser atualizados, evitando detectar erroneamente um objeto estático como sendo um objeto em movimento.

Outro aspecto tratado nos sistemas foi a armazenagem das imagens. Visando facilitar a visualização e análise das imagens por operadores humanos, os sistemas apenas armazenam as imagens nas quais foram detectados movimentos significativos. Como resultado, minimiza-se também o tamanho dos arquivos de armazenamento. Outra consequência do uso destes sistemas é a redução da fadiga dos vigilantes encarregados de observar diversos monitores de imagem, pois como o mesmo não é mais necessário acompanhar

constantemente todas as imagens, mas apenas analisar os eventos de movimento detectados e sinalizados pelo sistema.

Para facilitar a identificação dos objetos causadores de movimento por agentes humanos, tanto o VEDAM1 como o VEDAM2 inserem molduras em torno destes objetos. A inserção de moldura feita através do algoritmo IMU no VEDAM1, este algoritmo consegue inserir apenas uma moldura por cena. No VEDAM2 a inserção de moldura é feita através do algoritmo IMM, que permite a inserção de moldura em torno de cada objeto de uma mesma cena, facilitando ainda mais a identificação dos objetos individuais.

No VEDAM1 e VEDAM 2 as sequências de imagens a serem armazenadas são comprimidas visando a diminuição do tamanho dos arquivos que estas utilizam. No VEDAM1 a compressão de vídeo é feita através do codec Indeo5. No VEDAM2 a compressão de vídeo é feita através do codec XVID. A compressão no VEDAM2 apresenta resultados melhores que a aplicada no VEDAM1, obtendo taxas de compressão de aproximadamente 83:1 enquanto que o VEDAM1 tem um desempenho 5 vezes inferior.

A ocorrência de falsos alarmes devido à movimentação de vegetação, sombras e variação de luminosidade tanto em ambiente interno como externo foi significativamente reduzida nos dois sistemas. Nos testes realizados notou-se que a grande maioria dos movimentos foi detectada com sucesso tanto em ambientes internos como externos. Situações nas quais movimentos de pessoas não foram detectados se resumem a cenários nos quais as pessoas estavam muito distantes da câmera, ou utilizavam roupas que se assemelhavam muito ao fundo da imagem. Para a validação completa dos dois sistemas ainda é necessário realizar alguns testes de campo, nos quais diversos movimentos de pessoas precisam ser analisados simultaneamente por observadores humanos e pelo sistema proposto.

Visando dar continuidade a este trabalho, são propostos alguns trabalhos que utilizam como base os sistemas desenvolvidos. Alguns exemplos de trabalhos que podem ser desenvolvidos a partir deste trabalho são:

- a transmissão das imagens detectadas com movimento através da rede local e internet (CANTU, 2008);
- implementação dos algoritmos de detecção e compressão em uma plataforma de hardware em DSPs ou FPGAs (PEDRONI, 2004);
- a integração deste sistema ao sistema de transmissão de imagens através da rede celular GSM/EDGE proposto por SALVADOR et. al. (2008);
- controle do movimento da câmera a partir das informações obtidas da imagem, de modo a acompanhar os objetos em movimento de modo automático (MOECKE, 2008);

- desenvolvimento de um algoritmo que permita a detecção de sombras (ELLIS et. al.);
- o desenvolvimento de um algoritmo que permita a identificação de pessoas, implementando a detecção de rosto;
- uso de zoom para identificação de transeuntes, assim como o registro de passagem;
- desenvolvimento de uma interface gráfica para o sistema.

Certamente o trabalho realizado não resultou em um produto pronto, mas o protótipo dos sistemas VEDAM1 e VEDAM2 implementados serão de grande importância tanto para a realização dos trabalhos futuros mencionados como para outros que a criatividade momentânea não permite vislumbrar.

Lista de Abreviaturas e Siglas

- AVI** – Áudio e vídeo entrelaçado (*Audio and Video Enterleave*)
- CMOS** – Semicondutor metal-óxido complementar (*complementary metal-oxide semiconductor*)
- CMYK** – Ciano, Magenta, Amarelo e Preto (*Cyan, Magenta, Yellow and black*)
- Codec – Codificador e decodificador
- DSP** – Processador de sinais digitais (*Digital Signal Processor*)
- DFB** – Detecção de Fundo por Blocos
- DFI** – Detecção de Fundo Inteiro
- DMFMed** – Detecção de Movimento através de Filtragem Mediana
- DMFMorf** – Detecção de Movimento através de Filtragem Morfológica
- EDGE** – Taxas de Dados Melhorada para a Evolução do GSM (*Enhanced Data Rates for GSM Evolution*)
- FPGA** – Arranjo de Portas Programáveis em Campo (*Field-Programmable Gate Array*)
- GSM** – *Global System for Mobile Communications*
- HSV** – Matiz, saturação e valor (*Hue, Saturation and Value*)
- IMM** – Inserção de Moldura Única
- IMU** – Inserção de Molduras Múltiplas
- Indeo3** – Indeo vídeo versão 3
- Indeo5** – Indeo vídeo versão 5
- LCD** – Monitor de cristal líquido (*Liquid Crystal Display*)
- MByte – Mega Byte
- MPEG** – Grupo de Especialistas em Imagem em Movimento (*Moving Pictures Experts Group*)
- NTSC** – Comitê Nacional do Sistema de Televisão (*National Television System(s) Committee*)
- RGB** – Vermelho, verde e azul (*Red, Green and Blue*)
- RLE** – *Run Length Encoding*
- VEDAM1** – Vigilância Eletrônica por Detecção Automática de Movimento 1
- VEDAM2** – Vigilância Eletrônica por Detecção Automática de Movimento 2
- Webcam** – câmera de vídeo de baixo custo que capta imagens transferindo-as diretamente para o computador
- YCbCr** – Luminância, desvio para o azul e desvio para o vermelho (*luma, blue-difference and red-difference*)
- YIQ** – Luminosidade, fase e quadratura (*Luma, in-phase and quadrature*)

Referências Bibliográficas

ALVARENGA, B.; MÁXIMO, A. Curso de física. Volume 2. São Paulo: Scipione, 2000. 414 p.

BROFFERIO, S.; CARNIMEO, L.; COMUNALE, D.; MASTRONARDI, G. A Background Updating Algorithm for Moving Object Scenes. In: CAPPELLINI, V. (ed.). Time-varying Image Processing and Moving Object Recognition, 2., Amsterdam, The Netherlands: Elsevier Publishers B.V., 1997. p. 289-296.

CANTU, E. Multicast de vídeo digital gerado por sistema de vigilância sobre rede TCP/IP. Projeto de pesquisa PIBITI, 2008.

CARVALHO, C.A.; MENEGUETTE, M. Junior; SILVA, E.A. Influência do espaço de cores na compressão JPEG de imagens orbitais. Revista brasileira de cartografia n° 55/01.

CEDRAS, C.; SHAH, M. Motion-based recognition: a survey. Image Vision Comput. v.13, p.129–155, 1995.

DIVX. DivX 7 Enjoy Free High-Quality Video Playback On Your Computer. Disponível em: < <http://www.divx.com/en/products/software/windows/divx> >. Acessado em: 22/02/2009.

ELLIS, T.; ROSIN. P.L. Image difference threshold strategies and shadow detection. Cardiff School of Computer Science, Reino Unido, 2000. Disponível em: < <http://users.cs.cf.ac.uk/Paul.Rosin/resources/papers/shadows.pdf> >. Acessado em: 07/03/2009.

A: FERGUSON, T.; MELANSON, M. Indeo Vídeo 3.2 (IV32) stream format for AVI and QT. Monash University, Austrália, 2001. Disponível em: <<http://www.csse.monash.edu.au/~timf/videocodec/indeo3.txt>>. Acessado em: 18/02/2009.

B: FERGUSON, T. Cinepak (CVID) stream format for AVI and QT. Monash University, Austrália, 2001. Disponível em: <<http://www.csse.monash.edu.au/~timf/videocodec/>>

cinepak.txt>. Acessado em: 18/02/2009.

FOURCC. YUV Formats. Disponível em: <<http://www.fourcc.org/yuv.php#IYUV>>. Acessado em: 19/02/2009.

GONZALEZ, R.C.; WOODS, R.E. Digital Image Processing. Prentice Hall, 2ª Ed. New Jersey, 2002. 812 p.

JONES, B.J. Low-cost outdoor video motion and non-motion detection. In: 29th Annual 1995 International Carnahan Conference on Security Technology, Sanderstead, Surrey Reino Unido. Proceedings... Surrey: IEEE, 1995. p. 376-381.

LIGOS CORPORATION. Indeo. Disponível em: <<http://ligos.com/index.php/home/products/indeo/>>. Acessado em: 15/02/2009.

LIGOS CORPORATION. Indeo Video and Audio Codecs Technical Documentation. Atualizado em 5 de abril de 2007. Disponível em: <http://www.ligos.com/pdf_docs/Indeo_doc.pdf>. Acessado em: 22/02/2009.

LÓPEZ M.T.; CABALLERO A.F.; FERNÁNDEZ M.A.; MIRA J.; DELGADO A.E. Visual surveillance by dynamic visual attention method. Pattern Recognition, v. 39, n. 11, p. 2194-2211, 2006.

MACHADO, A.M.L. Extração automática de contornos de edificações utilizando imagem gerada por câmera digital de pequeno formato e dado lidar. Tese. Universidade Federal do Paraná, Curitiba, 2006.

MOECKE, M. Controle de uma câmera pelo movimento detectado através de sinais de vídeo. Projeto de pesquisa PIBITI, 2008.

PRATT, W.K. Digital image processing: Paks Inside, 3 ed. New York: A.Wiley, 2001. 735 p.

PEDRONI, V.A. Circuit Design with VHDL. Cambridge: MIT Press, 2004. 363 p.

PINHEIRO, R.H.; LINS, R.C. Segmentação de vídeo baseada em movimento..XXV Simpósio Brasileiro de Telecomunicações, Recife: SBRT, 2007. DVD.

SALVADOR, C.J.; ROSA, P.H.; MEDEIROS, D.S.; SILVA, F.S.; NORONHA, M. Neto; MOECKE, M. Transmissão de seqüências de imagens através da rede celular GSM/EDGE.

II Jornada da Produção Científica da Educação Profissional e Tecnológica da Região Sul, Pelotas, 2008.

SAKAI, Y.; NAGASHIMA, H.; AMAIKE, M.; NAGASHIMA, H. Automatic surveillance camera equipment and alarm system. Patente US5448320, 1995.

SCURI, A.E. Fundamentos da imagem digital. Tecgraf/PUC-RIO, 1999.

SOUTO, R.P. Segmentação de imagem multiespectral utilizando-se o atributo matiz. Tese. INPE, São José dos Campos, 2003.

TOTH, D.; AACH, T.; METZLER, V. Illumination – invariant change detection. In: 4th IEEE Southwest Symposium Image Analysis and Interpretation, Proceedings. 2000. p. 3-7.

WANG L., HU W., TAN T., e MAYBANK S.. A survey on visual surveillance of object motion and behaviors. IEEE Trans. on Systems, Man and Cybernetics, vol. 3: p. 334-352, 2004.

XVID. General questions. Disponível em: < <http://www.xvid.org/FAQ.14.0.html> >. Acessado em: 22/02/2009.

YOUNG, M. The Pinhole Camera: Imaging without Lenses or Mirrors. The Physics Teacher, December, 1989. Disponível em: <<http://www.pinhole.com/archive/371>>. Acessado em: 06/03/2009.