

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Marcelo Felipe Maia Hor-Meyll Alvares

DEMULTIPLEXAÇÃO AUTOMÁTICA DE VÍDEO ASSÍNCRONO

Recife, 08 de abril de 2009

MARCELO FELIPE MAIA HOR-MEYLL ALVARES

DEMULTIPLEXAÇÃO AUTOMÁTICA DE VÍDEO ASSÍNCRONO

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para a obtenção do grau de **Mestre em Engenharia Elétrica**.

ORIENTADOR: Prof. RAFAEL DUEIRE LINS, Ph.D.

Recife, 08 de abril de 2009

© Marcelo Felipe Maia Hor-Meyll Alvares, 2009

A473m Alvares, Marcelo Felipe Maia Hor-Meyll.

Demultiplexação automática de vídeo assíncrono / Marcelo Felipe
Maia Hor-Meyll Alvares. – Recife: O Autor, 2009.

xii, 134 folhas, il : grafs., tabs., figs.

Dissertação (Mestrado) – Universidade Federal de Pernambuco.
CTG. Programa de Pós-Graduação em Engenharia Elétrica, 2009.

Inclui Bibliografia, Apêndices e Anexo.

1. Engenharia Elétrica. 2. Demultiplexação. 3. Automática.
4. Vídeo. 5. Perícia. 6. Criminal. I. Título.

CDD 621.3



Universidade Federal de Pernambuco

Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE
DISSERTAÇÃO DO MESTRADO ACADÊMICO DE

MARCELO FELIPE MAIA HOR-MEYLL ALVARES

TÍTULO

“DEMÚLTIPLEXAÇÃO AUTOMÁTICA DE VÍDEO ASSÍNCRONO”

A comissão examinadora composta pelos professores: RAFAEL DUEIRE LINS, DES/UFPE, VALDEMAR CARDOSO DA ROCHA JÚNIOR, DES/UFPE e GEORGE DARMITON DA CUNHA CAVALCANTI, CIN/UFPE sob a presidência do primeiro, consideram o candidato **MARCELO FELIPE MAIA HOR-MEYLL ALVARES APROVADO.**

Recife, 08 de abril de 2009.

EDUARDO FONTANA
Coordenador do PPGEE

RAFAEL DUEIRE LINS
Orientador e Membro Titular Interno

**GEORGE DARMITON DA CUNHA
CAVALCANTI**
Membro Titular Externo

VALDEMAR CARDOSO DA ROCHA JÚNIOR
Membro Titular Interno

Resumo da Dissertação apresentada à UFPE como parte dos requisitos necessários para a
obtenção do grau de Mestre em Engenharia Elétrica

DEMULTIPLEXAÇÃO AUTOMÁTICA DE VÍDEO ASSÍNCRONO

Marcelo Felipe Maia Hor-Meyll Alvares

Recife, 08 de abril de 2009

Orientador: Prof. Rafael Dueire Lins Ph. D.

Área de Concentração: Processamento Digital de Sinais

Palavras Chave: demultiplexação, automática, vídeo, perícia, criminal.

Número de páginas: 138

O tema desta dissertação é resultado do trabalho pericial em casos reais envolvendo vídeos provenientes de sistemas de vigilância bancários. Aqui é apresentada uma nova solução computacional em software para o problema de demultiplexação de vídeos se segurança multiplexados no tempo. A abordagem é particularmente interessante nos casos em que, por alguma razão, não é possível realizar a demultiplexação com o equipamento específico. Tendo como critério de classificação a luminância média dos quadros e explorando características como as componentes de baixa frequência das cenas, a abordagem apresentada vem sendo utilizada com bons resultados pelos peritos criminais no Brasil possibilitando a análise de vídeos em que se busca trechos de particular interesse gravados por sistemas analógicos multiplexados no tempo.

Abstract of Dissertation presented at UFPE as a partial fulfillment of the requirements for
the degree of Master in Electrical Engineering

AUTOMATIC ASYNCHRONOUS VIDEO DEMULTIPLEXING

Marcelo Felipe Maia Hor-Meyll Alvares

Recife, April 08, 2009

Supervisor: Prof. Rafael Dueire Lins Ph. D.

Area of Concentration: Digital Signal Processing

Keywords: demultiplexation, automatic, video, expert, forensics.

Number of pages: 138

The theme of this MSc. dissertation is the work on real forensics videos involving banking images. Its contribution is the development of a software approach to the problem of demultiplexing time multiplexed security videos. This approach is of particular interest in cases in which, for some reason, the regular demultiplexing hardware/software is not available. Based on comparisons of the mean frame luminance and having low frequency characteristics of the scenes as references, this approach is being used with good results by forensics video experts in Brazil to help locating sequences of particular interest in surveillance time multiplexed videos recorded in magnetic video cassettes.

Lista de Figuras

Figura 1.1	Exemplo de um quadro proveniente de um sistema de vídeo multiplexado no tempo.....	17
Figura 2.1	Fotografia do monitor do sistema de segurança usado no dia da prisão dos suspeitos para reproduzir a fita com a gravação da ação dos meliantes.....	21
Figura 2.2	Petrechos que foram encontrados com os meliantes e que estavam sendo utilizados para a perpetração do furto.....	22
Figura 2.3	Características da caixa de malotes (vista frontal).....	22
Figura 2.4	Detalhe da porta traseira da caixa de malotes (vista traseira).....	22
Figura 2.5	Abertura na face superior da caixa de malotes usada normalmente para a introdução dos malotes.....	23
Figura 2.6	Malote empresariam em lona sintética azul.....	23
Figura 3.1	Exemplo de um quadro híbrido (a) composto por dois campos (b e c) provenientes de câmeras distintas.....	29
Figura 3.2	Exemplo de zoom em uma região de um quadro entrelaçado, percebe-se que os campos apresentam imagens ligeiramente diferentes, da tecla F12 de um teclado sendo pressionada. Figura retirada da referência V.....	30
Figura 3.3	Uma das imagens que compõem “O cavalo galopante de Muybridge”.....	31
Figura 3.4	Esquema de comutação das câmeras no vídeo analisado.....	33
Figura 4.1	Diagrama em blocos do circuito integrado Bt848 da empresa Brooktree.....	35
Figura 4.2	Diagrama em blocos detalhado do circuito integrado Bt848, destaque para os dois blocos de entrada.....	36
Figura 4.3	Comparação entre os espectros de frequência dos sinais de: a) vídeo composto e b) S-Vídeo.....	37
Figura 4.4	Diagrama em blocos do decodificador e quantizador de sinais de vídeo do circuito integrado Bt848 da empresa Rockwell.....	38
Figura 4.5	Forma de onda típica de uma linha de retraço contendo informações VBI.....	40

Figura 4.6	Diagrama em blocos da porção de processamento digital do sinal de vídeo de um chipset da empresa conexant.....	41
Figura 4.7	Esquema de separação das componentes Y e C.....	44
Figura 4.8	Esquema de separação das componentes Y e C.....	44
Figura 4.9	Bloco de processamento do sinal de luminância.....	47
Figura 4.10	Detalhe de uma linha horizontal.....	47
Figura 4.11	Bloco de processamento do sinal de crominância.....	49
Figura 4.12	Esquemas de sub-amostragem de cor.....	51
Figura 4.13	Forma de onda típica de uma linha de um sinal de vídeo composto (ITU-R BT.470-7).....	58
Figura 4.14	Imagem proveniente de um sistema de multiplexação de vídeo que grava informações nas linhas do retraço vertical. Detalhe das linhas de retraço em zoom fora de proporção para facilitar sua visualização.....	59
Figura 4.15	Forma de onda típica de uma linha de um sinal de vídeo composto (ITU-R BT.470-7).....	60
Figura 4.16	Imagem proveniente de um sistema de multiplexação de vídeo que grava informações nas linhas do retraço vertical. Detalhe das linhas de retraço em zoom fora de proporção para facilitar sua visualização.....	62
Figura 5.1	Diagrama em blocos do CI SAA7118 da Philips, bastante similar ao Bt848 da Brooktree.....	64
Figura 5.2	Esquemas do sistema helicoidal de varredura em videocassetes: (esquerda) conjunto do cilindro e fita magnética.....	67
Figura 5.3	Esquemas do sistema helicoidal de varredura em videocassetes: (direita) revelação das pistas gravadas na fita.....	67
Figura 5.4	Processo de referenciamento indireto de cores utilizado em imagens referenciadas/indexadas e de intensidade.....	73
Figura 5.5	Representação em três matrizes de uma imagem RGB.....	74
Figura 5.6	Gráfico da luminância média dos quadros de um arquivo de teste com 300 quadros.....	76
Figura 5.7	Painel composto por quadros pertencentes a uma câmera (parte superior) e respectivo quadro-referência ampliado (parte inferior).....	83
Figura 5.8	Tela da janela de propriedades do sistema operacional do computador utilizado no processamento das imagens.....	84

Figura 5.9	Tela da janela do programa Process Explorer que exibe o comportamento do computador durante o processamento de um vídeo multiplexado.....	85
Figura 6.1	Painel com os quadros da ação dos réus junto à caixa de malotes.....	88

Lista de Tabelas

Tabela 4.1	Composição das palavras de 32 bits que representam os pixels.....	54
Tabela 4.2	Características dos sistemas de difusão dos sinais analógicos de televisão.....	58
Tabela 6.1	Arquivo vhs1.avi.....	89
Tabela 6.2	Arquivo anzo1_teste.avi.....	94
Tabela 6.3	Arquivo entrelaçado rápido.avi.....	99
Tabela 6.4	Arquivo Of_1220_08.avi.....	104

Lista de Abreviaturas e Siglas

CAIXA	Caixa Econômica Federal
CP	Código Penal Brasileiro
CCD	<i>Charge Coupled Device</i>
SETEC	Setor Técnico Científico do Departamento de Polícia Federal
SR/DPF/PE	Superintendência Regional da Polícia Federal em Pernambuco
VHS	<i>Video Home System</i> (Sistema de Vídeo Caseiro)
SP	<i>Standard Play</i> (Modo Padrão)
SLP	<i>Super Long Play</i> (Modo Super Longo)
<i>fps/qps</i>	<i>Frames Per Second</i> (Quadros Por Segundo)
CI	Circuito Integrado
S-Vídeo	<i>Separate Video</i> (Vídeo Separado Y-C)
Y	Componente de Luminância do Sinal de Vídeo
C	Componente de Crominância do Sinal de Vídeo
Cb	Componente de diferença de cor (B-Y) do Sinal de Vídeo
Cr	Componente de diferença de cor (R-Y) do Sinal de Vídeo
PC-TV	Placas de computador para reprodução dos sinais de vídeo e áudio de televisão em computadores pessoais (PC)
RCA	<i>Radio Corporation of America</i> , entidade americana que padronizou o conector que hoje conhecemos com esse nome para a interconexão de sinais de áudio e vídeo
AGC/CAG	<i>Automatic Gain Control</i> (Controle Automático de Ganho)
CVBS	<i>Composite Video Baseband Signal</i> (Vídeo Composto Sinal em Banda Básica)
A/D	<i>Analog-Digital</i> (Analogico Digital) refere-se ao conversor analógico/digital
PLL	<i>Phase Locked Loop</i> (Malha em fase) técnica para a geração de frequências precisas derivadas da precisão de uma referência de frequência
VOF	<i>Video Output Formatter</i> (Formatador do Vídeo de Saída) bloco do processamento do vídeo onde os bits da imagem digitalizada é formatado para adequar-se a determinado padrão
SDTV	<i>Standard Definition Television</i> (televisão de definição padrão) padrão convencional de sistemas de vídeo com resolução inferior a 750.000 pixels e com relação de aspecto 4:3, podendo apresentar-se também na versão <i>wide-screen</i> (tela larga) com relação de aspecto 16:9

HDTV	<i>High Definition Television</i> (televisão de alta definição) padrão de alta definição de sistemas de vídeo com resolução superior a 750.000 pixels e com relação de aspecto 16:9
ITU	<i>International Telecommunication Union</i> (União Internacional para Telecomunicações) agência da Organização das Nações Unidas voltada para a padronização de tecnologias de informação e comunicações
DSP	Digital Signal Processor (Processador de sinais digitais) circuito integrado dedicado (processador) dispendo de conjunto de instruções especialmente adaptado para realizar o processamento de sinais digitais
PCI	<i>Peripheral Component Interconnect</i>

Sumário

Resumo	iv
Abstract	v
Lista de Figuras	vi
Lista de Tabelas	ix
Lista de Abreviaturas e Siglas	x
Sumário	xii
Capítulo 1 – Introdução	14
1.1 Objetivos.....	17
1.2 Organização da dissertação.....	18
Capítulo 2 – Histórico	20
Capítulo 3 – Material sob análise	27
3.1 Entrelaçamento de imagens.....	28
3.2 O Comportamento da multiplexação.....	32
Capítulo 4 – Exames	34
4.1 Processamento analógico do vídeo.....	34
4.2 Processamento digital do vídeo.....	39
4.3 Processamento das informações contidas no intervalo do retraço vertical (VBI).....	40
4.4 Fluxo do processamento digital do vídeo.....	42
4.5 Processamento do sinal de luminância.....	46
4.6 Processamento do sinal de crominância.....	48
4.7 Redimensionamento Horizontal e Vertical.....	49
4.8 Bloco de formatação do vídeo de saída.....	52
4.9 A escolha dos parâmetros de amostragem dos sinais de vídeo composto.....	55
Capítulo 5 – Um novo algoritmo de demultiplexação	63
5.1 Captura.....	63
5.2 Pré-processamento.....	64
5.3 Processamento.....	70
5.3.1 Os critérios de classificação dos quadros em análise.....	77

5.3.2 Os critérios de criação de novos repositórios de câmeras.....	78
5.3.3 Comportamento do processamento.....	84
Capítulo 6 – Resultados Obtidos.....	86
6.1 Resultados Obtidos.....	86
6.2 Testes com o algoritmo.....	87
Capítulo 7 – Conclusões e Trabalhos Futuros.....	114
7.1 Sugestões para trabalhos futuros.....	114
Referências.....	116
Apêndice I – Algoritmo de Demultiplexação de Vídeo.....	119
Apêndice II – Tabela de Diferença de Luminância (criação de repositórios de quadros).....	126
Apêndice III – Artigo a ser submetido a evento internacional	133
Anexo I – DVD contendo:.....	137
Dissertação em formato pdf	
Código desenvolvido	
Amostras de vídeos multiplexados	
Artigo a ser submetido em formato pdf	
Codec Ligo’s Indeo 5.2	

Capítulo 1

Introdução

Em pleno século 21 vivemos em uma época de desconfiança, insegurança e ao mesmo tempo curiosidade. Com o advento da popularização e aprimoramento técnico dos processos de fabricação de câmeras, tornou-se viável a utilização desses equipamentos em diversas atividades que vão desde aquelas que permeiam o nosso cotidiano até a busca por outras formas de vida e a exploração de outros astros no espaço sideral.

A disseminação de sistemas de vigilância eletrônica cresce vertiginosamente em taxa ascendente que acompanha a sensação mundial de insegurança. As causas dessa insegurança, apesar de não se encontrarem no escopo desse trabalho, estão comprovadamente associadas à desigualdade, nas mais diversas acepções que a palavra pode assumir.

Outra forte razão para essa rápida popularização é a possibilidade de utilização desses sistemas como uma forma bastante eficiente de controle social. Esse controle está disseminado em todos os níveis dos agrupamentos humanos, indo desde o menor deles: nossas próprias famílias, quando desconfiamos dos cuidadores de pessoas incapazes (crianças, idosos, doentes, etc.), passando pelas câmeras dos condomínios, elevadores, áreas comuns. Elas também estão presentes nos nossos relacionamentos com grupos mais amplos, como em agências bancárias, *shoppings centers*, no controle do tráfego urbano, na luta contra o terrorismo e até em imagens de satélite da Terra e do espaço.

O binômio insegurança-vigilância tem causas diversas e que muitas vezes, por serem realimentadas por suas próprias conseqüências, dificulta sabermos quem causa o que, como é o caso, por exemplo, da produção em massa dos sistemas de vigilância. A grande demanda por esse tipo de produto gera o aprimoramento dos processos fabris, que por sua vez permitem uma redução dos custos de produção repassados ao consumidor gerando, dessa forma, mais demanda. Outro fator primordial que impulsiona a disseminação dessa verdadeira “febre das câmeras” é o custo do elemento humano nas atividades de vigilância. Quando vistos sob a ótica do empregador, salários cada vez mais altos, contribuições previdenciárias, ações trabalhistas,

licenças por motivo de saúde, férias, adicionais por insalubridade, etc. quando contabilizadas, mostram ser, senão o principal, um dos maiores ônus no balancete das empresas desse ramo. Além dos custos operacionais, somem-se outros inconvenientes associados ao elemento humano como os custos de treinamento, reciclagem, inexigibilidade de memória fotográfica, inexigibilidade de atenção plena durante todo o período, dentre muitos outros. Dessa feita, os sistemas de vigilância, quando bem planejados, dimensionados e gerenciados, apresentam inúmeros benefícios em certos aspectos, no entanto, jamais substituindo o elemento humano em outros.

Antes de tudo, o profissional de segurança que vai planejar um sistema de vigilância deve ter em mente qual o real intuito deste, o que se espera do sistema, qual seu objetivo? Trata-se de uma mera vigilância de uma área ampla, que visa reduzir o custo do elemento humano, restringindo-o a poucas pessoas, dentre elas os observadores de monitores que em caso de movimentação estranha acionam as autoridades competentes? Ou se deseja realizar uma filmagem em detalhes dos clientes para subsidiar a realização de exames de identificação biométrica facial de suspeitos? A resposta pode parecer simples uma vez que há uma tentação ao exagero, no entanto, essa decisão deverá levar em consideração uma análise complexa de outra variável que influencia sobremaneira os projetos: o custo. Este fator toma maiores proporções à medida que selecionamos componentes de boa qualidade e ao nos aproximamos da tecnologia de ponta, aumentando exponencialmente o fator custo com a sofisticação dos equipamentos envolvidos.

O que se tem notado, com base no material apresentado para análises forenses nos últimos cinco anos, durante os quais o autor tem trabalhado com análises de imagens dessa natureza, é que, via de regra, os sistemas normalmente atendem à primeira proposta, ou seja, objetivam a mera vigilância de áreas amplas tendo como objetivo principal um controle remoto da situação global das agências. É regra a utilização de sistemas de baixo custo, apresentando, dentre outras limitações algumas principais:

1. Utilização de câmeras com pouca resolução, grande distorção, sobretudo esférica, associada à péssima qualidade do seu conjunto ótico que é normalmente composto de uma lente do tipo grande angular¹;
2. Emprego de técnicas de multiplexação tanto no tempo² quanto no espaço³ (Figura 1.1) que reduzem o dinamismo e a resolução do vídeo com vistas a economizar espaço de armazenamento do mesmo;
3. Utilização de sistemas de armazenamento analógicos ou mesmo digitais com capacidade de gravação de detalhes reduzida em razão das mídias utilizadas, para poupar espaço de armazenamento e outras limitações;
4. Armazenagem de imagens estáticas seqüenciais temporizadas, que são, na verdade, uma seqüência intermitente de fotografias em lugar de um vídeo dinâmico e contínuo. Com o emprego dessa técnica, além de se sacrificar o elemento continuidade do vídeo, via de regra, os intervalos de atualização das imagens são insuficientes até mesmo para se acompanhar o deslocamento de suspeitos, chegando ao ponto, como já se constatou inúmeras vezes, de que se as ações praticadas tiverem duração inferior ao período de atualização das imagens, essas podem simplesmente deixar de ser capturadas;
5. Reutilização em excesso das fitas analógicas que registram as imagens do sistema, o que é uma prática freqüente, degradando ainda mais a qualidade das imagens ali gravadas.

¹ As lentes do tipo grande angular têm como principal característica sacrificarem a linearidade das imagens em detrimento de um ângulo de visão mais amplo.

² Multiplexação de imagens analógicas no tempo é, sumariamente, uma técnica que cria uma seqüência de imagens (via de regra campos) com sua resolução nativa, provenientes das diversas câmeras que compõem o sistema, atualizando, a cada instante de tempo, apenas imagens provenientes de uma câmera de cada vez.

³ Multiplexação de imagens no espaço é sumariamente uma técnica que cria um reticulado composto das imagens provenientes de todas as câmeras e as exibe todas de uma só vez com suas resoluções nativas reduzidas para que todas caibam em uma única tela.



Figura 1.1: Exemplo de um quadro proveniente de um sistema de vídeo multiplexado no tempo

É normalmente com base em imagens que apresentam os vícios expostos, seja em razão do planejamento do sistema de vigilância, seja por características dos seus componentes de baixa qualidade que a perícia é rotineiramente desafiada. Tendo por base imagens de baixa resolução, geradas a partir de um sistema que objetiva a mera vigilância de áreas amplas, são solicitadas identificações biométricas faciais de suspeitos, ou ainda, com base em vídeos intermitentes com períodos de atualização das imagens excessivamente longos, verificar trajetórias ou identificar suspeitos. Em muitos desses casos, infelizmente, os expertos vêem-se simplesmente impedidos de proferir qualquer parecer, tendo em vista a inadequabilidade do material encaminhado.

1.1 Objetivos

A presente dissertação tem por objetivo amplo abordar o tema multiplexação de vídeos de segurança sob um ponto de vista bastante prático, sempre com foco forense. Movidos pela necessidade de realizar a demultiplexação no tempo do vídeo de um caso real, os trabalhos focaram o desenvolvimento de um algoritmo que fosse capaz de classificar os quadros multiplexados em suas respectivas câmeras originárias. Apesar de na maior parte dos capítulos da

presente dissertação realizarmos inicialmente uma abordagem do assunto central de uma forma geral, adotaremos uma metodologia lógica dedutiva, ou seja, partindo desses conceitos gerais convergiremos para o caso real que é o objetivo *strictu* desse trabalho.

No decorrer da presente exposição realizaremos um estudo descritivo das técnicas mais comuns de multiplexação e demultiplexação de vídeo provenientes de sistemas de segurança, em especial bancários. Descreveremos o caso real que foi a inspiração dos trabalhos aqui narrados e no qual a técnica desenvolvida foi utilizada para solucioná-lo. São descritas também as características das imagens apresentadas para exame e as técnicas auxiliares utilizadas no pré-processamento das imagens questionadas. Trataremos das primeiras abordagens ao caso chegando às técnicas que obtiveram maior êxito na correta classificação das imagens. Com o intuito de preservar anônimas as identidades de funcionários, clientes e dos próprios suspeitos, que até o momento da produção desse trabalho ainda se encontram envolvidos em processo probatório, foram aplicados efeitos de distorção sobre seus rostos, caracteres de hora e data dos quadros e outros detalhes ou informações que possam identificá-los.

1.2 Organização da dissertação

A dissertação está dividida em sete capítulos e dois apêndices a saber:

O primeiro deles é a presente introdução onde são expostos genericamente alguns conceitos sobre sistemas de vigilância e seu relacionamento com aspectos da vida moderna e tendências comportamentais. São expostos também sinteticamente os objetivos e a organização do trabalho.

No Capítulo 2 narra-se o histórico do caso real que deu origem ao presente trabalho. Trata-se de uma solicitação de exame pericial em imagens que, a princípio, não podiam ser analisadas da forma como foram apresentadas, tampouco estavam disponíveis os equipamentos específicos para esse fim, demandando o desenvolvimento das técnicas que serão expostas.

O terceiro capítulo descreve do ponto de vista técnico, perceptual e do seu conteúdo as imagens contidas na mídia apresentada. Discorre-se ainda acerca dos equipamentos utilizados e procedimentos preliminares que se fizeram necessários no que se chamou de pré-processamento do vídeo em análise.

No Capítulo 4 é descrito o algoritmo propriamente dito de demultiplexação de vídeo, algumas técnicas que foram experimentadas sem sucesso e a razão destes. Abordamos ainda conceitos envolvidos e exemplos de campos, quadros, critérios de classificação, aprimoramentos das técnicas.

O Capítulo 5 descreve o algoritmo em si, as técnicas de abordagem ao problema, inclusive descrevendo as causas e conseqüências da evolução do algoritmo no decorrer de seu desenvolvimento. Por fim, descreve-se seu estado atual.

No Capítulo 6 são mostrados os resultados obtidos com a aplicação do algoritmo em alguns casos de vídeo semelhantes. Os resultados são apresentados mostrando-se características originais do vídeo, fases do pré-processamento, processamento, eficiência do algoritmo e causas de falhas de classificação. Neste capítulo também são feitas algumas sugestões de extensão do presente trabalho e idéias para trabalhos afins.

No Capítulo 7 realiza-se uma crítica geral do trabalho, o desfecho do ponto de vista pericial do caso com a conclusão do respectivo laudo. Aborda ainda a possibilidade de utilização do algoritmo em outros casos reais e as vantagens tanto técnicas quanto financeiras para as unidades da Criminalística tanto nos Institutos de Polícia Científica dos estados quanto nas unidades centrais (Instituto Nacional de Criminalística) e descentralizadas (Setores Técnico-Científicos das Superintendências Regionais da Polícia Federal) da Perícia Federal.

Com o intuito de melhor estruturar o presente trabalho, optou-se por apresentar sob a forma de dois apêndices informações relevantes sobre o trabalho.

O Apêndice I apresenta o código da última versão do algoritmo desenvolvido.

O Apêndice II traz uma tabela de diferenças de luminância entre quadros demonstrando o processo de classificação de quadros e criação dos repositórios de quadros.

O Apêndice III apresenta artigo técnico a ser submetido a evento internacional.

No anexo I há um DVD contendo esta dissertação em formato pdf, o código desenvolvido, amostras de vídeos multiplexados, o artigo a ser submetido a evento internacional e o Codec Ligo's Indeo 5.2.

Capítulo 2

Histórico

Expostos alguns pontos que tangem de maneira genérica o tema da presente dissertação, passamos ao histórico do caso real que levou ao estudo das técnicas de multiplexação e demultiplexação de vídeo de sistemas de segurança.

Atendendo a uma ocorrência policial em fevereiro do ano de 2005, uma equipe de peritos criminais da Polícia Federal, sob a coordenação do autor, foi acionada para averiguar uma denúncia de tentativa de furto em uma agência bancária localizada na região da zona da mata no estado de Pernambuco. Os peritos se deslocaram até a agência e realizaram os procedimentos de praxe em locais de crime que visam a contextualização do delito, a coleta de possíveis provas e vestígios além do levantamento papiloscópico de impressões digitais. Durante os exames, verificou-se que a agência dispunha de um sistema de vigilância eletrônica composto por algumas câmeras, um multiplexador, um equipamento de videocassete padrão VHS e um monitor de vídeo.

Balizados pelos testemunhos de alguns funcionários e clientes, e com base numa rápida visualização do vídeo demultiplexado de uma das câmeras ainda no próprio sistema da agência, os peritos puderam observar nas imagens a confirmação da denúncia do crime. Naquele mesmo dia, o autor, de improviso, oportunamente capturou um quadro das imagens (Figura 2.1) fotografando a tela do monitor do sistema da agência que exibia o vídeo contido na fita da movimentação daquele dia. Nessa fotografia é possível observar o exato momento no qual é perpetrada a fraude, quando o meliante com a mão esquerda segura o que seria aparentemente apenas um envelope branco e com a mão direita bota em prática o furto. Simultaneamente, para ocultar o crime, sua cúmplice apóia uma pasta, com o intuito de bloquear o campo de visão de outra câmera, mas, aparentemente sem notar, os criminosos deixam desimpedida a filmagem observada.

Como resultado dos exames preliminares no local do crime, foi gerado um primeiro documento: a Informação Técnica nº 10/2005 [1], no qual os peritos narram a vistoria ao

presidente do inquérito policial, relatando a existência do sistema de vigilância eletrônica e de uma fita que sabidamente continha imagens da atuação dos meliantes, prova incontestável do crime.



Figura 2.1: Fotografia do monitor do sistema de segurança usado no dia da prisão dos suspeitos para reproduzir a fita com a gravação da ação dos meliantes

O crime que estava sendo praticado era o furto de malotes empresariais que se encontravam depositados no interior de uma caixa de coleta para essa finalidade. Para tanto, os meliantes se utilizavam de petrechos que foram apelidados pelo autor de “conjunto de pesca” (Fig.2.2) composto por um envelope branco com fragmento de azulejo no seu interior que servia como carretel para cerca de 1,5 metro de linha de nylon monofilar com 0,5mm de espessura, na extremidade da qual encontravam-se amarrados uma chumbada para pesca e uma garatéia⁴. Não bastasse o anzol de três pontas, foram apreendidos em poder de um dos meliantes tubos de cola instantânea à base de cianoacrilato que eram usados na garatéia para facilitar que a mesma “fisgasse” os malotes ou que se colasse aos mesmos.

A caixa de coleta de malotes (Fig.2.3) era confeccionada em compensado de madeira laminado com fórmica, apresenta uma porta na face traseira (Fig.2.4) que fica normalmente

⁴ Garatéia é um tipo de anzol composto na verdade por três anzóis soldados por suas hastes e apresentando apenas um ilhós.

trancada, sendo acessada apenas por funcionários por ocasião do seu processamento e uma abertura na face superior (Fig.2.5) para a introdução dos malotes empresariais (Fig.2.6).



Figura 2.2: Petrechos que foram encontrados com os meliantes e que estavam sendo utilizados para a perpetração do furto

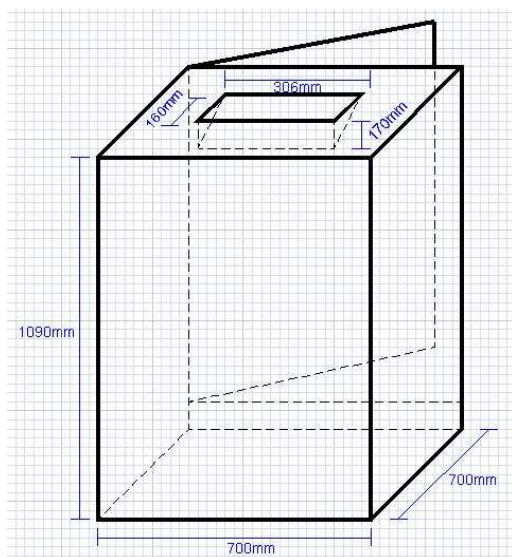


Figura 2.3: Características da caixa de malotes (vista frontal)



Figura 2.4: Detalhe da porta traseira da caixa de malotes (vista posterior)



Figura 2.5: Abertura na face superior da caixa de malotes usada normalmente para a introdução dos malotes



Figura 2.6: malote empresariam em lona sintética azul

O crime de furto está previsto no Código Penal Brasileiro em seu artigo 155, sendo a pena de 1 a 4 anos de reclusão. No entanto, esta pode aumentar para 2 a 8 anos de reclusão caso o furto seja considerado qualificado, devendo para tanto atender a alguns requisitos dentre eles dois nos quais o caso em tela poderia ser enquadrado por ter sido cometido:

“[...]

II – com abuso de confiança, ou mediante fraude, escalada ou **destreza**⁵;

[...]

IV – mediante **concurso de duas pessoas ou mais pessoas.**” (grifo nosso).

Dessa forma, o parecer do perito seria uma peça de suma importância para subsidiar a decisão judicial ao pronunciar-se no que se refere à qualificação do crime, uma vez que as imagens da ação conjunta dos meliantes corroboradas pela visualização das técnicas utilizadas qualificariam duplamente o crime.

Outro ponto que precisava ser esclarecido era a verificação da viabilidade da consumação do crime com os equipamentos apreendidos. A análise justificava-se pois era ponto pacífico que a defesa dos réus, em virtude dos estranhos petrechos apreendidos, alegaria que com os mesmos o crime nunca poderia ser consumado por absoluta ineficácia do meio empregado. Dessa forma, o

⁵ “[...] 2. **Agilidade de mãos, e de todos os movimentos.** 3. **Habilidade,** aptidão. 4. Sagacidade. **Astúcia.** (AURÉLIO BUARQUE DE HOLANDA FERREIRA, 2004, grifo nosso).

que se esperava é que a defesa invocasse o instituto do crime impossível. Definido na parte geral do Código Penal Brasileiro em seu artigo 17, é definido nos seguintes termos:

“Art. 17. Não se pune a tentativa quando, por ineficácia absoluta do meio ou por absoluta impropriedade do objeto, é impossível consumar-se o crime”

Caso o juízo se convencesse da absoluta ineficácia dos objetos apreendidos para a consumação do crime, ficaria afastada a tipicidade da conduta, deixando esta última de configurar crime. Com o intuito de averiguar a eficácia da técnica empregada pelos réus, foram tomadas duas providências:

A primeira foi a realização, pelos peritos, de um exame de reconstituição do que seria a perpetração da fraude. Para tanto, valeram-se da utilização dos petrechos encontrados na posse dos meliantes quando da prisão em flagrante dos mesmos na agência bancária. Com isso, desejava-se verificar a eficácia da técnica empregada, que caso se confirmasse, inviabilizaria a possibilidade de alegação de crime impossível por parte da defesa. A reconstituição foi realizada em meados de 2006, tendo como resultado um segundo parecer técnico relacionado ao caso, o Laudo Pericial nº 980/2006 [2]. Entre outras providências a peça pericial versa sobre a caracterização de todo o material, descreve o ensaio *in loco* para se averiguar quão eficaz eram os petrechos apelidados de “conjunto de pesca”. Para a surpresa de todos, a técnica da “pescaria” de malotes era extremamente eficiente, de forma que sem qualquer treinamento os peritos foram capazes de “fisgar” e içar malotes num tempo médio de cerca de apenas 20 segundos.

A segunda providência foi a análise da fita contendo as imagens da movimentação do dia da tentativa de furto, tema central dessa dissertação, com o intuito de ilustrar o inquérito policial com quadros que demonstrassem bem a atuação conjunta do casal e que não deixassem dúvidas com relação à existência das circunstâncias qualificadoras. Conforme já descrito, os peritos haviam fotografado a tela do monitor no exato instante em que o casal realizava a “pescaria” dos malotes, no entanto, quando a fita foi encaminhada para exames, muitos meses depois do dia do ocorrido, esta foi apresentada em sua forma nativa do sistema, ou seja, multiplexada no tempo. Da forma como foi apresentada (multiplexada), seria impossível localizar o exato instante em que o casal põe em prática seu plano pois como descreveremos a seguir a fita apresentava várias horas de vídeo. Além disso, havia na fita uma grande quantidade de informações irrelevantes, não havia caracteres de data e hora que pudessem delimitar o trecho a ser analisado e havia a

dificuldade ainda de realizar um exame puramente perceptual das imagens, da forma como foram apresentadas.

Dessa feita, na esperança de realizar a demultiplexação das imagens através do equipamento específico para essa finalidade, foi solicitado à CAIXA que disponibilizasse os equipamentos necessários para a demultiplexação do vídeo para que o mesmo pudesse ser digitalizado e em seguida fossem localizadas as melhores imagens que ilustrariam o delito. Após a demultiplexação, a análise se limitaria ao exame das imagens de uma das câmeras, pois com o auxílio das fotografias tiradas no dia do delito, seria possível identificar a câmera e o trecho de interesse. Assim, o intervalo de vídeo a ser analisado ficaria reduzido e o restante do trabalho poderia ser feito de forma perceptual utilizando, por exemplo, uma reprodução do vídeo com velocidade acelerada.

Surpreendente foi a resposta da CAIXA, ao informar que não dispunha mais de demultiplexadores compatíveis com aquele padrão de gravação da fita em análise. Tal afirmação fundamentou-se no fato de que em razão da necessidade do pedido de encaminhamento da fita e dos equipamentos ter que necessariamente percorrer um trâmite formal e burocrático, já haviam passados vários meses entre a ocorrência e a resposta da CAIXA. Além disso, não há uma padronização na multiplexação/demultiplexação de vídeo, ou seja, cada fabricante tem um padrão proprietário. Outro aspecto *sui generis* da CAIXA que deve ser levado em consideração é que a aquisição de equipamentos pela mesma passa necessariamente por um processo licitatório, de forma que, a cada compra realizada, novos padrões vieram sendo introduzidos gerando uma miríade de fabricantes e padrões de multiplexadores/demultiplexadores incompatíveis entre si, diversidade essa agravada pelo fato de que modelos mais antigos são rapidamente descontinuados em virtude de interesses econômicos das empresas fabricantes e da rápida obsolescência tecnológica dos equipamentos e componentes.

Frustrada a intenção inicial de realizar a demultiplexação através das vias normais, ou seja, utilizando o hardware proprietário e tendo em vista a inviabilidade dos exames puramente perceptuais e a importância da prova pericial como subsídio técnico ao pretor, o autor teve de buscar alternativas para realizar a demultiplexação do vídeo gravado na fita apresentada. A solução vislumbrada por fim foi realizar a digitalização de todo o conteúdo da fita apresentada e

através de um algoritmo de processamento de imagens digitais demultiplexar o vídeo, processo que foi tentado de algumas formas que serão abordadas nos capítulos 3 e 5.

Por fim, como resultado do método proposto, foi possível realizar a demultiplexação das imagens utilizando um algoritmo totalmente desenvolvido com esse intuito e com isso realizou-se a identificação da câmera e do trecho no qual aparecem os criminosos em ação. Um último documento técnico, o Laudo Pericial 1236/2006 [3] relata de forma sintética o exame e encaminha os melhores quadros do vídeo nos quais verifica-se a incontestável viabilidade do crime, ação conjunta dos criminosos e as circunstâncias qualificadoras sob as quais o crime foi perpetrado. Essas três peças, a Informação Técnica 010/05, o Laudo 980/06 e o Laudo 1236/06 formam o conjunto probatório técnico para esse caso, encerrando, desse ponto de vista, a atuação da Perícia nesse caso.

Cumprir informar que os laudos periciais têm a finalidade de esclarecer e subsidiar, como temos repetido no decorrer deste documento, as decisões dos magistrados, não cabendo, dessa forma, grandes detalhes técnicos nessas peças jurídicas. Informações como as teorias ou conceitos envolvidos para a obtenção dos resultados, não são, a princípio, matéria a ser abordada nos laudos, a menos que as técnicas venham a ser questionadas por auxiliares técnicos das partes ou mesmo por outros peritos. Normalmente é feita uma explanação superficial de como foram realizados os exames, uma descrição do material encaminhado e quais foram os resultados obtidos.

O Perito é o assessor técnico em assuntos não jurídicos do Poder Judiciário. É invocado sempre que informações das mais diversas áreas do conhecimento humano são necessárias para auxiliar no processo de tomada de decisões judiciais. Ele está presente seja avaliando bens, determinando a *causa mortis*, verificando a autenticidade de documentos e assinaturas, enfim em uma gama enorme de áreas de atuação. É nesse cenário que atua a Perícia Criminal da Polícia Federal que vem se especializando em áreas e subáreas do conhecimento, buscando sempre poder colocá-los em prática no dia-a-dia do sistema Judiciário em busca de uma sociedade mais humana e justa.

Capítulo 3

Material sob análise

Logo após os exames de levantamento de vestígios e caracterização do local de crime, os peritos que estiveram na agência, dentre eles o autor, relataram o que observaram dando ênfase à existência de um sistema de vigilância eletrônica que havia gravado a ação dos meliantes em uma fita que sabidamente continha imagens extremamente elucidativas quanto ao *modus operandi* dos criminosos. Dessa feita, na peça pericial exordial do caso, a já citada Informação Técnica 010/05 deu início ao processo de solicitação da fita. Em razão dos trâmites administrativos e judiciais a fita só foi enviada quase um ano após a solicitação da perícia.

A mídia apresentada tratava-se de uma fita magnética de vídeo padrão VHS (*Video Home System* ou Sistema de Vídeo Caseiro no vernáculo) e estava acondicionada em estojo de papelão próprio para esse fim. A fita era da marca EMTEC, modelo T160DC com capacidade de gravação de 160 minutos em modo SP (*Standard Play* ou Modo Padrão no vernáculo) e apresentava uma etiqueta adesiva identificando-a fixada em sua lombada com as inscrições:

“MESES: 02,04,06,08,10,12 02 0”;

A partir dessas inscrições deduziu-se que havia um procedimento de arquivamento das imagens da agência por um período de um mês, depois do qual a fita era reutilizada. Daí os peritos já poderiam esperar imagens deterioradas pela reutilização excessiva da fita magnética.

Durante uma análise preliminar, verificou-se que a fita apresentava oito horas (8h) de gravação analógica de vídeo em modo SLP (*Super Long Play* ou Modo Super Longo no vernáculo), sem áudio, sem caracteres de data ou horário tanto na área visível do vídeo quanto nas linhas do retraço, contendo imagens estáticas seqüenciadas, monocromáticas, entrelaçadas, de qualidade razoável, proveniente de um sistema multiplexado no tempo. A multiplexação era irregular no que dizia respeito à frequência de comutação das câmeras, à duração do período de gravação de cada uma delas e à ordem das mesmas. De imediato foi difícil identificar o número de câmeras pois no trecho inicial do vídeo o sistema parece se ajustar, não exibindo o vídeo de

algumas câmeras, no entanto, após analisar trechos localizados mais próximos do meio da fita constatou-se de modo perceptual a presença de 8 (oito) câmeras.

Levando-se em consideração alguma experiência do autor com equipamentos de multiplexação de vídeo, as irregularidades na multiplexação do vídeo presentes no trecho inicial das gravações poderiam estar de alguma forma associadas a diversas causas que vão desde um simples mau-funcionamento no equipamento multiplexador à existência de funcionalidades ou características avançadas do mesmo, como por exemplo: o fato da captura de quadros de uma determinada câmera estar condicionada à detecção de movimento naquela área ou indo além, a possibilidade de haver um aumento da frequência de atualização dos quadros, em detrimento de outras câmeras, de uma determinada área se lá for constatada uma grande movimentação. O fato é que seja qual for o motivo, que provavelmente nunca saberemos, o sistema gerava uma seqüência quase desordenada de imagens provenientes de oito câmeras.

3.1 Entrelaçamento de imagens

De imediato, surgiram algumas soluções preliminares como a idéia de uma simples separação das imagens partindo do inocente pressuposto de que as mesmas seguiam alguma seqüência, no entanto, como foi descrito acima, essa abordagem foi logo abandonada quando se constatou o comportamento irregular na multiplexação. Outras soluções começaram a ser testadas mas ao se iniciar qualquer processamento verificou-se que o fato das imagens serem entrelaçadas criava uma situação inconveniente: a existência de quadros híbridos formados por campos provenientes de câmeras distintas como podemos observar na Fig. 3.1. Nessa figura podemos observar na imagem superior um exemplo típico de um quadro híbrido Fig. 3.1 a) composto pelos campos Fig. 3.1 b) e c). Note-se que ao serem desentrelaçados os campos têm sua resolução vertical reduzida à metade, o que pode ser observado comparando-se a altura das imagens dos campos que é a metade daquela do quadro híbrido. No entanto, depois de alguma reflexão também fundamentada na análise tanto de quadros simples (assim chamados aqueles compostos por ambos os campos da mesma câmera), quanto daqueles híbridos, concluiu-se que o vídeo deveria ser desentrelaçado sempre antes de qualquer processamento pois em ambos os casos (quadros compostos por imagens da mesma câmera ou de câmeras distintas) a informação presente em cada um dos campos é única e diferente, seja por se tratar de registros de imagens

estáticas do mesmo local mas em instantes diferentes, seja por se tratarem de imagens estáticas de locais diferentes. Mesmo que os personagens na cena possam permanecer aparentemente estáticos, a cada campo capturado a câmera pode mudar parâmetros de exposição, foco automático ou ajuste de brilho e disponibilizar uma imagem mais nítida de um rosto ou objeto como pode-se observar na Figura 3.2 que demonstra um quadro de um vídeo no instante de uma transição. A possibilidade de obtenção de ao menos uma imagem que ofereça uma maior nitidez e/ou exposição mais adequada e/ou melhor foco e/ou captura de um ângulo mais favorável, justificaria o esforço computacional empregado. Dessa forma, a possibilidade de obtenção de detalhes suficientes para viabilizar a realização de um exame de identificação biométrica facial com resultado positivo para a identificação de um suspeito ou a identificação da placa de um veículo, por exemplo, em ao menos um campo já é argumento por demasiado forte para que não se descarte nenhum dos campos, por mais redundantes que os mesmos possam aparentemente parecer.



Figura 3.1: Exemplo de um quadro híbrido (a) composto por dois campos (b e c) provenientes de câmeras distintas



Figura 3.2: Exemplo de zoom em uma região de um quadro entrelaçado, percebe-se que os campos apresentam imagens ligeiramente diferentes, da tecla F12 de um teclado sendo pressionada. Figura retirada da referência [5]

Dessa forma, a técnica de entrelaçamento das imagens no vídeo apresentado, nesse caso, mostra-se mais como um problema do que uma vantagem que a mesma traz que é causar a impressão de suavidade na transição dos quadros, razão da sua utilização nos sistemas de vídeo. No caso específico de exames periciais, busca-se no vídeo, na verdade, imagens estáticas nítidas, sem perda de resolução, que possam ser usadas para realizar algum tipo de identificação ou esclarecer dinâmicas. Assim sendo, as vantagens de suavidade de movimento e cancelamento do efeito de cintilação não oferecem qualquer ajuda nos exames.

Discorrendo um pouco sobre a técnica de entrelaçamento de imagens, encontramos alguns conceitos bastante interessantes e elucidativos em [4] no seu Capítulo 3 – Varredura e Sincronização. Apesar de ser antigo, o livro é tido como uma referência até os dias de hoje no que diz respeito ao sistema de televisão analógica. Grob se refere ao entrelaçamento como um procedimento, uma forma de ludibriar a vista humana ao dobrar a frequência de varredura vertical. Na verdade, a ilusão de movimento pode ser criada apresentando imagens estáticas a uma taxa de 16 *fps* (*frames per second* ou no vernáculo quadros por segundo), tempo suficientemente curto para que os olhos humanos retenham as imagens dando a impressão de continuidade. Apesar dessa sensação de movimento, a taxa de 16 *fps* ou mesmo aquela utilizada

pela indústria cinematográfica que é de 24 *fps*, mostram-se insuficientes para evitar a sensação de cintilação, que é percepção desagradável da mudança abrupta de brilho de uma imagem para a seguinte. Para resolver esse problema, a técnica utilizada pelo cinema é exibir o mesmo quadro duas vezes de forma que a taxa de exibição de cenas por segundo seja 48 *fps*. No caso da televisão, com vistas a evitar o batimento com a frequência da rede elétrica que tem frequência bastante próxima e que causaria uma interferência que rolaria pela imagem, o sistema de televisão analógico utiliza uma taxa de atualização de quadros de 30 *fps*. Como cada quadro é composto por dois campos, a frequência de campos é o dobro daquela de quadros, ou seja, de 60 *fps*, dessa forma, possíveis interferências da rede elétricas ficarão estáticas na imagem. É interessante notar que o valor da frequência de varredura para que se cancele o efeito de cintilação depende de uma série de fatores mas principalmente do brilho das imagens exibidas e da luminosidade do ambiente onde está sendo exibido o vídeo. Por essa razão, as salas de cinema apresentam baixa luminosidade interna, obtendo com taxas mais baixas de exibição de quadros a mesma sensação agradável de movimento e variação sutil de brilho.



Figura 3.3: Uma das imagens que compõem
O cavalo galopante de Muybridge

Apesar da abordagem de Grob, outra bastante interessante foi encontrada em [5] onde o entrelaçamento é visto como uma forma bastante inteligente de compressão analógica para se economizar largura de banda sem o sacrifício da suavidade dos movimentos e da qualidade das imagens.

Some-se aos fatores citados o fato de que as imagens multiplexadas do caso analisado serem na verdade uma seqüência temporizada de imagens estáticas e não um vídeo dinâmico e fluido como os programas de televisão, para os quais foi desenvolvida a técnica do

entrelaçamento de vídeo. Dessa forma, a vantagem da continuidade já havia sido sacrificada pelo sistema.

Somando-se as razões analisadas verifica-se que a técnica de entrelaçamento combina imagens atualizadas a uma taxa que na verdade é o dobro daquela da exibição de quadros completos para o espectador. Essa combinação entrelaça informações distintas em um quadro único, agradável para ser assistido, mas que para a análise de seus campos componentes exige seu desentrelaçamento.

3.2 O comportamento da multiplexação

Com fulcro nos argumentos apresentados verifica-se a necessidade de se realizar o desentrelaçamento das imagens antes que qualquer processamento seja realizado. Passamos então a analisar o comportamento da multiplexação do vídeo em questão. O esquema de comutação observado, já superficialmente descrito, não apresentava uma seqüência perfeitamente ordenada. Muito longe disso, durante os estudos preliminares no material, chegou-se ao que seria a lógica de formação da seqüência de imagens que pode ser observada no esquema elucidativo da Figura 3.4 descrita pelo autor como “comutação cíclica desordenada”. Apesar de não seguirem uma seqüência rígida, as câmeras também não eram comutadas em uma ordem completamente aleatória, percebia-se que havia um certo grau de liberdade mas que esse se dava em torno de uma seqüência fundamental que era buscada pela multiplexador. O que se chama aqui de seqüência fundamental é, segundo análises nas imagens, na verdade uma tendência do processo de multiplexação em buscar amostrar as imagens de cada uma das câmeras de forma equilibrada, guardando assim uma característica cíclica latente.

Apesar de guardar uma tendência cíclica, a comutação apresentava um grau de aleatoriedade suficiente para inviabilizar uma demultiplexação das imagens com base apenas na seqüência fundamental, frustrando as chances de se chegar a um resultado final satisfatório com essa abordagem.

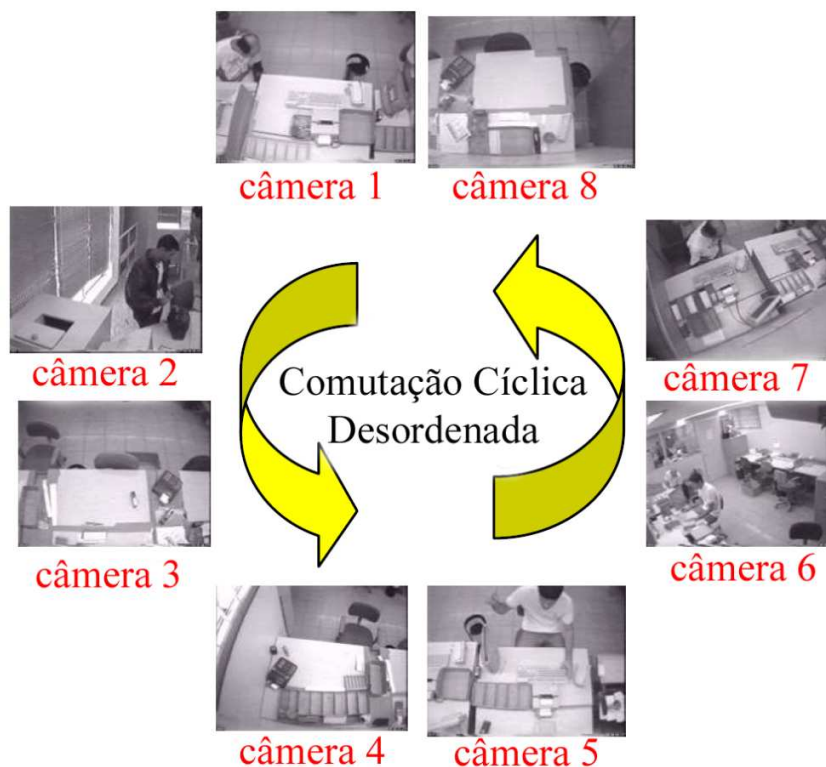


Figura 3.4: Esquema de comutação das câmeras no vídeo analisado

Depois de visualizar o vídeo do ocorrido inúmeras vezes em vários de seus trechos é que se chegou ao esquema de comutação acima, concluindo-se que o sistema era composto pela multiplexação de oito (8) câmeras. Sem perder o foco das investigações periciais que era a identificação do casal no intervalo da perpetração do crime que acontece junto à caixa de malotes empresariais e tendo como subsídio a fotografia do monitor de vídeo do sistema de vigilância eletrônica tirada no dia da prisão dos meliantes, Figura 2.1, as buscas por imagens restringiram-se àquelas provenientes da câmera 1.

Capítulo 4

Exames

Depois de contextualizar a situação e descrever o material encaminhado e as imagens nele gravadas, passamos a discorrer sobre os exames realizados. Como já foi dito, o primeiro passo da análise foi a digitalização das imagens constantes na fita. Para tanto, valeram-se os peritos de uma placa de captura de vídeo. Com o intuito de entender como se dá o processo de digitalização de vídeo e principalmente, saber o que esperar do arquivo gerado, passamos a analisar a tecnologia envolvida e as técnicas de que se utilizam esses equipamentos para transformar o vídeo analógico em matrizes de bytes.

Em busca na bibliografia citada, e tendo por base também pesquisas realizadas na Internet, procurou-se por informações sobre o funcionamento detalhado das placas de captura de vídeo. Apesar de encontrar-se vasta gama de páginas sobre esse tipo de equipamento, quando buscadas informações com palavras-chave intuitivas, a maioria esmagadora dos sítios apresenta conteúdo relacionado à comercialização de produtos e não à elucidação do seu modo de funcionamento e discriminação de detalhes de seus circuitos. Essas informações só começaram a ser mais fartamente encontrados associadas à expressão chave em inglês: “*frame grabber*” ou no vernáculo: capturadora de quadros. Para entendermos melhor o que esse equipamento é capaz de fazer, e como o faz, passamos agora a um resumo de seu funcionamento, fruto de uma pesquisa nos sítios Internet de vários fabricantes de placas e em especial em folhas de dados de vários circuitos integrados dedicados a essa finalidade.

4.1 Processamento Analógico do Vídeo

Iniciamos por ilustrar quais são as etapas básicas e como são realizadas as fases de decodificação e quantização dos sinais de vídeo. Para tanto, nos valem dos diagramas em blocos (Figuras 4.1 e 4.2) de um dos *chipsets* (conjuntos de circuitos integrados) dedicados para essa finalidade mais vendidos no mundo durante um período considerável, o Bt848 da divisão

Brooktree pertencente à empresa Rockwell Semiconductors Systems. A proposta desse circuito integrado (CI) é apresentar uma solução de baixo custo para a digitalização de vídeo multi-sistema (NTSC/SECAM/PAL), composta por um único CI, e com interface de saída para barramentos PCI (*Peripheral Component Interconnect*).

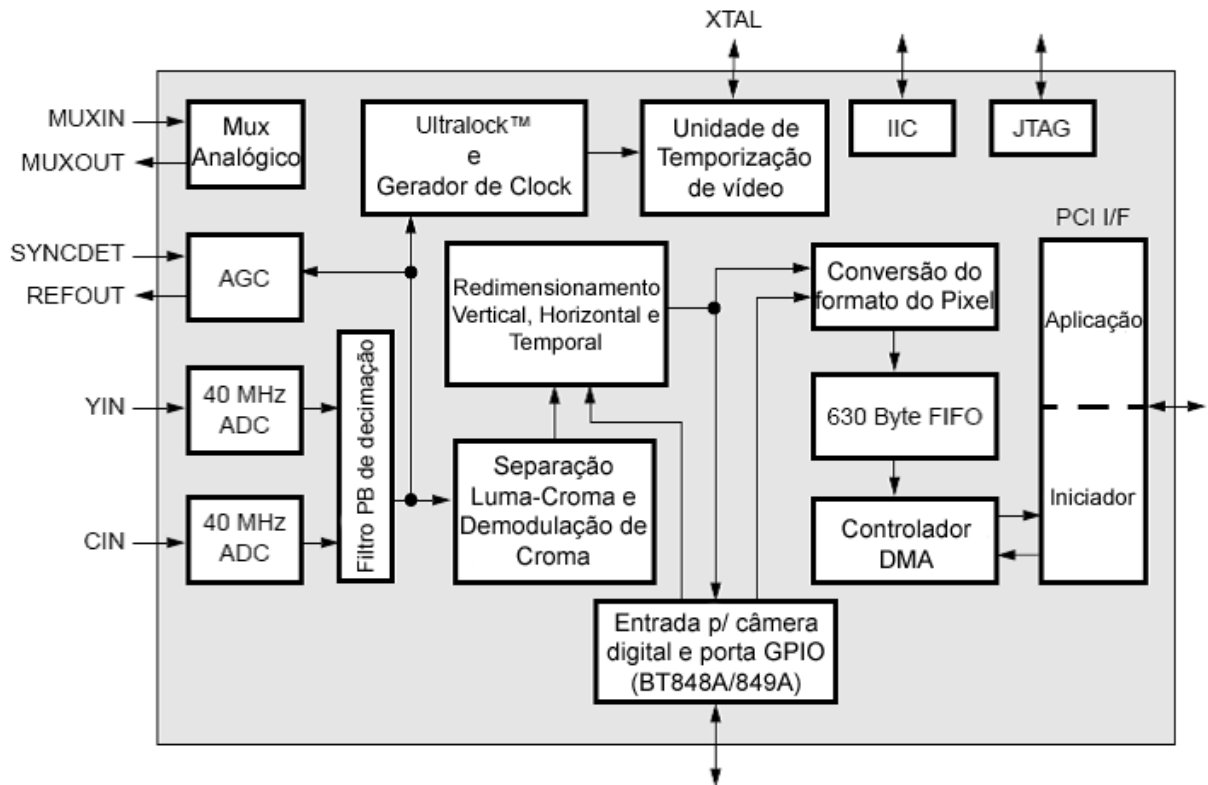


Figura 4.1: Diagrama em blocos do circuito integrado Bt848 da empresa Brooktree [22]

Em dispositivos dessa natureza, os sinais analógicos de vídeo entram por um multiplexador ao qual normalmente são ligados sintonizadores internos de TV ou recebem vídeo externo pelas entradas de vídeo composto ou S-Vídeo. Além do vídeo, normalmente apresentam entradas para áudio externo e também interno associado ao vídeo e também da sintonia de estações FM comerciais. Esse conjunto oferece assim uma solução multimídia completa que é a proposta do fabricante. Foquemos então na digitalização de vídeo externo proveniente das opções mais comuns de interconexão de sinais de vídeo analógico (vídeo composto e S-Vídeo).

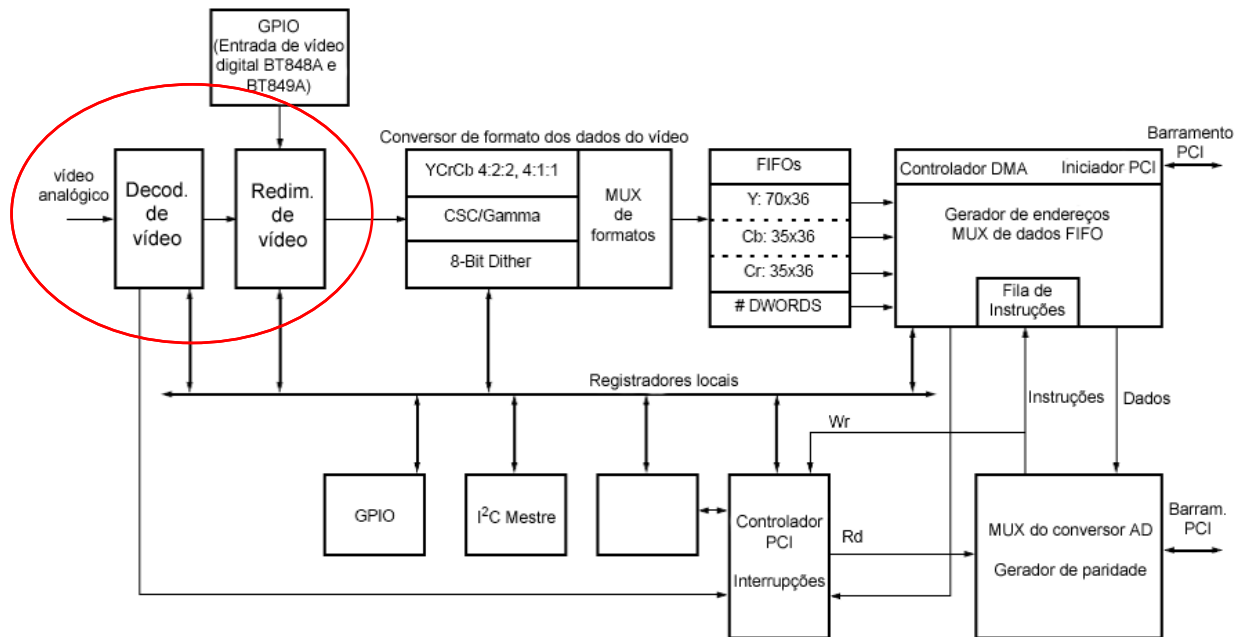


Figura 4.2: Diagrama em blocos detalhado do circuito integrado Bt848, destaque para os dois blocos de entrada [22]

Os sinais de vídeo analógicos são mais comumente apresentados num formato conhecido como vídeo composto (CVBS – *Composite Video Baseband Signal*, no vernáculo Vídeo Composto em Banda Básica). Nessa modalidade, estão combinadas a luminância (Y) e a crominância (C) em um único sinal transmitido por um condutor coaxial como podemos observar na Figura 4.3 a. Quando da chegada desse sinal aos equipamentos de digitalização ou de reprodução, faz-se necessária a separação das componentes Y e C pois, para que a exibição das imagens seja realizada, os mesmos devem ser tratados de forma distinta como veremos a seguir.

A solução mais simples para que essa separação seja realizada é o emprego combinado de um filtro passa-baixas e outro passa-faixa. O primeiro, como se pode ver na Figura 4.3 a) irá retirar a componente de luminância que ocupa a região de baixa frequência do espectro. Um filtro passa-faixa realiza então a filtragem da informação de cor. Em função de limitações na implementação desses filtros, uma parte dos sinais de luminância e/ou crominância sempre é perdida e/ou misturada ao outro de forma inseparável, tanto durante o processo de modulação para a criação do sinal composto, quanto no seu processo inverso de separação das duas componentes. Em função dessas limitações, que causam perda na qualidade da imagem final, foi proposta uma outra forma de transmissão dos sinais de vídeo, o S-Vídeo [15]. Nesse padrão as

componentes Y e C são conduzidas separadamente por pares de condutores distintos blindados por uma armadura coaxial. Na Figura 4.3 b) podemos observar a maior facilidade e especificações menos críticas dos filtros de separação das componentes Y e C do sinal de vídeo. Como consequência imediata dessa separação intrínseca de Y e C, tem-se um aproveitamento máximo dos sinais de cor e luminância, proporcionando uma experiência visual mais detalhada.

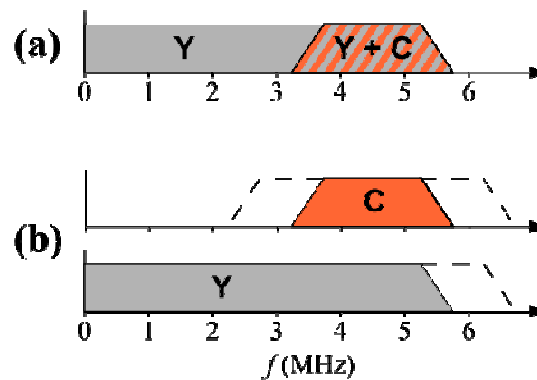


Figura 4.3: Comparação entre os espectros de freqüência dos sinais de:
a) vídeo composto e b) S-Vídeo

Como podemos observar na Figura 4.4, o *chipset* Bt848 recebe os sinais de vídeo analógicos através de um multiplexador de entrada. Via de regra, as placas internas de captura de vídeo, especificamente aquelas chamadas PC-TV, possuem um sintonizador interno capaz de sintonizar estações de TV e de rádio FM. Além disso, oferecem entradas para vídeo composto (RCA) e S-Vídeo permitindo a captura de vídeos de fontes externas. Nasce daí a necessidade do *chipset* apresentar a possibilidade de comutação entre esses sinais. Acompanhando o diagrama em blocos da Figura 4.4, é interessante notar que os sinais de vídeo composto (Y+C) e a componente de luminância do sinal S-Vídeo entram no multiplexador de entrada, ligado por sua vez ao conversor analógico digital (A/D) de luminância. O sinal de cor (C) proveniente do sinal S-Vídeo por sua vez, já é tratado separadamente, sendo ligado ao conversor A/D de crominância.

problema, as placas que oferecem a opção de captura de sinais com interconexão vídeo componente realizam a seguinte adaptação: utilizam um dos conversores A/D para o sinal de luminância, sinal mais percebido pela vista humana; e o outro conversor de forma compartilhada entre as componentes diferença de cor Pb e Pr, através de uma técnica de multiplexação no domínio do tempo. A técnica funciona bem pois, como veremos mais adiante, a vista humana tolera muito bem a chamada sub-amostragem de cor, técnica largamente utilizada inclusive nos algoritmos de compressão de imagem [4].

Vencida a fase de comutação do sinal de entrada, os sinais Y+C (vídeo composto) ou Y e C (S-vídeo) passam para o bloco de conversão analógica/digital onde os sinais são submetidos ao controle automático de ganho (AGC ou CAG) e pela filtragem anti-alias (passa-baixas) para finalmente serem submetidos aos conversores A/D. Estes, via de regra, apresentam resolução de 8 a 11 bits com frequência de amostragem de cerca de 27MHz e banda passante de 12MHz. Com o auxílio de uma base de tempo implementada com a técnica de PLL (*Phase Locked Loop*), os conversores A/D digitalizam apenas a porção de interesse do vídeo, utilizando os pulsos de sincronismo horizontais e verticais como referências para a digitalização. Como se pode observar pelo valor da banda passante (12MHz), o sinal de vídeo considerado para fins de digitalização ocupa uma banda real de 6MHz. Para evitar o problema de sub-amostragem, são implementadas no próprio multiplexador de entrada, filtros anti-alias com frequência de corte de 3dB em 8MHz.

Depois de digitalizados, tanto Y ou Y+C quanto C são levados ao bloco de filtragem digital (passa-baixas) e em seguida ao bloco de separação das componentes Y e C. Um circuito de Controle Automático de Ganho (CAG ou AGC) é adicionado ao chipset para permitir um ajuste na amplitude dos sinais de entrada, otimizando a excursão do sinal e, com isso, reduzindo ao máximo o ruído de quantização.

4.2 Processamento Digital do Vídeo

Digitalizados os sinais de luminância, crominância, áudio e informações contidas no intervalo de retraço, passam os dados à seção digital do *chipset* onde serão processados e tratados.

Na seção anterior tratamos de como ocorre a digitalização das informações analógicas de luminância e crominância sem tocar numa espécie de dados que também é digitalizada. Essa

aparente omissão, na verdade, foi oportuna em virtude de sua natureza originalmente digital como veremos a seguir.

4.3 Processamento das informações contidas no intervalo do retraço vertical (VBI – *Vertical Blanking Interval*)

Além de digitalizar a informação visível do sistema de televisão analógica, ou seja, a imagem propriamente dita, as placas de captura de vídeo também são capazes de decodificar as informações contidas no chamado VBI (*Vertical Blanking Interval*) ou intervalo do retraço vertical. Previsto no sistema original de TV analógica em razão da inércia magnética das bobinas defletoras verticais, o intervalo de retraço que dura cerca de 1,6ms ou 25 linhas de $64\mu\text{s}$ (vide Fig.4.5) acabou sendo aproveitado, com a evolução tecnológica, para a transmissão de informações digitais como *teletext*, *closed caption* (legenda oculta), mecanismos de proteção contra cópias (*copyright*), sinalização de formato de “tela larga” (*wide screen signaling* ou WSS), etc. Em razão de se tratar de informações digitais, esses dados são separados logo após a conversão analógico/digital, sendo enviados a um bloco normalmente denominado *VBI slicer* ou fatiador do intervalo de retraço vertical como podemos observar na Figura 4.6.

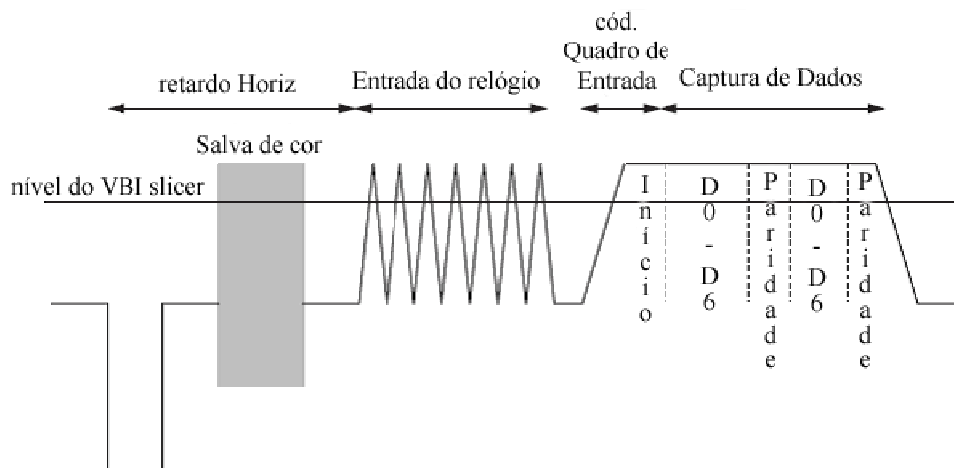


Figura 4.5: Forma de onda típica de uma linha de retraço contendo informações VBI

serviços exploram a porção do VBI do sistema de TV e funcionam com taxas de transmissão de dados bastante baixas, variando com o sistema e serviço utilizado, cuja ordem de grandeza gira em torno de 1Kbit/s. Tratam-se de serviços com características bastante peculiares pois, pelo fato de funcionarem em regime de ampla difusão, não sofrem de problemas de interrupção dos serviços de acessos em massa como as redes de telefonia ou Internet.

4.4 Fluxo do processamento digital do vídeo

Após a digitalização é realizado um novo ajuste no nível dos sinais, dessa vez digital. A partir daí inicia a adequação do sinal digital proveniente dos conversores A/D. O bloco de adequação inicial dos sinais recebe comandos de blocos posteriores funcionando em malha fechada. Isso inclui ajustes dos níveis de entrada de branco e preto provenientes do bloco de tratamento da luminância e nível DC. Para tanto, utiliza como referência, assim como no sistema analógico, os níveis padrões do sinal de televisão, por exemplo, o pórtico traseiro. A partir de seu valor, ajusta o nível DC do sinal de vídeo como um todo.

A próxima fase é a conversão da taxa de amostragem proveniente da referência de frequência ditada pelo cristal do conversor A/D para as taxas padronizadas de pixels. Além de ajustar a frequência de pixels, esse bloco tem de se preocupar em fazer com que os pixels continuem guardando sua posição relativa nas linhas, ou seja, a nova amostragem não pode gerar um deslocamento demasiado dos pixels dentro da linha, tampouco um transbordo de um pixel de uma linha para outra.

Cabe aqui um parêntesis para a análise dos três padrões de frequência de pixels mais difundidos, são eles: o ITU-R BT.656 que estabelece uma frequência de pixels de 13,5MHz, estabelecendo um total de 858 pixels por linha sem contudo estabelecer um número de linhas por quadro; o padrão de 525 linhas por quadro, nativa do padrão NTSC, PAL-M e PAL 60, com pixels quadrados⁶, que estabelece uma frequência de pixels de 12,2727MHz; e o padrão de 625 linhas por quadro, nativa dos demais padrões PAL e SECAM, também com pixels quadrados com frequência de 14,75MHz. A etapa de conversão da taxa de amostragem envolve outros aspectos críticos do processo. Um subsistema crítico em uma placa de captura de vídeo é o seu

⁶ Pixel quadrado é aquele cujas dimensões horizontal e vertical são iguais. Apesar de aparentemente óbvia, essa especificação é necessária haja vistas que alguns padrões de difusão digital de imagens, notadamente o 480i e 576i por exemplo, utilizam pixels retangulares.

PLL de vídeo. Como já descrito anteriormente, o PLL de vídeo é responsável pela geração de diversas referências de tempo (*clocks*) do circuito, entre eles o da taxa de criação de pixels. Existe uma relação bastante rígida entre o *clock* do PLL e a taxa de criação de pixels uma vez que o *clock* gerado pelo PLL vai acompanhar a seqüência de pixels desde a conversão da taxa do conversor A/D até sua chegada ao filtro de decimação já na entrega dos dados ao barramento PCI. Esse compromisso é de fundamental importância para o hardware uma vez que o fornecimento de uma taxa de dados constante é essencial na utilização de *codecs* (*enCoder/Decoder*) como o MPEG. Como foi dito, o *clock* constante do PLL acompanha os dados até o filtro polifásico de decimação, a partir daí, a taxa de dados de saída é criada independentemente do relógio do PLL em razão da flutuação entre o ajuste da taxa de conversão de dados vindos do conversor A/D e o relógio do PLL. Para garantir uma taxa constante, apesar dessas flutuações, existe uma FIFO mantida sempre pela metade que é capaz acomodar essas variações transientes. Fechado o parêntesis sobre a geração da frequência de geração de pixels, voltemos ao fluxo do processamento digital do vídeo.

Uma última tarefa é realizada antes da separação da luminância e crominância, a detecção dos pulsos de sincronismo vertical, de forma a garantir o controle de que campo está sendo digitalizado (ímpar ou par) permitindo o sincronismo de todos os contadores.

A separação das componentes Y e C (Figura 4.6) se dá após a digitalização e se aplica apenas aos sinais de vídeo composto pois os sinais já separados do padrão S-Vídeo passam direto por esse bloco. A técnica mais comum para se realizar essa separação (Figura 4.7) é a associação da utilização de um filtro do tipo rejeita faixa estreita (*notch filter*) sintonizado na portadora de crominância (que varia de acordo com o sistema de cor empregado) para se obter a luminância e um filtro passa-faixa para a filtragem do sinal de crominância que é então seguida pela demodulação do sinal de croma nos sinais de diferença de cor U e V para os padrões PAL e SECAM ou I e Q para o padrão NTSC. Após a demodulação do sinal de cor, é aplicado um filtro do tipo *comb* ou pente também chamado construtivo-destrutivo, em razão da sua resposta em frequência de picos e vales regularmente espaçados. O intuito da aplicação desse filtro é realizar com precisão a remoção das frequências harmônicas residuais de crominância, entremeadas intencionalmente pelo sistema no sinal de luminância e vice-versa, conferindo maior nitidez à luminância e melhor nitidez e separação das matizes primárias da imagem ao sinal de croma (vide Fig.4.8).

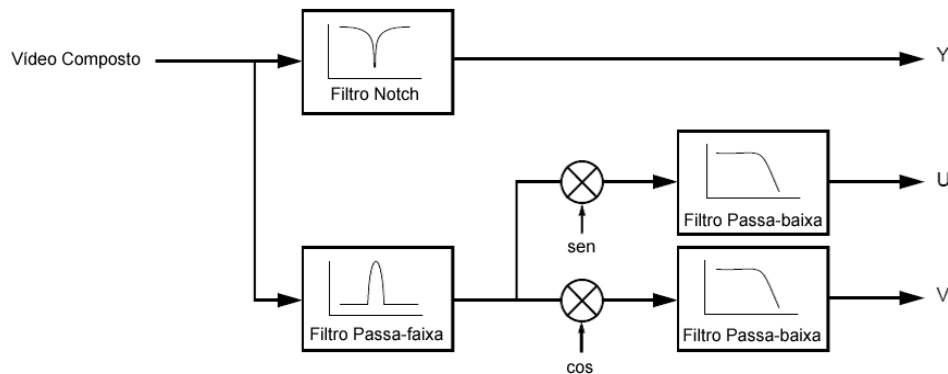


Figura 4.7: Esquema de separação das componentes Y e C

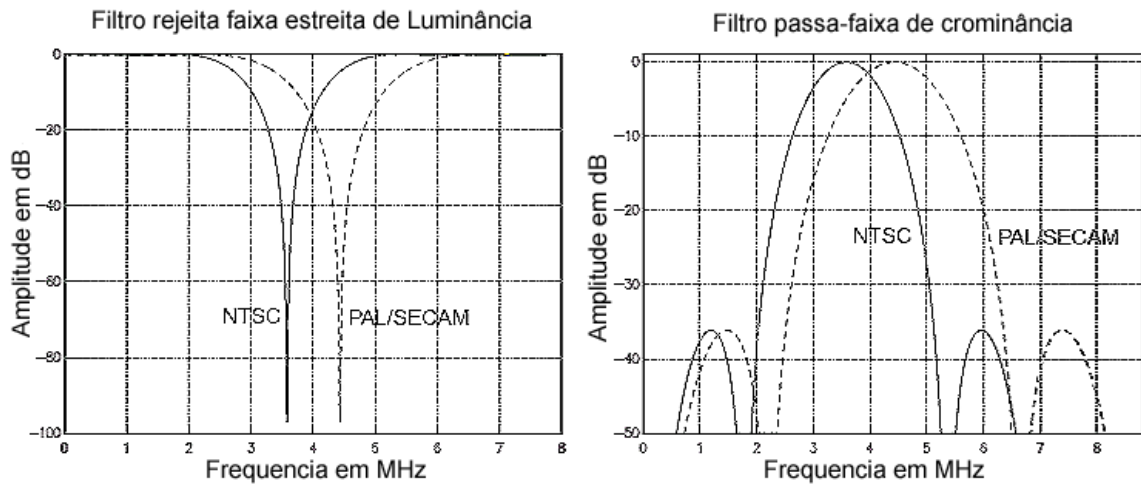


Figura 4.8: Resposta em frequência dos filtros de separação das componentes Y e C:
 À esquerda: filtro rejeita faixa estreita de luminância;
 À direita: filtro passa-faixa de crominância

Além da técnica de filtragem descrita, que é conhecida por “*Luma Notch and Chroma Comb*”, existem outras empregadas em placas de captura de vídeo. Exemplos dessas técnicas são: o filtro pente de duas linhas; o filtro pente de três linhas; o filtro pente bidimensional adaptativo de três linhas; e mais modernamente o filtro pente 3D; técnicas que descreveremos sucintamente a seguir:

- O filtro pente de duas linhas – Esse filtro soma linhas consecutivas se aproveitando do fato de que linhas próximas apresentarem praticamente a mesma informação além de explorar a característica de inversão da fase do sinal de cor a cada nova linha. Desta forma, partindo da premissa razoável de que as informações de linhas próximas é bastante parecida, somando-se essas linhas, a

componente de cor simplesmente desaparece, deixando apenas o sinal de intensidade. O contrário também pode ser feito se ao invés de somarmos essas linhas as subtrairmos, com isso, a componente de intensidade desaparece deixando apenas a componente de cor;

- O filtro pente de três linhas – Utiliza as mesmas características de inversão de fase e constância no conteúdo das linhas empregando uma média ponderada entre a intensidade plena da linha atual e metade da intensidade das linhas anterior e posterior. Com isso se obtém um resultado melhor do que o filtro anterior, no entanto, ainda apresentando transições não tão nítidas quanto o próximo filtro;
- O filtro pente 2D adaptativo de três linhas – Seu maior mérito é o de tornar o *dégradé* de pontos quase imperceptível nas transições de cores na imagem. Isso é feito através de um processo de flexibilização dos pesos 50 – 100 – 50 descritos na técnica de três linhas. Os pesos atribuídos variam ao longo da linha de acordo com as semelhanças entre as linhas, atribuindo alta importância a linhas semelhantes para descartar croma/luma e penalizando linhas diferentes, daí a razão dessa técnica ser chamada de adaptativa. É possível, assim, eliminar o *dégradé* de pontos tanto de cor como de intensidade em grande parte da imagem e tornar o restante quase imperceptível;
- O filtro pente 3D – A técnica aqui empregada utiliza a informação do quadro anterior e/ou seguinte para buscar redundâncias causadas por objetos ou cenas estáticas na imagem que por características do sistema também apresentarão fase invertida em relação ao quadro atual, possibilitando o buscado cancelamento de uma das componentes (luma/croma). Este filtro apresenta performance perfeita para regiões estáticas da imagem e desempenho compatível com o filtro pente 2D adaptativo de três linhas, componente obrigatório para um bom filtro pente 3D, para as regiões onde ocorre movimento. Outro aspecto interessante explorado por este processo é que a vista humana é particularmente sensível a detalhes em imagens estáticas, não sendo verdadeiro o mesmo quando se trata de cenas com grande dinamismo. O processo como um todo é complexo e envolve um *buffer* de memória suficientemente grande para armazenar três quadros (anterior, atual e posterior) além de exigir circuitos integrados dedicados a essa tarefa, capazes de

realizar a comparação entre todas as linhas. Diz-se tridimensional por se valer das linhas anterior/posterior (coordenada Y), variar o peso das linhas ao longo da linha horizontal (coordenada X) e verificar se em quadros adjacentes (anterior/posterior) sua respectiva linha apresenta características semelhantes (coordenada tempo).

De acordo com as informações constantes em sua ficha de dados, o CI Bt 848 utiliza a técnica do filtro *notch* com filtro passa-faixa associada ao emprego de um filtro pente, no entanto, não são oferecidas maiores informações acerca de qual das técnicas é utilizada. Ao que tudo indica, trata-se de um filtro de duas linhas. Outros integrados mais modernos como o CX23880 da Conexant já trazem implementados filtros pente adaptativos 2D multi-linhas, sendo as placas que empregam os filtros 3D mais raras e caras.

4.5 Processamento do sinal de luminância

Logo após a separação dos sinais de luminância e crominância, o sinal de luminância passa pelo bloco de ajuste fino dos níveis de branco, processo que é chamado de “*White crush*” como pode-se observar na Figura 4.9. Após este ajuste que se utiliza do pórtico posterior (*back porch*) como referência para a equalização do nível de branco, essa parte do sinal é então removida do sinal de vídeo.

A seguir a luminância é filtrada com o intuito de evitar, mais uma vez, o efeito alias. O ponto de corte do filtro é selecionado de acordo com o padrão da nova amostragem da taxa de pixels utilizado, pois, como já observamos, o mesmo influencia diretamente a banda passante do sinal de luma. Após o filtro anti-alias, o sinal passa por outro filtro, dessa vez para dar ênfase aos sinais de alta frequência, o chamado *peaking filter*. Por fim, o sinal passa pelos blocos de ajuste de contraste e brilho. No primeiro é ajustado o contraste da luminância através da multiplicação do valor das amostras por constantes de contraste selecionadas pelo usuário ou automaticamente ajustadas pelo circuito. Além desse ajuste, esse bloco realiza a importante tarefa de converter o padrão YUV (NTSC) ou YIQ (PAL) no espaço de cor digital YCrCb. Além do ajuste de contraste, o brilho também é equalizado no último bloco de processamento do sinal de luminância.

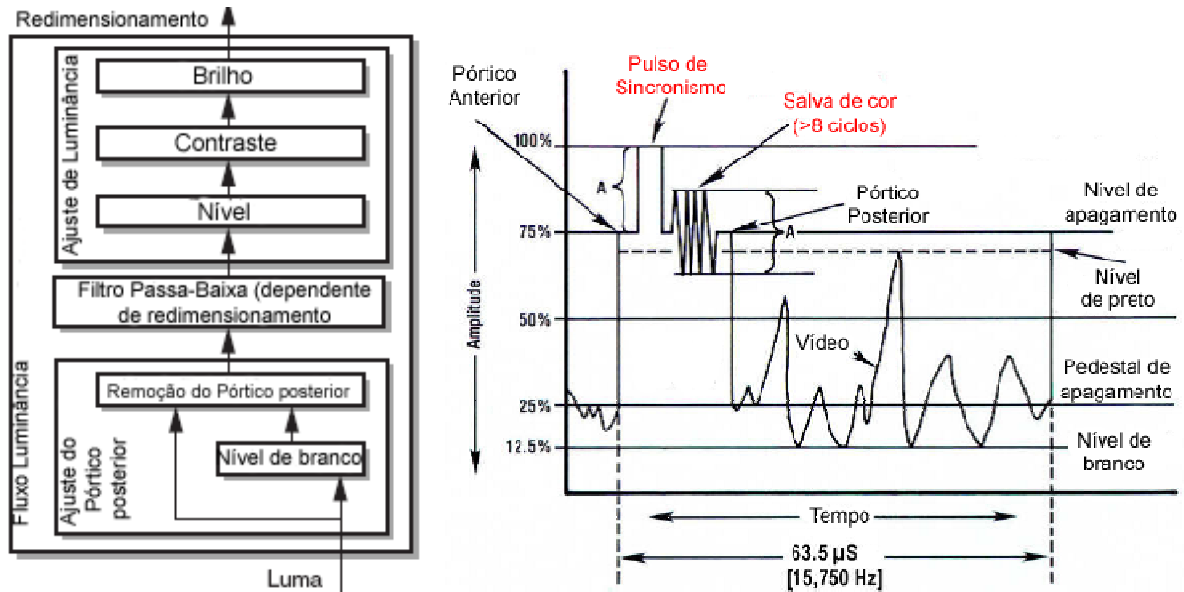


Figura 4.9 e 4.10 – Bloco de processamento do sinal de luminância e detalhe de uma linha horizontal

Abrimos aqui um parêntesis para discorrer um pouco sobre do sistema de cores YCbCr. Trata-se de um sistema digital de cores que emprega a luminância (Y) e os sinais de diferença de cores Cb (B-Y) ou crominância azul e Cr (R-Y) ou crominância vermelha. Na realidade, o sistema YCbCr usa a base RGB para representar as cores, sendo a componente G obtida a partir da conhecida relação da luminância com as componentes B e R, assim como no sistema analógico YPbPr. Apesar de usar a mesma base de cores do RGB, o YCbCr as codifica de forma diferente, explorando a maior sensibilidade do olho humano a variações de intensidade do que à variação das cores propriamente ditas. Assim, enquanto no padrão RGB cada pixel tem valores diferentes para cada uma das componentes R, G e B, no YCbCr vários pixels compartilham mesmos valores de Cb e Cr apresentando valores de Y distintos. Dessa forma, reduz-se a grande redundância da representação RGB explorando essa característica da visão humana. Os sistema que utilizam o espaço de cores YCbCr codificam um sinal de luminância com grande resolução componente que ocupa uma banda larga e codificando os sinais de cores Cb e Cr em sub-bandas reduzidas com menor resolução devido à compressão ou sub-amostragem das cores.

Tanto o padrão para vídeo analógico YPbPr quanto os digitais YCbCr e RGB são definidos pela norma ITU-R BT.601 [8] abreviadamente chamada BT.601 que veremos em seção específica.

Além do padrão BT.601 existe outro bastante famoso o ITU-R BT.709 [9] que estabelece um espaço de cores YCbCr diferente para o sistema de TV de alta definição (HDTV). Essa norma além de estabelecer detalhes no novo padrão digital de transmissão em alta definição, altera os valores das constantes de cromaticidade para uma melhor adaptação aos novos tubos de raios catódicos (CRT) e displays baseados nas tecnologias de plasma e cristal líquido (LCD).

Vários são os codificadores que utilizam o espaço de cores do padrão YCbCr, o mais famoso, sem sombra de dúvida, é o MPEG. Além deste, também se utilizam dessa representação as transmissões nos padrões IEEE1394 e interfaces digitais seriais SDI (*Serial Digital Interface*) de vídeo digital nos formatos MiniDV, DV, etc.

4.6 Processamento do sinal de cromaticidade

O processamento da cromaticidade segue o diagrama em blocos da Figura 4.10 e se inicia com os sinais U e V ou I e Q provenientes do bloco de separação da luminância e cromaticidade. A partir daí o sinal é tratado de forma diferente dependendo do sistema de cor, em especial, no caso do sistema SECAM. No caso de outros sistemas (PAL ou NTSC), é realizado um ajuste na matiz das cores. O processamento da componente de cor do sinal de vídeo do sistema SECAM é processado parte no bloco de separação de luminância / cromaticidade e parte no bloco de processamento de cor. Em razão de fugir do escopo desse trabalho, nos limitaremos a esclarecer que esse tratamento especial se faz necessário pois no sistema SECAM o sinal de croma é modulado em FM, exigindo demodulação para a obtenção dos sinais Db e Dr.

Através do bloco de controle de ganho (AGC), a malha realimenta o amplificador de entrada ajustando a amplitude da componente de cor de acordo com o valor padrão de U obtido no *burst* de croma de cada linha horizontal. Além disso, é implementado um filtro com uma janela de alguns quadros para evitar variações bruscas de cor nas imagens. Assim, a malha garante um sinal de cor mais estável.

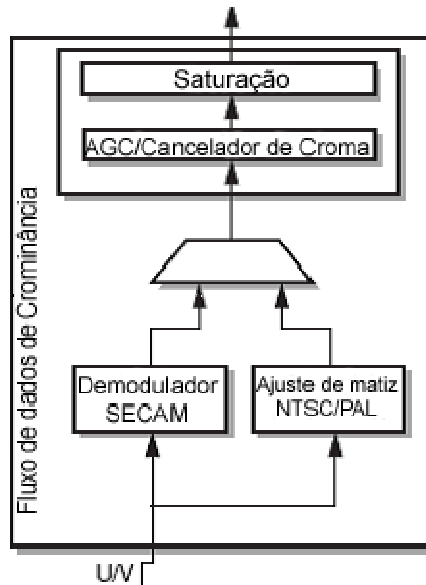


Figura 4.11 – Bloco de processamento do sinal de crominância.

Uma função muito interessante no processo de digitalização de imagens, em especial daquelas provenientes de câmeras de vigilância é a chamada “*Chroma Kill*”. Ela entra em cena cancelando a componente de cor toda vez que o *burst* de croma apresenta nível muito baixo, especialmente quando este está abaixo de um limiar estabelecido. Com isso, evita-se que o AGC de croma realize uma falsa detecção de uma suposta subportadora de cor ou no caso de sinais de vídeo cuja componente de croma apresente-se bastante atenuada ou deteriorada esta seja detectada como colorida. Como resultado de ambos os casos, deseja-se evitar que o AGC amplifique a subportadora de cor falsa ou deteriorada e acabe por inserir ruído em imagens preto e branco contaminando a luminância. Essa falsa detecção é bastante comum em processos de digitalização e acaba por deteriorar a nitidez de imagens com boa resolução de luminância.

Por fim, o processamento do sinal de crominância resume-se a ajustar a saturação de forma independente das componentes U e V.

4.7 Redimensionamento Horizontal e Vertical

Seguindo o fluxo do sinal pelo bloco de processamento digital do vídeo nos deparamos com os blocos de redimensionamento horizontal e vertical. No caso da decimação horizontal, o objetivo é realizar uma redução no número de pixels por linha, enquanto que no caso da decimação vertical o intuito é realizar uma redução no número de linhas do campo ou quadro.

Ambos os processos são sincronizados com a taxa de pixels gerada durante o processo de digitalização, que até esse ponto do fluxo de dados permanece com a sua resolução máxima. A dupla decimação tem como objetivo reduzir o tamanho ocupado pelos quadros, sobretudo quando se opera em um sistema de vigilância no qual é realizado o armazenamento de muitos quadros, compondo extensos arquivos de vídeo. Via de regra, esse redimensionamento da resolução horizontal e vertical dos quadros é realizado de forma independente para as resoluções horizontais e verticais que podem ser controladas pelo software de captura. A independência do processamento também se aplica aos sinais de luminância e crominância guardadas, no entanto, as limitações do sistema analógico originário.

O método aplicado no redimensionamento tem que ser simples e rápido para não comprometer a taxa de quadros gerada pela placa. Uma solução bastante simples consiste em descartar pixels e linhas dos quadros. Dependendo do número de pixels e linhas excluídos, é possível realizar a adaptação das resoluções requeridas. No entanto, tendo em vista que esse procedimento limita-se a descartar parte da informação, descontinuidades seriam geradas em ambos os sentidos, criando componentes de alta frequência no sinal, comprometendo sua qualidade e não se mostrando eficazes no que diz respeito ao fator de compressão do vídeo.

Para resolver essa limitação, no caso da decimação horizontal, um método baseado na interpolação inteligente pixel-a-pixel é normalmente utilizado nas placas de captura de boa qualidade. O método chamado interpolação polifásica explora redundâncias em pixels vizinhos realizando uma primeira compactação da imagem já em hardware, orientada pela posição dos pixels na linha e no quadro.

Uma tarefa importante realizada na fase de decimação horizontal é a compressão da crominância para a geração dos sinais 4:2:2 estabelecidos pela ITU-R em seu padrão BT.601. Trata-se de uma sub-amostragem da componente de cor, com vistas a reduzir a banda passante do sinal de vídeo. O processo, que também é utilizado no sistema de vídeo analógico, explora a menor acuidade visual do ser humano para detalhes coloridos, quando comparada à acuidade visual para detalhes de brilho [4].

Na Figura 4.11 a seguir, pode-se observar como são compostos os pixels nos espaços de cor descritos em negrito no topo das colunas. O primeiro número da representação da sub-amostragem de cor discrimina o fator da taxa de amostragem para o primeiro elemento do espaço de cor, seguido pelos fatores dos outros dois elementos.

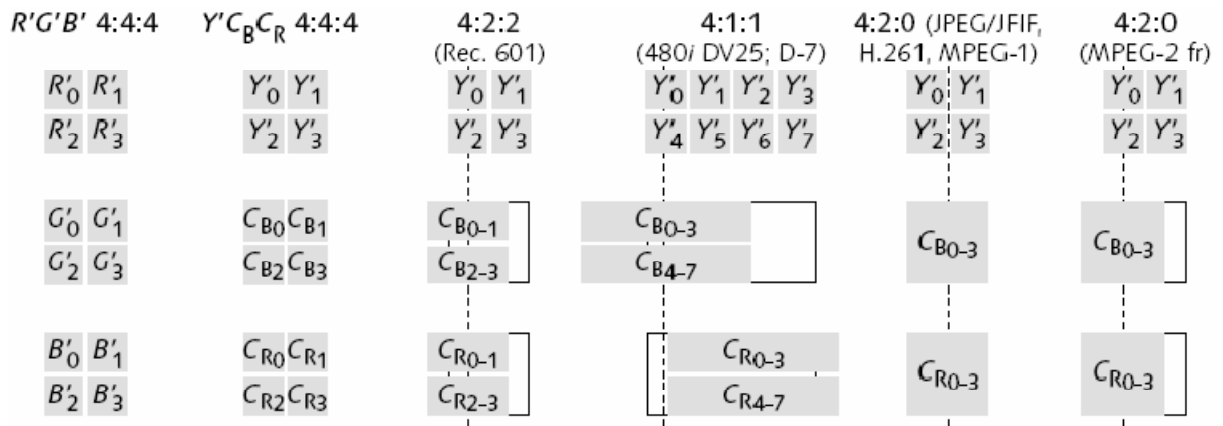


Figura 4.12 – Esquemas de sub-amostragem de cor [7]

A título exemplificativo podemos invocar o caso do vídeo analógico, o sinal de luminância ocupa 4,2MHz, enquanto o sinal de crominância ocupa apenas 1,5MHz, demonstrando que a técnica não é nova. Para a amostragem 4:2:2, os sinais diferença de cor C_b e C_r ou seus respectivos corrigidos pelo fator gama C_b' e C_r' são sub-amostrados à metade da taxa da amostragem da luminância. Se o padrão sem compactação 4:4:4 for escolhido, nenhuma sub-amostragem é realizada e o trem de bits original de 10 bits não é alterado por esse bloco.

Já no caso da decimação vertical, o método é mais dependente do sistema analógico de TV que originou o trem de pixels, em especial nos sistemas PAL e SECAM. No caso do sistema PAL o processo deve armazenar as linhas para que possa realizar médias que conferem a maior precisão de cores desse sistema. Quando o sistema originário é SECAM, também é fundamental o armazenamento das linhas para que as componentes D_r/D_b , que passam as informações de cor em linhas alternadas e que são provenientes da linha de processamento de crominância, possam ser demoduladas e se possa obter finalmente os pares C_r/C_b . Além da adequação ao sistema no que diz respeito à cor, também deve estar prevista a adaptação para o tratamento de vídeo entrelaçado ou não. Após essas adaptações, é realizada, nos mesmos moldes do processo empregado na decimação horizontal, uma interpolação polifásica tanto para luminância como para crominância.

Esta fase do processamento do vídeo tem fundamental importância para orientar os trabalhos de digitalização de vídeo, sobretudo forenses. Por esta razão é recomendável sempre digitalizar o vídeo na resolução máxima nativa do equipamento de digitalização. Com isto, além de se obter amostras com maior riqueza de detalhes, garante-se que esses detalhes são os mais

fiéis possíveis da leitura analógica. Para resolver o problema de tamanho em disco, pode-se empregar, já no computador e em ambiente mais controlado, alguma técnica de compactação.

4.8 Bloco de formatação do vídeo de saída

O último bloco do processamento de vídeo no dispositivo de captura é o chamado VOF (*Video Output Formatter*) ou formatador do vídeo de saída. O bloco VOF é o responsável por adequar a seqüência de bits de brilho e cor que a ele chegam em um dos padrões de saída de vídeo selecionados pelo usuário. No caso do *chipset* Bt848 e em quase a maioria dos demais *chipsets*, a forma de apresentação da seqüência de quadros do vídeo na entrada do bloco VOF é sob a forma de em um bloco contínuo. O espaço de cor é representado pelas componentes YCrCb compactado pela sub-amostragem das componentes de cor Cr e Cb no padrão 4:2:2. Este formato de apresentação, no qual a luminância e as componentes diferença de cor são armazenados juntos na mesma região de memória, é conhecido como modo empacotado (*packed mode*). Ao receber então o vídeo no seu formato nativo, o VOF deve ser capaz de realizar operações de sobre/sub-amostragem das componentes de cor para a conversão nos formatos YCrCb que utilizem compactações de cor 4:4:4, 4:1:1. Além disso, deve ainda estar apto a realizar as conversões entre espaços de cor, como transformar amostras do espaço YCrCb para o RGB (RGB32, 24, 16, 15, etc). No caso do espaço RGB, há ainda, fora a escolha da profundidade de cor, a opção pela aplicação de um filtro de *dither*, tornando o vídeo mais suave, sem transições. Outro formato bastante útil é o Y8, também conhecido como escala de cinza, que é bastante útil nos casos de digitalização de vídeos originalmente preto-e-branco. Nesses casos, a captura das supostas componentes de cor do vídeo acaba por adicionar ruído, reduzindo a nitidez da imagem e aumentando inutilmente o tamanho do arquivo. O processo de implementação do formato escala de cinza, descarta as componentes diferença de cor do trem de dados de entrada e compacta a luminância em 8 bits.

Independente do formato de saída, é sempre possível realizar a separação do bloco único (modo empacotado), formato nativo de dados YCrCb apresentado ao VOF, em três seqüências de dados separadas (modo planificado ou planar). Esse processo de planificação dos dados visa facilitar a aplicação posterior de algoritmos de compressão de vídeo no qual as redundâncias de cada uma das componentes serão tratadas de modo independente. Outra providência tomada independente do formato de saída escolhido é o mapeamento das seqüências de dados em

palavras de 32 bits (*DWORDS*), tamanho este que está associado à estrutura das memórias de quadros das placas (*FIFOs*). Essas palavras de 32 bits apresentadas como saída do bloco VOF, na realidade, são formadas por 4 sub-palavras (pistas) de 8 bits concatenadas. Dependendo do formato de saída, essas sub-palavras de 8 bits podem representar cada uma das componentes dos formatos YCrCb ou RGB ou ainda várias componentes compactadas de 8 bits da luminância (Y), no caso do padrão Y8. Da mesma forma, para alguns formatos, sobretudo aqueles com compressão diferente para as componentes de cor (4:2:2, 4:1:1, etc), mais de uma palavra de 32 bits pode ser necessária para representar um pixel. Um quadro bastante ilustrativo da disposição da informação nas palavras de 32 bits, encontrada na ficha de dados do circuito integrado Bt848 é apresentado a seguir na Tabela 4.1.

Tendo em vista o escopo desse capítulo, que se propõe a esclarecer o que ocorre durante o processo de digitalização de vídeo, não prosseguiremos na análise do fluxo de dados nos blocos de memórias FIFO e seus controle, tampouco nos blocos dos controladores DMA e interface PCI, padrão mais comum nas placas de captura. Resumidamente, nesses blocos os dados sofrem alteração apenas no que diz respeito à sua apresentação, compatibilizando o fluxo de dados com o barramento PCI, que deve compatibilizar o fluxo de dados dos blocos anteriores com as memórias FIFO e o barramento PCI.

Format	DWORD	Pixel Data [31:0]			
		Byte Lane 3 [31:24]	Byte Lane 2 [23:16]	Byte Lane 1 [15:8]	Byte Lane 0 [7:0]
RGB32 ⁽¹⁾	dw0	Alpha	R	G	B
RGB24	dw0	B1	R0	G0	B0
	dw1	G2	B2	R1	G1
	dw2	R3	G3	B3	R2
RGB16	dw0	{R1[7:3],G1[7:2],B1[7:3]}		{R0[7:3],G0[7:2],B0[7:3]}	
RGB15	dw0	{0,R1[7:3],G1[7:3],B1[7:3]}		{0,R0[7:3],G0[7:3],B0[7:3]}	
YUY2—YCrCb 4:2:2 ⁽²⁾	dw0	Cr0	Y1	Cb0	Y0
	dw1	Cr2	Y3	Cb2	Y2
BtYUV—YCrCb 4:1:1	dw0	Y1	Cr0	Y0	Cb0
	dw1	Y3	Cr4	Y2	Cb4
	dw2	Y7	Y6	Y5	Y4
Y8 (Gray Scale)	dw0	Y3	Y2	Y1	Y0
8-bit Dithered	dw0	B3	B2	B1	B0
VBI Data	dw0	D3	D2	D1	D0
YCrCb 4:2:2 Planar	dw0 FIFO1	Y3	Y2	Y1	Y0
	dw1 FIFO1	Y7	Y6	Y5	Y4
	dw0 FIFO2	Cb6	Cb4	Cb2	Cb0
	dw0 FIFO3	Cr6	Cr4	Cr2	Cr0
YUV12 Planar	Sub-amostragem vertical pelo controlador DMA 4:2:2				
YCrCb 4:1:1 Planar	dw0 FIFO1	Y3	Y2	Y1	Y0
	dw1 FIFO1	Y7	Y6	Y5	Y4
	dw2 FIFO1	Y11	Y10	Y9	Y8
	dw3 FIFO1	Y15	Y14	Y13	Y12
	dw0 FIFO2	Cb12	Cb8	Cb4	Cb0
	dw0 FIFO3	Cr12	Cr8	Cr4	Cr0
YUV9 Planar	Sub-amostragem vertical pelo controlador DMA 4:1:1				
Notes: (1). O byte alfa pode ser escrito como 0 ou não escrito (2). UYVY pode ser obtido trocando os bytes					

Tabela 4.1 – Composição das palavras de 32 bits que representam os pixels.

4.9 A escolha dos parâmetros de amostragem dos sinais de vídeo composto

Uma questão bastante recorrente é “com que resoluções (horizontal e vertical) devemos digitalizar os sinais de vídeo provenientes de sistemas analógicos?”. Apesar de ser uma questão aparentemente simples, exige uma análise aprofundada do problema. Refinando um pouco a pergunta inicial, poderíamos ser mais precisos em perguntar: “com que resoluções (vertical e horizontal), profundidade de bits, padrão de espaçamento das amostras e taxa de amostragem devemos digitalizar sinais analógicos de vídeo de forma a não haver perda de detalhes presentes na imagem original?”. Todos esses aspectos da captura digital dos sinais de vídeo geram dúvidas recorrentes no *métier* dos engenheiros de vídeo, tendo algumas sido abordadas em obras como [6] e [7], referências nas quais são analisados alguns aspectos do problema que passamos a tratar.

Antes de qualquer análise, é necessário saber se há procedimentos padronizados para a captura de vídeo analógico. Foi nessa busca que encontramos diversas citações ao padrão ITU BT.601-5 [8] que estabelece parâmetros para a codificação de vídeo em formato SDTV. A recomendação ITU estabelece dois padrões de amostragem que servem tanto para sistemas com 525 linhas quanto para os de 625 linhas, utilizando elementos de imagem representados por 8 ou 10 bits. O primeiro padrão apresenta uma taxa de amostragem de 13,5MHz, compatível no que se refere à largura de banda aos sistemas de transmissão analógicos atuais. O segundo, por sua vez, tem uma taxa de 18MHz, superior à anterior, permitindo uma resolução horizontal maior. Esse segundo padrão surgiu no intuito de adaptar o padrão 13,5MHz à relação de aspecto 16:9, nativa do padrão de TV digital tela larga.

No que se refere ao processo de quantização dos elementos de imagem, o padrão prevê amostras representadas por 8 ou 10 bits. Apesar dessas duas possibilidades, trata durante toda a norma apenas da amostragem de 8 bits. Para essa representação, as amostras são codificadas com 8 bits sem sinal (*unsigned char*), cujo valor decimal varia de 0 a 255 ou em notação hexadecimal de 00 a FF. Dentro desse intervalo, a norma estabelece que o sinal de luminância deve ser representado por 220 níveis alocados na faixa que vai de 16 a 235, dessa forma, a ausência de luminância (preto) é representada pelo valor 16 e a luminância máxima (branco) por 235. No caso dos sinais diferença de cor, esses devem ser representados por 225 níveis alocados na faixa que vai de 16 a 240. Apesar de estabelecidos os valores máximos de 235 e 240, o padrão possibilita a excursão dos sinais além desses valores até 254. Os extremos da representação 0 e 255 foram reservados para representar a sinalização de fim de sincronização.

Ainda para a codificação com 8 bits, essa deve ser uniformemente espaçada seguindo o padrão PCM, regra essa que se aplica para a digitalização da luminância e também dos sinais de diferença de cor. Um aspecto interessante da norma é que ela parte do pressuposto de que os sinais de entrada são os sinais pré-corrigidos com o fator gama, que chamaremos de gama corrigidos ($'$), das componentes de cor. Seguindo a notação adotada pela BT.601-5 os sinais de entrada pré-corrigidos com o fator gama seriam:

$$E'_R, E'_G, E'_B$$

Os sinais de luminância e diferença de cor gama-corrigidos ($'$) podem ser obtidos a partir das fórmulas:

$$E'_Y = 0,299 \cdot E'_R + 0,587 \cdot E'_G + 0,114 \cdot E'_B; [0 \ 1]$$

$$(E'_R - E'_Y) = 0,701 \cdot E'_R - 0,587 \cdot E'_G - 0,114 \cdot E'_B; [-0,701 \ 0,701]$$

$$(E'_B - E'_Y) = -0,299 \cdot E'_R - 0,587 \cdot E'_G + 0,886 \cdot E'_B; [-0,886 \ 0,886]$$

Com o intuito de re-normalizar os sinais de diferença de cor para que os mesmos apresentem um intervalo de variação unitário como o da luminância, os sinais acima corrigidos (C) passam a ser:

$$E'_{C_R} = \frac{0,5}{0,701} \cdot (E'_R - E'_Y) = 0,713 \cdot (E'_R - E'_Y) = 0,500 \cdot E'_R - 0,419 \cdot E'_G - 0,081 \cdot E'_B; [-0,5 \ 0,5]$$

$$E'_{C_B} = \frac{0,5}{0,886} \cdot (E'_B - E'_Y) = 0,564 \cdot (E'_B - E'_Y) = -0,169 \cdot E'_R - 0,331 \cdot E'_G + 0,500 \cdot E'_B; [-0,5 \ 0,5]$$

A partir desses sinais corrigidos os valores dos sinais acima são calculados e arredondados para os inteiros mais próximos:

$$Y = \text{unsigned char } (219 \cdot (E'_Y) + 16); [16 \ 235]$$

$$C_R = \text{unsigned char } (224 \cdot [0,713 \cdot (E'_R - E'_Y)] + 128) = 160 \cdot (E'_R - E'_Y) + 128; [16 \ 240]$$

$$C_B = \text{unsigned char } (224 \cdot [0,564 \cdot (E'_B - E'_Y)] + 128) = 126 \cdot (E'_B - E'_Y) + 128; [16 \ 240]$$

Com isso se obtém os valores das amostras 4:4:4 do sinal. Para se obter as componentes 4:2:2, as amostras já digitais passam por um filtro de sub-amostragem das componentes C_R e C_B .

No que se refere à taxa de amostragem, o padrão prevê 13,5MHz para a luminância e 6,75MHz (padrão 4:2:2) ou 13,5MHz (padrão 4:4:4) para cada um dos sinais diferença de cor. Quanto ao número de amostras por linha ativa, prevê 720 amostras/linha para a luminância e 360(4:2:2)/710(4:4:4) para os sinais diferença de cor.

Mas como o padrão chegou a esse valor de amostras/linha e a essa taxa de amostragem?

Como podemos observar na Tabela 4.2, os padrões dos sistemas NTSC M e PAL M apresentam 525 linhas/quadro e taxa de atualização de 59,94 campos/s (60 campos/s inicialmente no sistema preto e branco, reduzido, com o advento do sistema a cores, para 59,94 visando acomodar o espectro da crominância) e 29,97 quadros/s (NTSC-M colorido), isso implica em uma frequência de varredura horizontal de

$$f_h = 525 \frac{\text{linhas}}{\text{quadro}} \cdot 29,97 \frac{\text{quadros}}{s} = 15.734,25 \text{Hz}$$

No caso dos sistemas PAL (B, G, H, I e N) e SECAM (B, D, G, H, K, K' e L), estes possuem 625 linhas/quadro e taxa de atualização de 50 campos/s e 25 quadros/s, o que resulta na seguinte frequência de varredura horizontal $f_h = 625 \frac{\text{linhas}}{\text{quadro}} \cdot 25 \frac{\text{quadros}}{s} = 15.625 \text{Hz}$

É o que confirma Grob [4], diferente do que a referência [6] nos traz, os sistemas NTSC e PAL possuem frequências de linha nitidamente diferentes.

Segundo Grob [4], a mais alta frequência de vídeo comportada pelos canais padrão (6MHz) de difusão de sinais de televisão analógica é de 4MHz. Tendo por base essa informação e a frequência de varredura horizontal f_h do sistema utilizado, é possível realizar um cálculo da máxima resolução horizontal do sistema de TV analógica. Como os sistemas de captura de imagem disponíveis nos Setores Técnico Científicos da Polícia Federal utilizam equipamentos NTSC-M ou PAL-M, analisaremos o sistema M (525 linhas/59,94campos/s).

A determinação das máximas resoluções horizontal e vertical do vídeo analógico, embora pareça tarefa fácil, envolve alguns detalhes que precisam ser analisados.

Quanto à representação da resolução vertical, o próprio número total de linhas do sistema M (525) nos dá uma forte dica da resolução necessária. No entanto, devemos excluir desse total as linhas que restam apagadas durante o período de retraço vertical. O sistema de 525 linhas realiza duas varreduras de 262,5 linhas cada, dessas, 20 na varredura ímpar e 22 na varredura par são referentes aos intervalos de sincronismo e retraços verticais dos campos ímpares e pares do quadro. Isso nos deixa **483 linhas ativas para a exibição de imagens** e 42 linhas de retraço como vemos na Figura 4.12.

	NTSC M	PAL B,G,H	PAL I	PAL N	PAL M	SECAM B,G,H	SECAM D,K,K',L
Linhas/campos	525/59,94	625/50	625/50	625/50	525/59,94	625/50	625/50
Freq. Horizontal (kHz)	15,734,25	15.625	15.625	15.625	15.734,25	15.625	15.625
Freq. Vertical (Hz)	60	50	50	50	60	50	50
Freq. Subportadora de Cor (MHz)	3,579545	4,43361875	4,43361875	3,582056	3,575611		
Largura de Banda do sinal de Vídeo (MHz)	4,2	5,0	5,5	4,2	4,2	5,0	6,0
Freq. Subportadora de som (MHz)	4,5	5,5	5,9996	4,5	4,5	5,5	6,5

Tabela 4.2 – Características dos sistemas de difusão dos sinais analógicos de televisão.

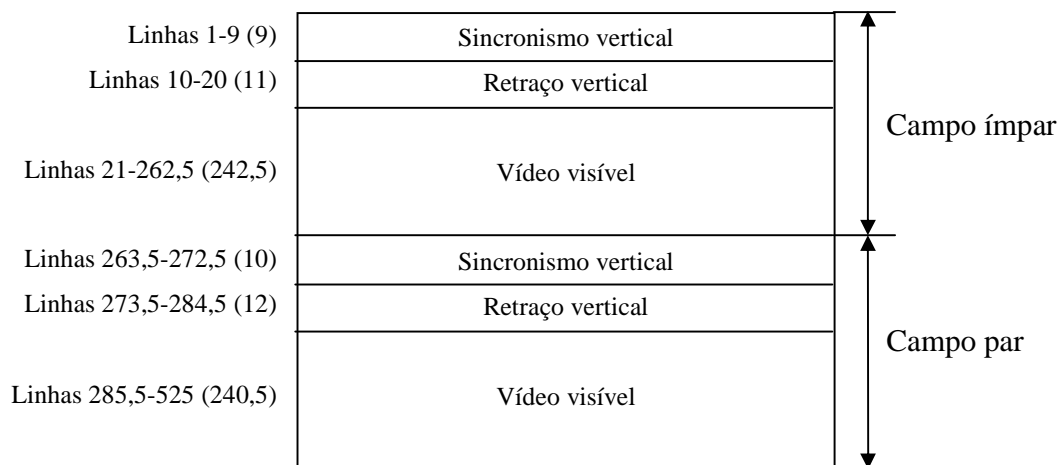


Figura 4.13 – Distribuição das linhas no quadro padrão M

Aparentemente o problema da determinação da resolução vertical estaria terminado se não fosse um fenômeno conhecido como fator Kell ou fator de entrelaçamento. Raymond Kell e outros engenheiros da RCA testavam em 1933 um sistema de televisão com 180 linhas de resolução, varredura progressiva e taxa de atualização de quadros de 24 qps. Durante ensaios utilizando cartões com padrões horizontais e verticais alternados de preto e branco, Kell verificou

que a resolução prática do sistema, originalmente concebido para 180 linhas, atingia efetivamente apenas 64% dessas. O problema (Vide Fig.4.13) que tem base no quase impossível sincronismo perfeito entre o foco da câmera e as linhas dispositivo de exibição (tubo de raios catódicos ou qualquer outra tecnologia) ocorre toda vez que os detalhes de alta frequência localizam-se entre as linhas de varredura quando deveriam coincidir com as mesmas para poderem ser corretamente representados. Quando a resolução do padrão capturado e a sua exibição são muito próximos, esse sincronismo torna-se quase impossível de ocorrer mesmo para filmagens estáticas, daí a necessidade de haver uma certa margem (o fator Kell). O valor originalmente proposto para esse fator (64%) sofreu questionamentos e ataques ao longo da história, tendo sido adotado o valor de 70%.

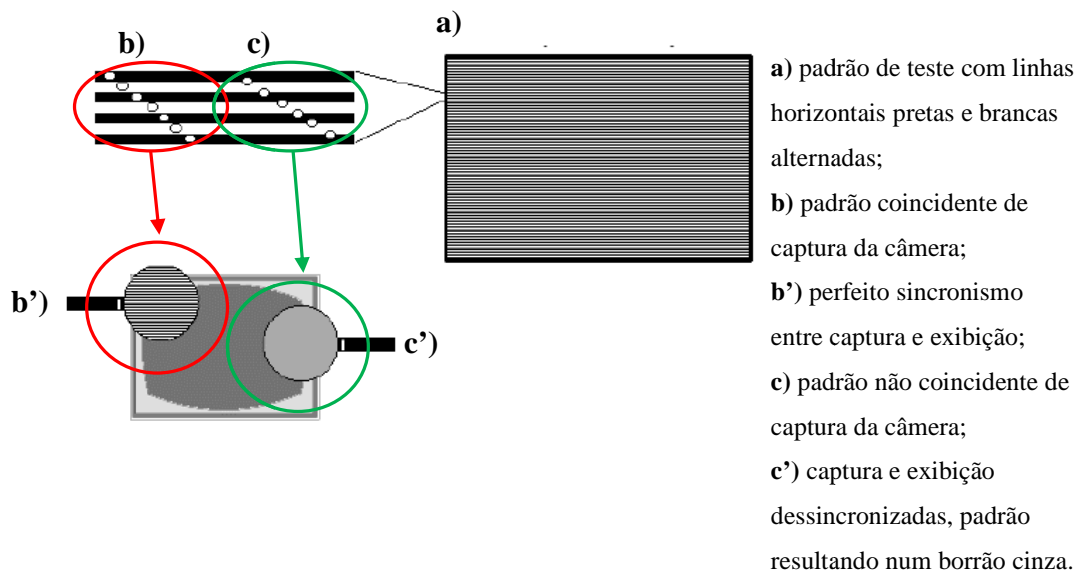


Figura 4.14 – Exemplo clássico da influência do fator Kell

Explicado o fenômeno que decorre do fator Kell, sua aplicação na prática implica na multiplicação do número de linhas ativas obtidas por 0,70, o que nos daria que resolução horizontal, de fato, poderia ser obtida de uma filmagem:

$$\text{resolução vertical} = 483 \cdot 0,7 = 338$$

Muitas referências citam essa resolução vertical como sendo aquela nativa do vídeo derivado do sistema SDTV analógico, afirmando que resoluções acima deste valor nada mais seriam que redundância. Para fins forenses, no entanto, acreditamos ser essa resolução

inadequada, pois o que os exames periciais buscam, via de regra, não é a uma mera vigilância do ambiente, mas as minúcias contidas no vídeo. Ao final dessa seção, descreveremos com maiores detalhes o modelo proposto por esse trabalho e as razões que nos levaram a estabelecê-lo.

Para calcularmos a resolução horizontal precisamos de outras informações do sistema de televisão analógico. Como podemos observar, ainda na Tabela 4.1, os sinais de vídeo ocupam uma largura de banda de 4,2MHz e também que nos sistemas M a frequência de varredura horizontal é de 15.734,25 KHz. Precisamos saber agora o período ativo de uma linha, para tanto recorreremos à norma ITU-R BT 470-7 [10] que estabelece os seguintes valores de temporização de uma linha de um sinal de vídeo analógico composto NTSC M (Figura 4.14):

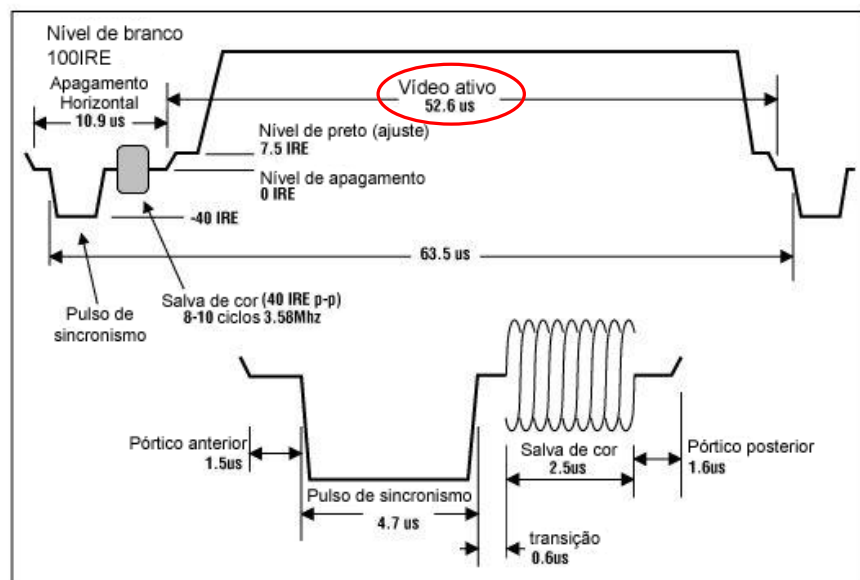


Figura 4.15 – Forma de onda típica de uma linha de um sinal de vídeo composto (ITU-R BT.470-7)

A partir desse valor do intervalo ativo de vídeo de uma linha podemos passar a calcular quantas oscilações por segundo existem no intervalo de uma linha ativa:

$$\text{resolução horizontal} = \frac{4,2 \cdot 10^6 \frac{\text{oscilações}}{\text{s}}}{1} = 220,92 \frac{\text{oscilações}}{\text{linha}} \frac{\text{s}}{52,6 \cdot 10^{-6} \frac{\text{s}}{\text{linha}}}$$

Utilizando o teorema de Nyquist chegamos a taxa de 442 amostras/linha para que seja possível a perfeita reconstrução do sinal.

Tendo em vista que embora o fator Kell reduza em 30% a capacidade de exibição de detalhes nas imagens, para fins de digitalização dos sinais de vídeo, ele não será levado em consideração, uma vez que o objetivo é capturar toda informação disponível na mídia encaminhada, dessa forma, a resolução vertical deverá ser aquela anterior à aplicação do fator Kell, ou seja, 483 linhas ativas.

Antes de concluirmos esta seção cabe ainda uma última consideração no que se refere à relação de aspecto do vídeo analisado. Até aqui, a resolução de captura do sinal analógico de vídeo seria 442 amostras horizontais x 483 amostras verticais, isso implicaria numa imagem quase quadrada com relação de aspecto de aproximadamente 0,915. Sabe-se que a relação de aspecto padrão do sistema padrão de televisão (SDTV) analógico é $\frac{4}{3}$ ou aproximadamente 1,333. Para que durante o processo de digitalização não ocorram distorções devido a essa diferença, faz-se necessário utilizar a relação de aspecto 1,333. Isso implica em fixar a maior resolução (483) e multiplicá-la pela relação de aspecto para encontrar 644 amostras. Dessa forma, aproximando os valores, chegamos a resolução dos equipamentos de captura comerciais que é de 640x480 pixels (VGA).

Depois de analisado o sinal de vídeo no que se refere às suas características, seriam suficientes os parâmetros para a digitalização de sinais de vídeo analógico: resoluções horizontal e vertical: 640x480 pixels. No entanto, durante os trabalhos periciais, não raros são os casos em que é fundamental capturar as linhas do traço vertical. Um caso típico são os sistemas de multiplexação de imagens que gravam informações como data, horário, número da câmera, além de outros dados considerados importantes para o funcionamento do sistema nessas linhas. Infelizmente, não foi esse o caso da fita analisada, mas com alguma frequência fitas com informação nas linhas de traço são apresentadas para exames. Até o momento, a perícia da Polícia Federal não analisou de forma sistêmica o protocolo de gravação dos dados nessa região oculta do vídeo, no entanto, acredita-se que uma vez identificado ao menos o campo de identificação da câmera ativa, a demultiplexação seria relativamente simples. Essas informações são gravadas em formato binário representadas por trechos de linhas com luminância mínima (preto) ou máxima (branco) representando respectivamente zeros e uns como se pode observar na figura 4.16 a seguir.

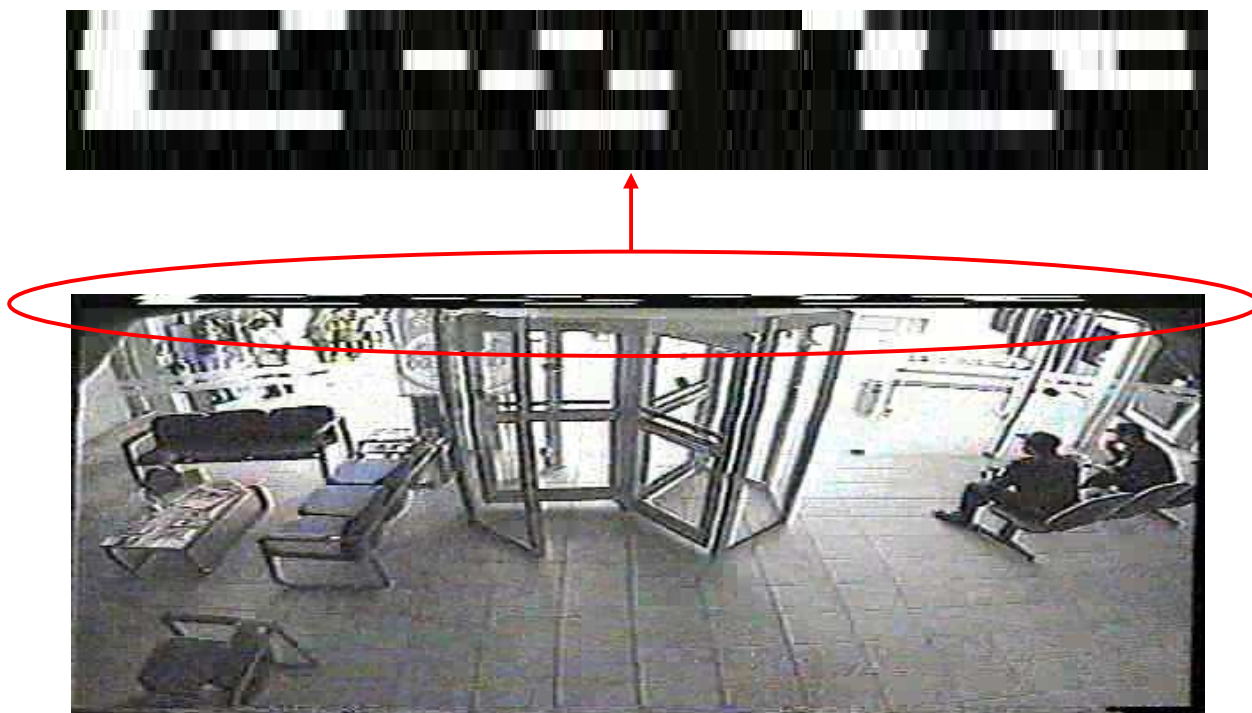


Figura 4.16 – Imagem proveniente de um sistema de multiplexação de vídeo que grava informações nas linhas do retraço vertical. Detalhe das linhas de retraço em zoom fora de proporção para facilitar sua visualização

Para esses casos, não seria suficiente utilizar a regra da resolução de 640x480 pixels pois há informação relevante na região de retraço vertical que no cálculo anterior foi descartada. Deve-se então, no lugar das 480 linhas utilizar como resolução vertical o próprio número total de linhas do padrão M, ou seja, as 525 linhas, sem aplicar qualquer fator de correção. Apesar de comercialmente serem mais comuns as placas com resoluções de 640x480 pixels, é possível encontrar placas de captura de vídeo que disponibilizam resoluções verticais de 576 linhas entrelaçadas que atendam o padrão 576i25 (ou 625/50/2:1). Mais conhecido como padrão PAL digital de SDTV, essas placas atenderiam as demandas nos casos em que as imagens apresentassem informações na região de retraço.

Capítulo 5

Um novo algoritmo de demultiplexação

Depois de contextualizar a situação, as características do sistema de CFTV, descrever o conteúdo da fita encaminhada e analisar pontos cruciais do processo de digitalização de imagens analógicas, passamos a descrever efetivamente os procedimentos e implementações computacionais realizadas no intuito de obter a demultiplexação do vídeo encaminhado.

5.1 Captura

Como já foi dito, o primeiro passo da análise foi a digitalização das imagens constantes na fita. Para tanto, valeram-se os peritos de um videocassete marca JVC modelo HR-S9500U conectado através de sua saída S-Video a uma placa de captura de vídeo da marca Pinnacle, modelo DC-10 PLUS. A placa em questão apresenta arquitetura baseada no *chipset* da Philips SAA7118. Uma rápida observação do diagrama em blocos do CI SAA7118 (Fig.5.1) nos permite verificar que sua arquitetura é bastante similar àquela do CI Bt848, estudada no capítulo anterior com alguns avanços em relação a este.

Em sintonia com o que foi analisado no capítulo anterior, tendo em vista que o vídeo do caso em tela não apresentava informações do processo de multiplexação nas linhas de traço vertical, as resoluções de 640x480 pixels foram escolhidas para digitalizar a região ativa do vídeo. Para auxiliar no processo de captura, juntamente com a placa Pinnacle utilizou-se o software que a acompanha a placa, o Pinnacle Studio 8.

Configurado para realizar apenas a compactação em hardware e digitalizar as imagens com o máximo de minúcias, ao fim do processo o programa gerou um arquivo de vídeo entrelaçado de extensão avi, sem áudio, com imagens de resolução de 640x480 pixels, compactadas pelo processo de sub-amostragem de cor 4:2:2, conhecido como YUY2 que utiliza o espaço de cor delimitado pelas elementares YCbCr. A captura em cores foi forçosamente escolhida apesar do conteúdo da fita ser monocromático, uma vez que o hardware/software nas

versões utilizadas não disponibilizavam a opção do formato Y8 (escala de cinza). Reproduzido à taxa de quadros de 29,97 *fps*, a exibição dos 877.970 quadros capturados dura um período de 08:08:15 (oito horas, oito minutos e quinze segundos), aproximadamente o intervalo para a exibição de todo o conteúdo de uma fita padrão VHS em um videocassete doméstico no modo de velocidade super longa (*Super Long Play – SLP*).

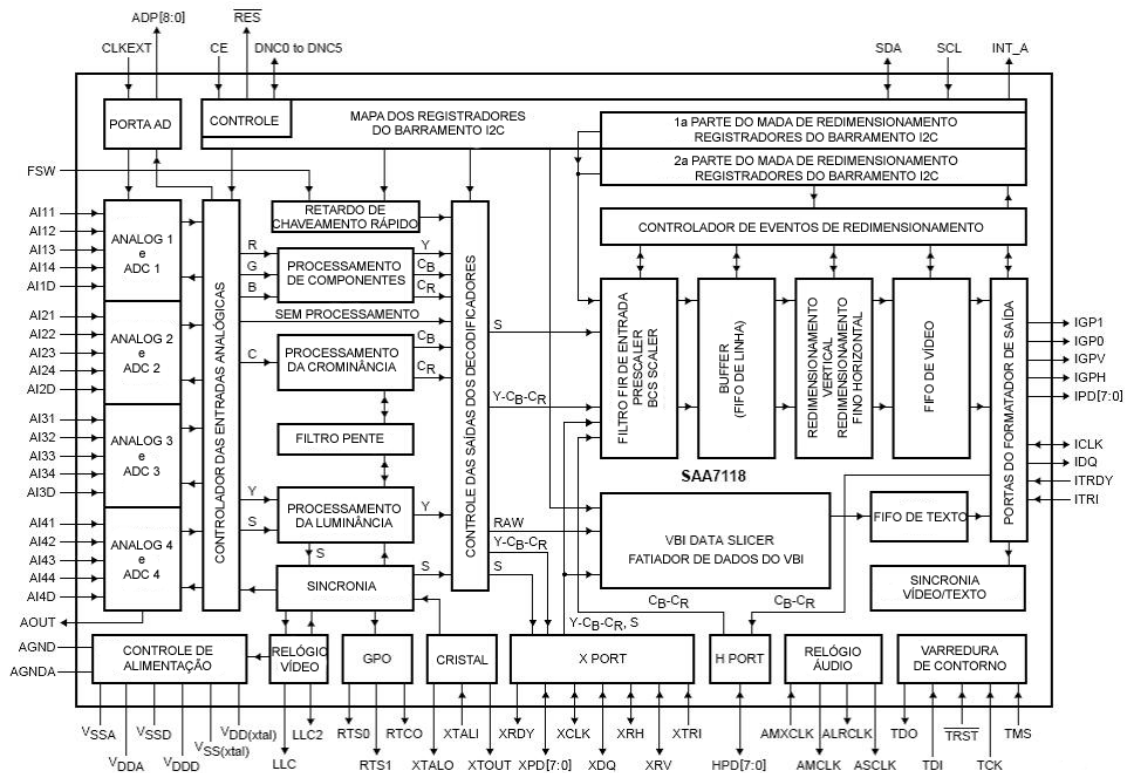


Figura 5.1 – Diagrama em blocos do CI SAA7118 da Philips, bastante similar ao Bt848 da Brooktree

5.2 Pré-processamento

Durante os trabalhos, tendo em vista a urgência em encontrar as imagens do crime para comprovar a materialidade do delito investigado, optou-se por realizar o processamento das imagens com a ferramenta MATLAB. A decisão levou em consideração o grande número de funções já implementadas no programa, próprias para o tratamento de imagens, além de outras que facilitaram e agilizaram sobremaneira as operações matemáticas envolvendo a manipulação de matrizes. Apesar de todas as facilidades elencadas e que levaram à sua utilização do MATLAB em sua versão 7, observou-se uma limitação que foi encontrada durante os trabalhos.

Verificou-se que o MATLAB apresenta alguns problemas ao trabalhar com *codec's* (Codificador/Decodificador) diferentes daquele que é o seu codec nativo, o Indeo. Na documentação do programa constatou-se que *a priori* o MATLAB seria capaz de funcionar com qualquer codificação de vídeo, desde que houvesse menção do mesmo nos comandos de leitura/escrita e que os *codecs* estivessem instalados no sistema operacional em questão. Apesar disso, em pesquisa realizada na base de conhecimentos da Mathworks, empresa que desenvolveu a ferramenta, e em diversos fóruns de usuários do programa [20], verificou-se que o problema é recorrente e que por vezes, como nos trabalhos apresentados, o erro do programa não cessa mesmo depois da instalação dos *codecs* necessários. Em virtude desse obstáculo encontrado, tendo em vista o formato originário do vídeo digitalizado ser, ao que as tentativas demonstraram, incompatível com o MATLAB, optou-se por, realizar uma fase de pré-processamento na qual o vídeo digitalizado seria convertido de seu formato original para o Indeo5. Cabe aqui um breve parêntesis acerca desse codec.

Indeo é, na verdade, a forma enxuta da designação do codec Indeo Video. Originalmente concebido pela Intel para trabalhar embarcado processado por DSPs (*Digital Signal Processors*), sua tarefa era de compactar imagens de vídeo serializado com qualidade de TV. O nicho de mercado do Indeo, quando do seu desenvolvimento, era o de videoconferência e difusão de vídeo digital. Tendo em vista essa aplicação, o projeto do codec primou por tornar o processamento do vídeo do lado do observador o mais leve possível, uma vez que a interação do observador é, via de regra, pequena, daí sua característica assimétrica (processamento mais pesado na codificação e mais leve na decodificação). Uma das prioridades do projeto foi sempre manter a compatibilidade com os diversos sistemas operacionais disponíveis no mercado dos anos 90, sobretudo com aqueles produzidos pela Microsoft, Apple e IBM. Apesar de ter sido referência em sua época áurea, a Intel, depois de algum tempo, e com o advento dos *codecs* MPEG e outros mais eficientes, vendeu os direitos da patente à Ligos Corporation que atualmente é sua proprietária [24]. O codec foi sendo aperfeiçoado, inclusive baseando-se em transformadas *wavelet* em suas versões mais recentes [25]. Suas versões mais conhecidas são as 3.2, 4, 4.5 e 5, estas duas últimas utilizadas neste trabalho. Em virtude de ainda ser um codec de código fechado, protegido por patentes, pouca informação foi encontrada sobre de sua implementação.

Com base no histórico narrado do codec, acreditamos que a opção da Mathworks pelo Indeo fundamentou-se em explorar a questão da compatibilidade com os diversos sistemas

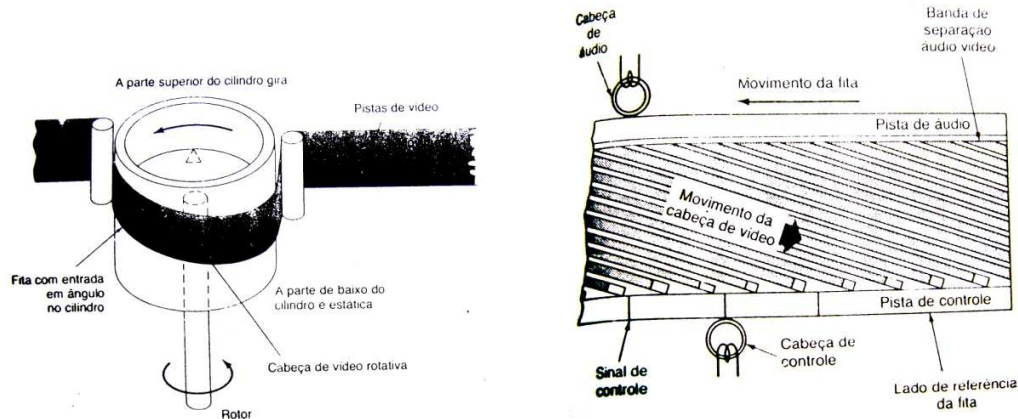
operacionais, que sempre foi uma prioridade do MATLAB. Além disso, o codec Indeo é componente intrínseco das distribuições do sistema operacional da Microsoft em sua versão Windows XP conferindo grande popularidade ao mesmo.

Assim, antes de qualquer processamento do vídeo no formato apresentado pela placa de captura (YUY2), havia a imperativa necessidade de se realizar pré-processamento que, no mínimo, realizasse o desentrelaçamento dos quadros, como discutido e exposto no Capítulo 3, do material sob análise e convertesse o arquivo de vídeo do seu formato nativo YUY2 (YCrCb 4:2:2) para outro que fosse compatível com as rotinas do MATLAB que requerem codificação Indeo5. Além dessas providências, ainda na fase de pré-processamento, tendo em vista características da fita analisada, fez-se necessária a exclusão do seu trecho inicial. Durante as análises, observou-se que o trecho inicial do vídeo gravado na fita e conseqüentemente no arquivo, se apresenta contaminado com um nível de estática suficiente para tornar esse ruído um problema para a detecção das câmeras durante a fase mais sensível do processo de classificação das imagens: seu início. Apesar de estarmos acostumados a observar esse fenômeno nas antigas fitas VHS, buscou-se uma explicação técnica para o fato já que a gravação era relativamente recente. Segundo Grob [4], a contaminação do trecho inicial das fitas magnéticas de vídeo é natural e geralmente é causada durante o processo de alinhamento das cabeças de leitura/trilhas da fita. Durante um determinado período que depende dos equipamentos utilizados, os servomecanismos de controle de velocidade e de fase do cilindro-cabeça do videocassete estão sincronizando a posição angular correta da cabeça de vídeo do referido cilindro com o pulso de sincronismo vertical, gravado na pista longitudinal inferior da fita (pista de controle) vide Figuras 5.2 e 5.3 a seguir.

Esse período de sincronia é anormalmente longo na fita do caso em tela. A explicação encontrada para essa característica foi o fato de que o procedimento de sincronia ocorre duas vezes:

- na primeira vez quando o equipamento de *time-lapse* inicia a gravação das imagens na fita analisada, gravando em seu trecho inicial um período com estática devido ao processo normal de alinhamento das cabeças;
- na segunda vez quando o videocassete reproduzidor utilizado no processo de digitalização do vídeo lê o trecho comprometido com estática no qual não

consegue sincronizar-se. Somente após esse primeiro trecho contaminado quando se inicia a leitura do trecho livre de estática é que ocorrerá o normal sincronismo dos servomecanismos do videocassete leitor;



Figuras 5.2 e 5.3 – Esquemas do sistema helicoidal de varredura em videocassetes: (esquerda) conjunto do cilindro e fita magnética; (direita) revelação das pistas gravadas na fita [4]

Além da limitação em relação ao codec que precisa ser utilizado, verificou-se ainda um outro problema em relação ao MATLAB. Quando o arquivo a ser processado possui tamanho superior a 2GB o MATLAB gera erros de leitura de quadros no comando *aviread* [20]. O problema verificado foi encontrado em fóruns de discussão do software da Mathworks [20] que admite a limitação e sugere uma biblioteca criada por um usuário, que em nossos testes não resolveu o problema além de apresentar limitações em relação à função *aviread* original. A solução encontrada foi pré-dividir o arquivo a ser processado em múltiplos não superiores a 2GB na fase do pré-processamento.

Uma vez identificados os problemas, passamos a descrever o que é realizado no tratamento propriamente dito das imagens que inicia com o pré-processamento das mesmas.

Para realizar as tarefas do pré-processamento (remoção do trecho com ruído, desentrelaçamento e decodificação YUY2 e recodificação Indeo5), utilizou-se os programas gratuitos de código aberto AviSynth [12] e VirtualDub [13].

As tarefas de remoção dos trechos de ruído, decodificação e recodificação do arquivo de vídeo foram realizados com o auxílio do programa VirtualDub. De código aberto e gratuito, o programa permite que se realizem diversos tipos de filtragem e edições em arquivos de vídeo,

além de ser capaz de decodificar/codificar múltiplos formatos de arquivos de vídeo. Apesar de sua versatilidade, não dispõe de um filtro de desentrelaçamento próprio que atendesse as necessidades do trabalho, por isso buscou-se utilizar os recursos do AviSynth. Os procedimentos de desentrelaçamento do vídeo foram realizados com o auxílio do AviSynth, um programa que não possui interface gráfica, e que funciona sempre em conjunto com outro programa que apresente alguma interface amigável, no caso desse trabalho, utilizou-se o próprio VirtualDub. A necessidade do programa auxiliar (VirtualDub) se justifica uma vez que esse, na verdade, executa os comandos (*scripts*) do AviSynth permitindo a visualização prévia do resultado quadro-a-quadro. O processamento do arquivo inteiro é realizado em seqüência (*batch processing*).

Especificamente quanto ao processo de desentrelaçamento, acreditamos ser necessário realizar alguns comentários sobre como acreditamos que o processo precise ser realizado. Na literatura e em diversos sítios voltados para o processamento de vídeo, encontra-se farta referência ao assunto desentrelaçamento [5], [12], [13]. Para as aplicações não periciais, o processo visa resolver o inconveniente das transições abruptas de luminância e cor entre as linhas de um vídeo capturado de uma fonte entrelaçada quando se deseja gerar sua versão progressiva (não entrelaçada). O resultado não-pericial do processo é um vídeo mais homogêneo em sua nova versão progressiva ou desentrelaçada interpolada. O problema é que nas aplicações não-periciais, a finalidade desse processo é melhorar o aspecto do vídeo, tornando-o mais agradável à visão humana. Para se obter essa melhora aparente da imagem deve-se optar por uma das soluções bem conhecidas:

- o simples descarte de um dos campos (par ou ímpar) seguido do redimensionamento do campo remanescente com a manutenção da taxa de exibição de quadros. Alguns inconvenientes decorrem do emprego desta técnica: redução da resolução vertical do vídeo e o seu inconveniente mais grave: o descarte dos campos ímpares ou pares que podem oferecer informações importantes não disponíveis nos campos remanescentes [5], [12], [13];
- o que é mais comum, realizar-se a combinação dos dois campos que formam cada quadro utilizando diversas técnicas, dentre as quais destacamos as mais utilizadas:
 - a chamada mistura (*blending*) dos campos. Nesta técnica cria-se um novo quadro a partir do cálculo da média pixel a pixel RGB dos dois campos,

formando uma imagem híbrida com metade da velocidade de exibição de quadros e metade da resolução vertical. É uma das técnicas mais usuais pois seu processamento é rápido e apresenta resultados aceitáveis para vídeos estáticos ou com movimentos lentos. Para vídeos dinâmicos não apresenta bons resultados pois a sobreposição de campos resulta na formação de fantasmas na imagem. Em virtude da fusão de dois campos em um único quadro, o vídeo passa a ser progressivo (não-entrelaçado), resultando na redução da taxa de exibição de quadros à metade. Outra consequência é a redução da resolução vertical à metade pois o quadro resultado é a média dos dois campos. Em virtude da possibilidade de perda de detalhes e de aparecimento de fantasmas no vídeo, esta técnica não se presta para aplicações periciais [5], [12], [13];

- costura (*weave*) dos campos. Técnica parecida com o *blending*, simplesmente funde dois campos em um quadro sem calcular médias. O resultado é um quadro com resolução vertical plena, taxa de exibição de quadros pela metade mas que finda por misturar o conteúdo dos quadros nas áreas onde há sobreposição e deixar nas áreas onde não há sobreposição as linhas do entrelaçamento. Desta forma, percebe-se que a manutenção da resolução vertical plena fica comprometida: seja pela mistura com perda de detalhes quando ocorre sobreposição dos conteúdos dos campos; seja pela duplicação da resolução do campo através da inserção de linhas estranhas gerando, nessas regiões, serrilhado (linha cheia – linha vazia). Em virtude do comprometimento das regiões onde ocorre sobreposição e do efeito serrilhado das demais áreas do vídeo, também julgamos que tal técnica não se preste ao objetivo deste trabalho [5], [12], [13];
- *Bob*: o chamado efeito bob é decorrente do deslocamento relativo entre os campos que compõem cada quadro. Em razão dos campos estarem entrelaçados, há um deslocamento de uma linha entre eles, ou seja, o campo par está exatamente uma linha deslocado para cima ou para baixo em referência ao campo ímpar. Se os campos forem simplesmente

desentrelaçados e exibidos, haverá um desagradável efeito de sobe-desce (*Bob*) na exibição do vídeo. É para compensar este efeito que se utiliza a filtragem *Bob*. Normalmente também se chama *Bob* todo o processo de desentrelaçamento e correção do efeito sobe-desce da imagem. Em sua versão mais simples há redução da resolução do vídeo à metade e duplicação da taxa de exibição de quadros. Quando associada a outras técnicas como a *weave*, o desentrelaçamento *Bob* aplicado de forma inteligente pode mostrar-se uma das melhores soluções do ponto de vista perceptual para realizar a tarefa de desentrelaçamento[5], [12], [13];

Apesar da eficiência perceptual de algumas das técnicas descritas acima, em todas elas o vídeo processado sempre perde no quesito nitidez de detalhes, característica fundamental para as análises periciais, além de em muitos perder também resolução. Tendo em vista o foco investigativo dos trabalhos de análise, no qual o que interessa é a resolução estática dos campos, optamos por realizar um desentrelaçamento que consistisse simplesmente na técnica chamada desdobramento (*unfold*) [5], [12], [13]. Nela, cada um dos quadros entrelaçados é desentrelaçado (separação dos campos) em dois com resolução vertical reduzida à metade (vide exemplo na Fig. 3.1). Como resultado deste processo aplicado ao arquivo capturado, obteve-se um arquivo de vídeo composto pelo dobro do número de quadros (1.755.940), cada quadro com resoluções de 640x240 pixels.

Realizada a remoção do trecho ruidoso e o desentrelaçamento do vídeo, realizou-se em seguida a recodificação do arquivo agora no formato Indeo5. Encontrava-se concluída a fase de pré-processamento, estando pronto o arquivo para o processamento das imagens.

5.3 Processamento

Como já foi dito, o processamento foi realizado com o auxílio da ferramenta MATLAB, no qual foi desenvolvido um script para demultiplexar o vídeo. A premissa básica era a de que o programa deveria seguir resumidamente o seguinte fluxograma básico:

1. *abrir o arquivo de vídeo;*
2. *ler quadro a quadro o arquivo de vídeo;*

3. *obter informações globais do arquivo como número de quadros, tipo de codec utilizado, taxa de exibição de quadros, espaço de cor, etc;*
4. *para cada um dos quadros:*
 - a. *extrair do mesmo alguma característica que o identificasse;*
 - b. *realizar a comparação dessa característica com as de outros quadros já classificados;*
 - c. *decidir se essa característica é razoavelmente próxima àquelas dos quadros já classificados (se existirem):*
 - i. *caso negativo, criar um novo arquivo para armazenar quadros dessa nova câmera;*
 - ii. *caso positivo, decidir no arquivo de que câmera depositar o quadro analisado;*
5. *fechar o arquivo original;*
6. *fechar os arquivos das câmeras;*

Dessa forma, o *script* deveria ser capaz de, sem saber previamente o número de câmeras que compunha o sistema de CFTV, detectar esse número, criando um arquivo para cada uma das câmeras detectadas. Esses arquivos, verdadeiros escaninhos que deveriam conter apenas imagens de uma determinada câmera, seriam, então, passíveis de ser analisados por um ser humano que localizaria o trecho onde teria ocorrido a ação criminosa.

Em cada um dos passos do desenvolvimento do programa tivemos que adaptá-lo às particularidades do caso em análise. Passamos a descrever os procedimentos realizados no desenvolvimento do programa.

A abertura de arquivos de imagem no programa MATLAB [14] é feita carregando suas informações em uma estrutura (*struct*) chamada *image* composta pelas propriedades: dados da cor (*color data*) – *cdata*; e mapa de cores (*color map*) - *colormap*.

A representação de imagens no MATLAB pode ser feita de duas formas gerais:

- **Imagens referenciadas ou indexadas:** são aquelas representadas por uma matriz bidimensional (*cdata*) cujos elementos são índices de outra matriz bidimensional – o mapa de cores (*colormap*). Cada matriz armazena informações distintas: *cdata*

armazena a localização espacial do pixel na imagem e um variável que representam um índice para uma linha (trio) na matriz *colormap*; *colormap* por sua vez é um repositório ordenado de cores pré-definidas descritas em uma matriz $N \times 3$, onde N é o número de linhas (cores) do mapa e 3 refere-se ao número de componentes fundamentais do espaço de cor (ex.: RGB). Cada elemento da matriz *colormap* é identificado por um índice e já apresenta um trio RGB pré-definido composto por elementos do tipo *double* (16bits) cujos valores variam no intervalo $[0, 1]$. A leitura de imagens deste tipo é sempre realizada em dois passos: primeiro deverá ser realizada a leitura da matriz *cdata* que indica a localização espacial do pixel e o índice da cor; a seguir, com o valor do índice, é realizada a busca do mesmo na matriz *colormap* que descreverá os percentuais de cada uma das cores elementares (R, G e B por exemplo) que o pixel apresenta. A Figura 5.2, inspirada em exemplo similar ao apresentado pela referência [11], ilustra o processo de referenciamento indireto de cores utilizado mapa de cores;

- o Uma variante simplificada das imagens referenciadas são as imagens de intensidade, trata-se de matrizes de pixels (*cdata*) que representam um *dégradê* entre duas cores. Para esse tipo de imagem, um mapa de cores não seria necessário, uma vez que o próprio elemento da matriz *cdata* poderia encerrar a valor de intensidade. Apesar disso, o MATLAB utiliza para as imagens intensidade uma matriz de dados (*cdata*) cujos elementos a exemplo das imagens referenciadas apontam para elementos do mapa de cor (*colormap*). Este faz as vezes do *dégradé* entre duas cores, trata-se de um vetor coluna cujos elementos são os valores de intensidade na escala. Em regra essa escala varia do branco para o preto – escala de cinza (*gray scale*), mas há também a possibilidade de opção por uma escala entre duas cores quaisquer;

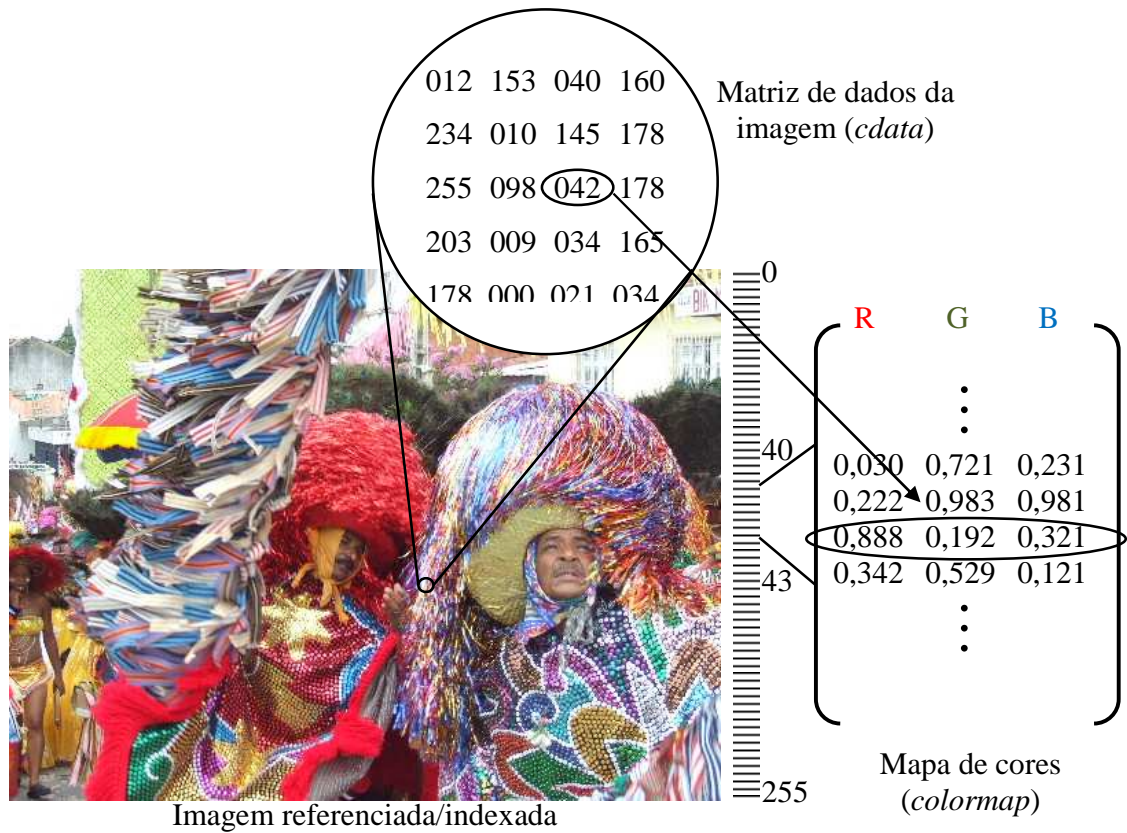


Figura 5.4 – Processo de referenciamento indireto de cores utilizado em imagens referenciadas/indexadas e de intensidade

- **Imagens RGB:** são representada por três matrizes bidimensionais, uma para cada componente fundamental do espaço de cor, mais comumente: R, G e B. Neste caso, a propriedade mapa de cor (*colormap*) não é levado em consideração uma vez que não há necessidade de outras referências para a definição das cores já que as intensidades das fundamentais já se encontram definidas em *cdata*. Os elementos das matrizes de imagens RGB podem ser variáveis do tipo *double* (16 bits) variando no intervalo [0,1], *unsigned char* (8bits – 0-255) ou *unsigned integer* (16bits – 0-65.535);

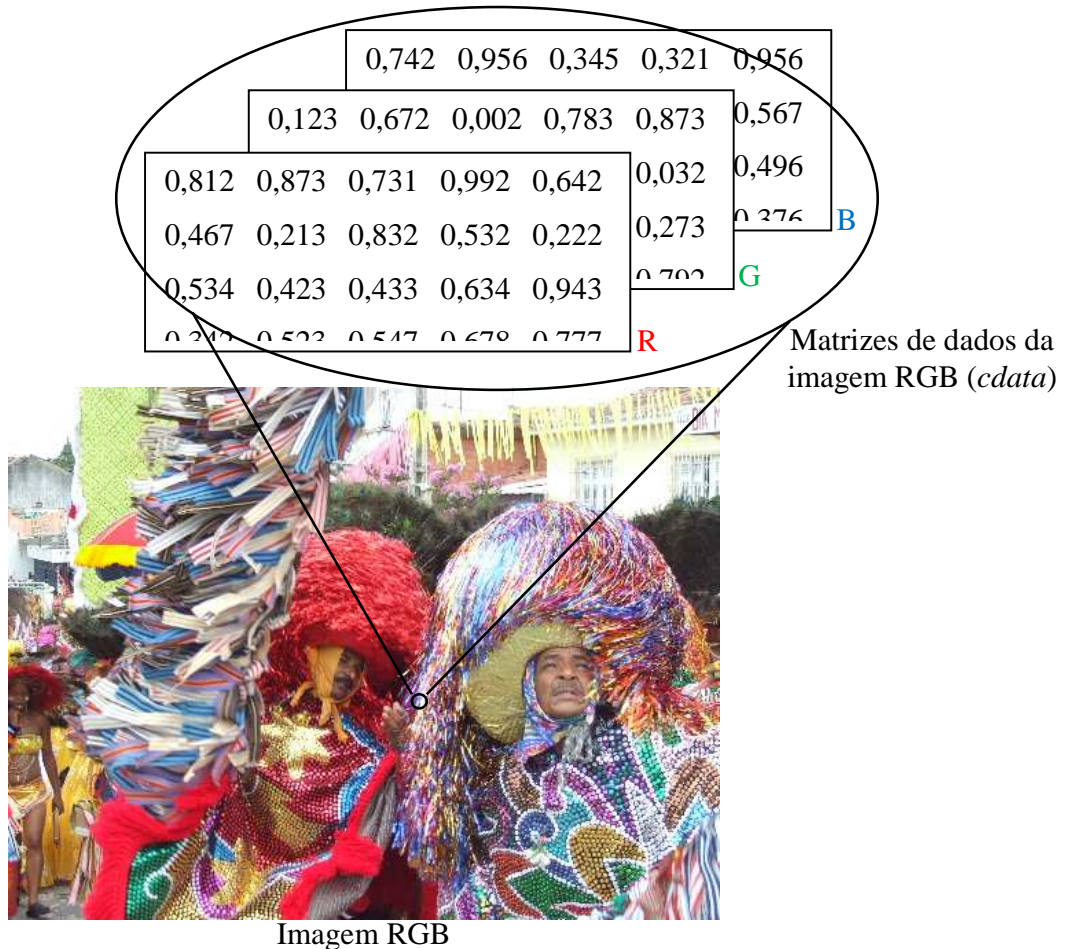


Figura 5.5 – Representação em três matrizes de uma imagem RGB

Em resumo, podemos concluir que a propriedade dados da cor (*color data*) ou *cdata* é a mais importante, uma vez que recebe as matrizes dos quadros que compõem o vídeo. A propriedade *colormap* por sua vez mapeia as cores em um vetor composto por suas três componentes fundamentais RGB. Dessa forma, cada cor C é definida por um vetor do tipo:

$$\text{mapa}(C) = [R(C), G(C), B(C)]$$

onde $R(C)$, $G(C)$ e $B(C)$ representam as intensidades de R, G e B para a cor C .

Em virtude das limitações do programa de captura, o vídeo contido na fita VHS foi capturado gerando um arquivo de vídeo colorido (RGB) *truecolor* e não em escala de cinza como se desejava. Para não haver perda de detalhes no vídeo, optamos por não alterar as imagens para escala de cinza, trabalhando apenas com uma das matrizes RGB. No processamento do caso originário utilizou-se no processamento a matriz R já que o vídeo é em escala de cinza e verificou-se que as diferenças entre elas (matrizes R, G e B) eram pequenas. Outra informação

relevante é que os elementos da matriz eram representados por uma escala de tons de cinza representada por variáveis do MATLAB do tipo *uint8* que em linguagem de programação usual traduz-se para *unsigned char* (8 bits sem sinal) assumindo valores na faixa de 0 a 255.

Dessa forma, trabalhando no MATLAB todo o tempo com imagens RGB, não exploramos a propriedade *colormap*, que quando listada aparecia, como esperado como uma matriz vazia. Todo o processamento então baseou-se na propriedades *cdata*. A notação utilizada no programa foi a seguinte:

```
vídeo(i).cdata(:,1)
```

onde *i* representa o número do frame na seqüência do vídeo e *(:,:1)* representa todas as linhas e colunas da matriz R (vermelho).

Tendo em vista o grande tamanho do arquivo de entrada (1.755.940 quadros), chegou-se à conclusão que seria impossível realizar a leitura do arquivo de uma só vez, já que as matrizes são carregadas em memória RAM. Assim, conciliando a alocação de memória RAM e a velocidade de processamento, a leitura foi sendo realizada em “rodadas” de 300 quadros, à exceção da última “rodada” cujo tamanho é adequado aos quadros remanescentes da divisão do número total de frames do arquivo pelo tamanho da janela, permitindo a leitura de todos os quadros.

Para cada um dos quadros é calculado um parâmetro de comparação. O parâmetro escolhido foi a luminância média dos quadros escolhido pelo fato de que o critério não deveria ser nem muito estrito, a ponto de identificar leves mudanças no cenário e de uma mesma câmera como uma nova câmera, tampouco genérico demais a ponto de classificar todos os quadros do vídeo multiplexado como provenientes de uma única câmera. Nesta linha de intelecção, um parâmetro suficientemente genérico seria a luminância média da cena. Para testar a eficiência do critério, verificando se o mesmo era viável, criamos alguns arquivos de teste com 300 quadros (1 rodada). Esses arquivos tiveram como origem vídeos reais de sistemas CFTV multiplexados que foram previamente submetidos a processo idêntico de pré-processamento (desentrelaçados e excluídos trechos com algum tipo de falha - trechos contaminados com estática, falhas de multiplexação, etc.). Com base nesses arquivos foram plotados gráficos bidimensionais ilustrando o comportamento da luminância média ao longo do vídeo da seguinte forma: seqüência de frames *versus* suas respectivas luminâncias médias. A seguir exibimos um gráfico (Fig.5.4) obtido de um dos arquivos de teste do vídeo analisado:

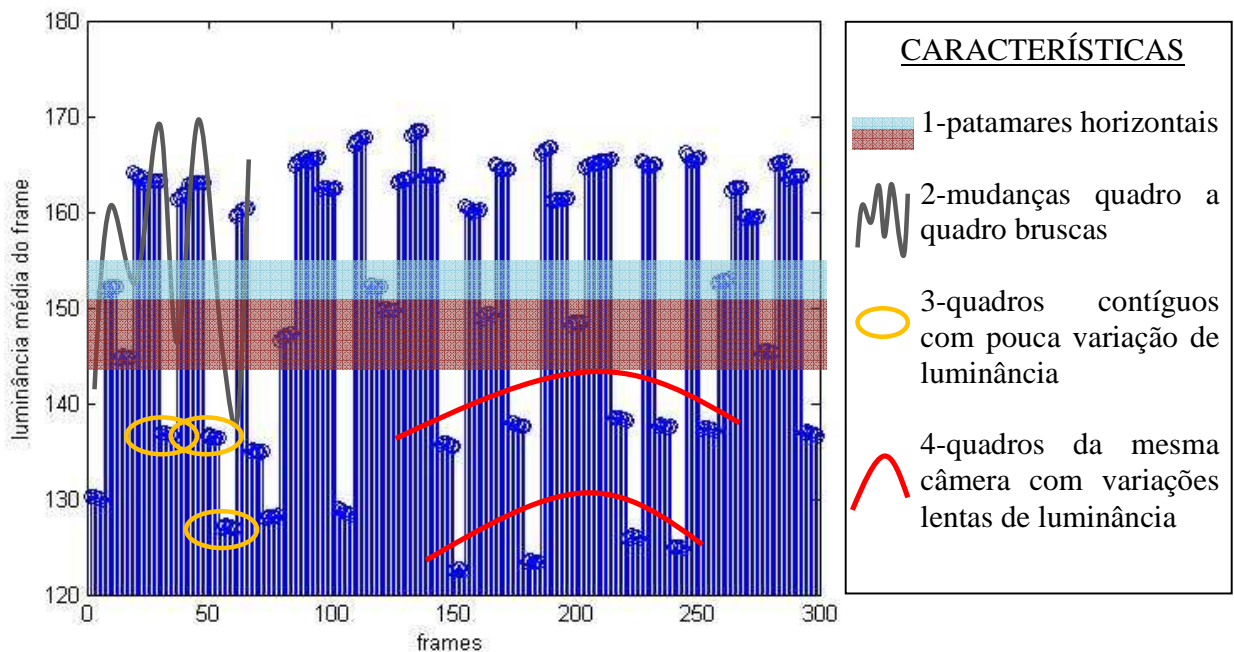


Figura 5.6 – Gráfico da luminância média dos quadros de um arquivo de teste com 300 quadros

Uma análise cuidadosa das características do gráfico acima nos permite tirar algumas conclusões interessantes:

1. há coerência nos picos, ou seja, existem patamares horizontais razoavelmente definidos dentro de uma faixa;
2. as mudanças do último quadro de uma câmera para o primeiro da câmera seguinte são bruscas, o que facilita a detecção das mudanças de câmera;
3. os quadros contíguos da mesma câmera apresentam pouca variação de luminância e número aproximadamente igual, o que facilita a classificação dos quadros seguintes ao primeiro da nova câmera, agilizando o algoritmo;
4. as variações de luminância de quadros da mesma câmera não contíguos existe mas é lenta, o que permite que o algoritmo se adapte às variações de luminância intra-câmera;

5.3.1 Os critérios de classificação dos quadros em análise

O processo de cálculo da luminância média apresenta algumas vantagens como o fato de empregar basicamente operações de soma e poucas multiplicações/divisões, reduzindo a complexidade computacional do algoritmo.

Apesar de fazermos menção ao método simplesmente falando de luminâncias médias, na verdade, a implementação não se resume à mera comparação direta das luminâncias médias dos quadros, mas sim ao cálculo das matrizes diferença pixel-a-pixel entre o quadro atual sob análise e os “quadros referência” de cada uma das câmeras já criadas. Depois de se calcular essa matriz diferença, a mesma é reduzida a um valor único, normalizado, que representa a luminância média do “quadro-diferença” para determinada câmera. A partir desse valor e de limiares pré-estabelecidos pelo usuário, realiza-se o processo de decisão quanto ao pertencimento do quadro em análise a uma das câmeras já criadas.

Para exemplificarmos o processo, suponhamos que o algoritmo já tenha identificado três (03) câmeras e estejamos analisando um quadro recém-lido do vetor de leitura. Para sabermos a que câmera pertence esse quadro, devemos buscar na estrutura da câmera 01 o seu “quadro-referência”, realiza-se a diferença pixel a pixel entre o quadro analisado e o quadro-referência da câmera 01. A matriz diferença é então resumida a um valor normalizado que é comparado com o limiar de decisão escolhido pelo usuário. Se o resumo da diferença for menor que o limiar, a princípio, o quadro analisado pertence à câmera em questão, caso contrário o algoritmo repete as operações até que as N câmeras existentes tenham sido testadas.

A idéia de utilizar-se limiares apóia-se no intuito de agilizar o algoritmo, uma vez encontrada uma câmera cujo resumo da diferença de luminosidades médias é considerado pequeno o suficiente, ocorre a classificação e o algoritmo passa a analisar o quadro seguinte da seqüência do vídeo sob análise. Desta forma, não há a necessidade de se testar a diferença para todas as câmeras. Inicialmente, nos primeiros algoritmos empregados na pesquisa, realizava-se o cálculo para todas as câmeras, no entanto, o critério de parada baseado no limiar mostrou-se bastante robusto, apresentando resultados tão satisfatórios quanto o seu antecessor que acabou sendo abandonado por ser mais lento.

5.3.2 Os critérios de criação dos novos repositórios de quadros

Talvez um dos maiores desafios do algoritmo seja o de definir o critério para a criação de novos repositórios de quadros. A idéia inicial de se criar um algoritmo de classificação de imagens, inicialmente, não abrangia a pretenciosa tarefa de automatizar a detecção do número de câmeras presentes no sistema gerador das imagens. No entanto, com o progresso dos trabalhos e em virtude da observação de características presentes nos gráficos das luminâncias médias dos frames (vide Fig.5.4), percebemos que a tarefa também poderia ser realizada de forma semi-automatizada ou mesmo automatizada.

Nos primeiros algoritmos, as câmeras nada mais eram que arquivos de extensão avi, que ficavam abertos recebendo, incremental e seqüencialmente, os quadros a elas atribuídos durante o processo de classificação. No entanto, percebeu-se que as câmeras deveriam encerrar outras informações, sendo entidades mais complexas que meros escaninhos de frames. Percebeu-se que outras características estavam individualmente a elas associadas e que deviam ser armazenadas em uma estrutura. Na versão atual do programa, cada uma das câmeras é composta por uma estrutura nas quais armazenam-se:

1. permanentemente: o chamado “quadro-referência” de cada câmera. Esse elemento é, talvez, o mais importante da câmera pois é dele, na realidade, que denota sua identidade e é atualizado a cada novo quadro classificado como pertencente à câmera em questão;
2. temporariamente: os quadros considerados a ela pertencentes pelo algoritmo durante uma “rodada”, até que esta termine e esses quadros sejam escritos no final do arquivo correspondente à cada uma das câmeras. Gravados no arquivo, são apagados desse vetor temporário alocado na estrutura;
3. temporariamente: o contador de quadros associados à câmera em determinada rodada, serve para indexar operações a quadros de determinada câmera. Após a escrita dos quadros nos respectivos arquivos é reiniciado por uma questão de economia de memória volátil.

Uma informação bastante pertinente que deve ser armazenada de forma permanente é a posição do quadro classificado no vídeo original. Não elencada na lista acima, a posição originária dos frames classificados fazia parte da estrutura da câmera em versões anteriores do programa. Ao fim do processo de classificação, era gerado um *log* para cada uma das câmeras no qual eram sucessivamente listados os quadros a ela pertencentes.

Esta informação mostrou-se relevante no caso de vídeos indevidamente classificados, nos quais, corrige-se o erro durante uma segunda rodada do algoritmo no arquivo onde foi detectado o erro tornando o processo de classificação mais rigoroso. Como resultado o processo apura o arquivo que continha erro dividindo-o em dois: um contendo apenas os quadros errados e outro com o arquivo original agora livre dos quadros errados. Assim, a informação de localização original dos quadros permite identificar exatamente onde ocorreu o erro e concatenar ou mesmo enxertar trechos *a posteriori* recriando as seqüências de cada uma das câmeras.

Na versão mais recente do algoritmo, durante a fase de pré-processamento, inclui-se na própria imagem, em uma região de silêncio (normalmente na parte inferior), o número e o instante de cada quadro. Com isto, torna-se instantâneo saber qual sua posição na seqüência e o exato instante em que cada quadro classificado é exibido dispensando a necessidade de referências indiretas como era o caso do arquivo de *log*.

Exposta a amplitude do elemento câmera no contexto do algoritmo, passamos a discorrer sobre os critérios para sua criação.

Durante o processo de desenvolvimento do algoritmo verificou-se que os resultados obtidos nas primeiras rodadas são bastante representativos no que se refere à eficiência na classificação. Um início ruim dificilmente leva a uma identificação precisa do número de câmeras e a uma correta classificação dos quadros. A razão dessa sensibilidade inicial está relacionada ao processo de criação dos frames-padrão que analisaremos mais adiante. O que precisamos ter em mente é que qualquer erro introduzido no momento de criação do frame-padrão é propagado por todo o processamento.

Durante a primeira “rodada”, ou seja, quando são tratados os 300 primeiros quadros do vídeo, há um grande esforço computacional no que se refere à criação dos novos repositórios de quadros. Esse esforço vai progressivamente diminuindo com as rodadas seguintes sendo substituído pela simples classificação dos quadros nas câmeras existentes. Isto se deve a duas características dos sistemas de vídeo multiplexado:

1. precisam contemplar todas as câmeras que compõem o sistema;
2. precisam minimizar o espaço de armazenamento das imagens;

A solução empregada pelos multiplexadores é amostrar poucos quadros de cada uma das câmeras. O critério de escolha do número de quadros tratados a cada rodada levou em consideração alguns fatores:

- Em primeiro lugar o espaço em memória RAM ocupado pelos quadros que precisam ser nela carregados para o processamento: Muitos frames requerem muito espaço de memória RAM;
- Número de acessos ao disco rígido: Verificou-se que uma das fases mais lentas do algoritmo são os acessos a disco, sobretudo para escrita. Se os frames são tratados em grande número por vez, reduz-se o número de acessos ao disco rígido, agilizando o processo. No entanto, isso requer mais espaço de memória RAM;
- É desejável que logo na primeira rodada seja possível formar um quadro padrão realmente representativo do cenário filmado, para tanto, são necessários, na versão atual do programa, 10 quadros de cada uma das câmeras.

Após essa análise verificou-se empiricamente que, para os vídeos analisados, uma janela de 300 quadros se ajustaria bem ao processamento. Para que se tenha idéia de quão representativo são 300 frames estáticos em termos de vídeo dinâmico, sistemas de vídeo padrão capturam imagens à taxa de 29,97fps (NTSC/PAL M colorido). A essa taxa, um intervalo de 300 quadros representaria cerca de 10 segundos da movimentação dos ambientes monitorados. Durante análises na fita do caso em questão e em outras da mesma natureza referentes a casos similares, observou-se que na primeira rodada, ou seja, nos primeiros 10 segundos dos vídeos analisados, quase a totalidade das câmeras são criadas. No intuito de melhor ilustrar o processo de criação de repositórios de quadros em análise sem, contudo, interromper o encadeamento de idéias, fazemos neste momento referência ao anexo II. Composto por uma tabela, o Anexo II sintetiza como se dá o processo de criação de repositórios de quadros para um arquivo de vídeo composto pelos 300 primeiros quadros do vídeo analisado. Vale lembrar que no processo de classificação, o parâmetro utilizado é a diferença de luminância.

Excetuam-se à regra da criação dos repositórios de quadros na primeira rodada os casos de mudanças muito bruscas de iluminação que acabam por ser detectadas como novas câmeras. Outro caso muito comum de erro de classificação observado foi o de fitas que apresentam falhas de alinhamento das cabeças em seus trechos iniciais. Entendemos, no entanto, que o resultado desse erro (dois vídeos com conteúdo da mesma câmera ao invés de apenas um) produzido por falhas dessa natureza não é grave do ponto de vista prático, pois os vídeos podem ser facilmente

fundidos *a posteriori* com o auxílio dos caracteres do número do quadro e instantes de tempo impressos nos quadros na fase de pré-processamento.

No que refere à implementação do processo de criação dos repositórios de quadros no algoritmo, em sua versão mais recente, este só é disparado quando, durante o processo de classificação, para nenhuma das câmeras o resumo da diferença de luminosidades médias obtido foi menor do que o limiar de semelhança estabelecido pelo usuário. Apesar de aparentemente simples, antes ainda da criação de um novo repositório de quadros, faz-se necessário observar alguns critérios que analisaremos a seguir.

Tendo em vista a natural existência de variações de luminosidade nos vídeos e sua característica de apresentarem, logo em seu trecho inicial, amostras dos quadros de todas as câmeras, foi inserido no código um fator que oferece resistência à criação de novos repositórios de câmeras. Esse fator está associado à localização temporal do quadro em análise no vídeo analisado. A idéia básica é criar uma espécie de inércia temporal ao processo de criação de repositórios de quadros da seguinte forma:

- maior facilidade de criação no início do vídeo quando a probabilidade de aparecimento de novas câmeras é muito grande pois são apresentados quadros das diferentes câmeras;
- aumento progressivo da dificuldade de criação dos repositórios de quadros com o desenrolar do vídeo justificado pela baixa probabilidade do aparecimento de câmeras novas e maior probabilidade de tratarem-se de quadros de câmeras já existentes com variações instantâneas de luminância acima da média.

Com esse fator, nas primeiras rodadas a dificuldade para criar câmeras é baixa pois é nelas que de fato elas devem surgir. Conforme vão sendo completadas as rodadas, a dificuldade deve se tornar maior pois depois de algumas rodadas, somente variações muito grandes de luminosidade possibilitam a criação de novos repositórios de quadros. Essa técnica associada às demais do algoritmo, conseguiu-se uma considerável redução, nos casos analisados, a detecção equivocada de câmeras, sobretudo em trechos avançados do vídeo. Problemas relacionados a limiares de detecção de câmeras muito baixos são identificados logo nas primeiras 3 rodadas com a presente técnica, permitindo que o usuário encerre o processo, ajuste seu valor e recomece o processamento sem perda de tempo.

Além da dificuldade progressiva imposta à criação de novos repositórios de quadros, também foi estudado um critério para se escolher o “quadro referência” de cada câmera. Um dos pontos criticados nos artigos da empresa Cognitech [16] e [17] da abordagem de utilizar-se a luminância média da cena para realizar a classificação dos quadros é justamente que as suas variações findam por inviabilizar sua utilização. Analisando *a priori*, de fato foi o que observamos. Eventos como a passagem de veículos de grande porte próximo à janelas voltadas para a rua, ou transeuntes vestindo roupas escuras nos ambientes monitorados de fato causavam variação na luminância média da cena. No entanto, conceitualmente espera-se que o “quadro referência” de uma câmera seja imune, ou quase, a esse tipo de evento transiente, devendo, contudo, ser capaz de adaptar-se a variações mais lentas como a própria variação de luminosidade do transcorrer do dia. Nesse sentido, imaginamos que o “quadro referência” deveria ser um quadro contendo apenas baixas frequências do vídeo representando a região filmada por determinada câmera, excluindo-se dele eventos transientes como variações rápidas de luz ou a passagem de transeuntes.

Para que obtivéssemos o resultado descrito, implementamos um filtro passa-baixas nos quadros como forma de obter uma média ponderada dos últimos quadros classificados como pertencentes a determinada câmera. O filtro deveria apresentar um ponto de corte não baixo demais a ponto de não se capaz de se incorporar variações desejáveis, adaptando-se apenas à variações muito lentas da luminosidade da cena, e não tão alto a ponto de ser sensível a mudanças transientes correndo o risco de que quadros classificados erradamente sejam capazes de retirar o “quadro referência” da região da qual ainda será capaz de corrigir-se. Desta forma, o que se buscou foi atualizar o “quadro referência” com uma contribuição representativa de cada quadro recém classificado como pertencente à respectiva câmera.

Percebeu-se que a técnica empregada apresenta duas vantagens principais:

1. Elementos espúrios presentes em apenas um ou poucos quadros tendem a sofrer um processo progressivo de esmaecimento, não comprometendo a generalidade da referência;
2. Quadros classificados erradamente não comprometem a câmera, visto que sua contribuição não é suficiente para arrastar o valor do resumo da diferença de luminosidades médias para uma região da qual não seja possível voltar. Em

virtude disso, mesmo classificando erradamente algum quadro, o algoritmo vai diminuindo a influência do quadro errado até que a mesma seja eliminada.

O filtro passa-baixas de quadros, precisa ser cuidadosamente projetado pois se mal ajustado pode atrapalhar o processo. O número de quadros utilizados para o cálculo da média-móvel vai depender de diversos fatores do vídeo como: dinamismo da cena, circulação de pessoas, outros eventos que possam ser considerados ruído.

A seguir, para ilustrar o efeito produzido pelo processo de média ponderada de quadros para a obtenção dos “quadros referência”, foi composto o painel da Figura 5.7. Pode-se observar na sua parte superior nove quadros pertencentes a uma das câmeras de um vídeo de CFTV bancário multiplexado e abaixo, ampliado, o respectivo “quadro referência” da câmera em referência.



Figura 5.7 – Painel composto por quadros pertencentes a uma câmera (parte superior) e respectivo quadro-referência ampliado (parte inferior)

5.3.3 Comportamento do processamento

O processamento das imagens foi realizado em um computador pessoal de arquitetura IBM PC com a seguinte configuração:

Configuração do computador utilizado	
Processador	Intel Core 2 Duo 6300, <i>clock</i> 1.86GHz
Placa mãe	Intel DG965SS – <i>chipset</i> Intel G965
Memória RAM	Kingston KVR667D2N5/1G – 1GB
Disco Rígido	Samsung HD160JJ – 160GB/7200RPM/8MB <i>cache</i>

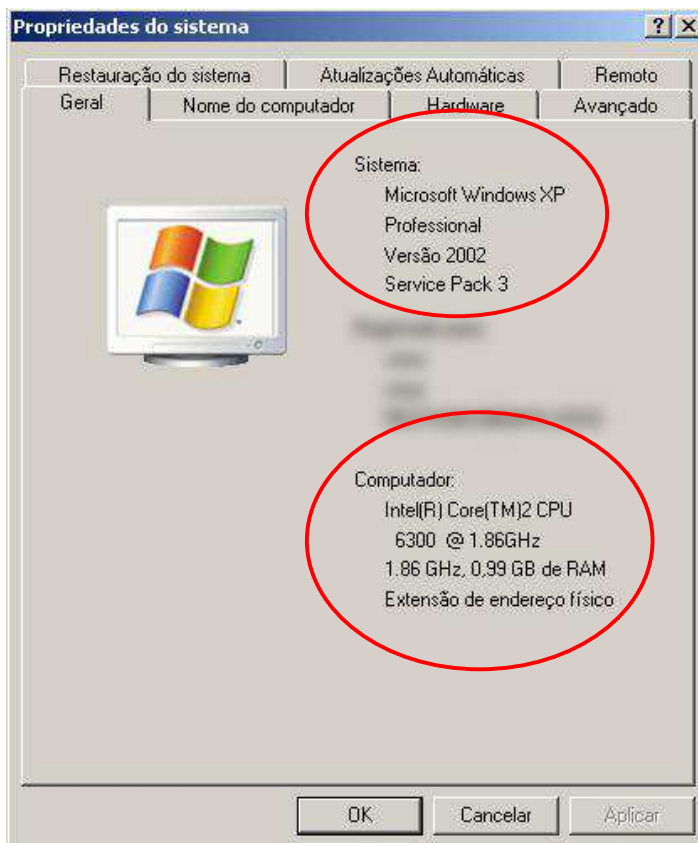


Figura 5.8 – Tela da janela de propriedades do sistema operacional do computador utilizado no processamento das imagens

Durante o processamento do algoritmo, foi monitorado o comportamento dos processadores, do acesso à memória RAM e os acessos a disco com o auxílio do programa Process Explorer de autoria de Mark Russinovich [29]. Para ilustrar o processamento, juntamos a seguir uma tela do referido programa ilustrando o comportamento do computador (Fig.5.9).

No que se refere à velocidade de processamento dos arquivos de vídeo, verificou-se os seguintes resultados utilizando o computador descrito:

- O pré-processamento realizado pelos programas AVISYNTH e VIRTUALDUB composto pelos processos de: desentrelaçamento dos quadros; inserção de caracteres de número e instante dos quadros; e codificação Indeo 5.2 (IV50) apresentou taxa média de processamento de 10fps;
- Quanto ao processamento propriamente dito das imagens, realizado pelo MATLAB 7 este apresentou taxa média.

Maiores detalhes sobre o processamento podem ser encontrados no Capítulo 6 no qual são apresentados individualmente informações sobre outros casos de vídeos multiplexados classificados com a técnica descrita.

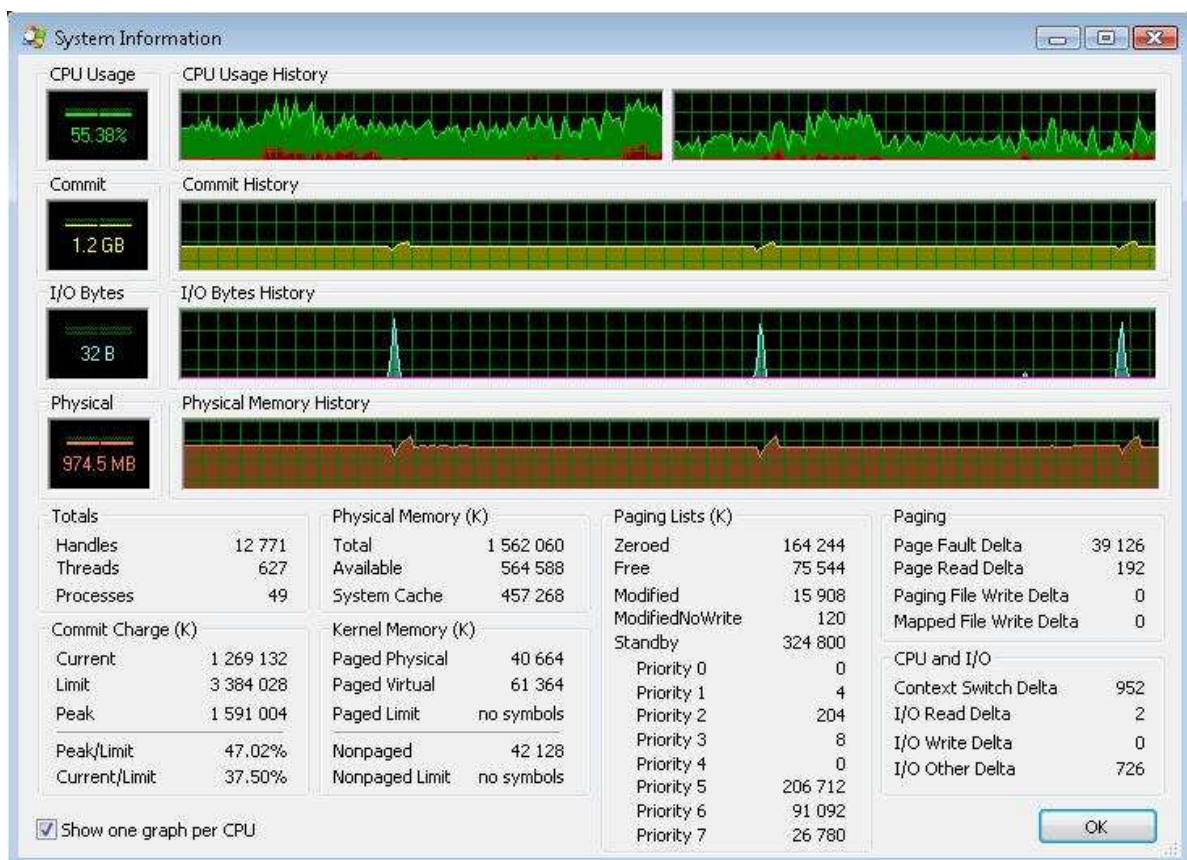


Figura 5.9 – Tela da janela do programa Process Explorer que exibe o comportamento do computador durante o processamento de um vídeo multiplexado

Capítulo 6

Resultados Obtidos

Neste capítulo mostramos os resultados obtidos com a utilização do algoritmo desenvolvido empregando as técnicas descritas. Além da conclusão do caso da “pescaria”, oportunamente trazemos uma bateria de testes que foram realizados com sucesso utilizando o programa desenvolvido. As Tabelas 6.1 a 6.4 demonstram quão eficiente o algoritmo pode ser quando utilizado em casos semelhantes, trazendo seus pontos fortes e fracos. Aproveitamos ainda para expor algumas vantagens de ordem não técnicas da utilização do algoritmo, inclusive disponibilizando o código do mesmo na íntegra (Apêndice I) no intuito de que este venha a ajudar a solucionar outros casos.

6.1 Resultados Obtidos

O maior resultado obtido foi, sem dúvida, o desenvolvimento de uma ferramenta que vem permitindo ao Setor Técnico-Científico da Polícia Federal tratar vídeos que antes não eram analisados em razão da indisponibilidade dos demultiplexadores. De fato seria impossível que a perícia dispusesse de ao menos um demultiplexador de cada marca e modelo em produção além daqueles já descontinuados. Soma-se a isso a inviabilidade de manutenção desses equipamentos. Outro ponto que merece destaque é o fato de que essa solução veio sem nenhum custo de aquisição de programas computacionais ou equipamentos, já que é implementada em software e exige equipamentos que os laboratórios de análises em audiovisual já dispõem.

Do ponto de vista imediato, o desenvolvimento do programa permitiu a solução do caso descrito. Apesar de os réus já se encontrarem presos respondendo a processo, as imagens reveladas eram prova incontestável de suas condutas delituosas e neutralizaram qualquer tentativa de defesa que sustentasse que os réus eram inocentes, seja alegando que os mesmos nada tinham feito, seja invocando o instituto do crime impossível. Uma estatística interessante sobre o trabalho é que dos 1.755.940 de quadros contidos na fita, digitalizados, pré-processados e analisados pelo programa, apenas 6 comprovam a atitude ilícita do casal.

O painel mostrado na Figura 6.1, composto por imagens do trecho de interesse encontrado através do processo de demultiplexação descrito, ilustra a ação dos réus ao tentar “pescar” os malotes. As imagens foram editadas aplicando-se um filtro de embaçamento (*blur*) nos rostos dos filmados para preservar suas identidades.

6.2 Testes com o algoritmo

À época da solução do caso, o programa ainda se encontrava sem algumas melhorias que foram implementadas depois e que agilizaram o tratamento das imagens. Para que se tenha uma idéia, o processamento, quando da descoberta dos quadros incriminadores, durava cerca de quatro dias. Várias foram as tentativas frustradas até que se chegasse ao resultado final o caso que desencadeou este trabalho. Com base nos resultados positivos a pesquisa no tema foi impulsionada, resultando em uma versão do programa mais abrangente e eficiente que vem possibilitando a sua utilização para a solução de outros casos similares.

Com o intuito de ilustrar concretamente o emprego do algoritmo em trabalhos realizados em imagens da mesma natureza que aquela do caso em tela, inclusive casos solucionados pelo autor com essa ferramenta, sintetizamos em tabelas informações sobre os seus processamentos. As tabelas encerram informações sobre características originais do vídeo, passos realizados no pré-processamento, processamento, além de dados sobre a eficiência do algoritmo. Para essa última informação, a qual se buscou dar maior ênfase, foram capturadas imagens representativas de cada uma das câmeras classificadas para que se possa avaliar seus conteúdos e a capacidade de discernimento do algoritmo. Nos casos em que houve falha na classificação, buscou-se identificar o seu motivo, descrevendo a análise sobre o erro.






Figura 6.1 – Painel com os quadros da ação dos réus junto à caixa de malotes.
Estão destacados os 6 quadros da ação criminoso

Tabela 6.1 – Arquivo vhs1.avi

Arquivos	vhs1.avi
Quadros	12.500
Resolução	640x480
Fps	29,97
Duração	0h21m03,66s
Taxa de dados	29442 kbps
Espaço de cor	TRUECOLOR (24bits)
Qualidade imagens	Quadros com baixa nitidez, regulares contraste e brilho
Pré-processamento	Desentrelaçamento (Avisynth)
	Inserção de caracteres de número e instante nos quadros (Avisynth)
	Codificação Ligos Indeo XP Vídeo 5.2 – IV50 (VirtualDub)
Duração pré-processamento	1h00m41s Taxa de pré-processamento de quadros: 10fps
Câmeras de fato	15
Câmeras detectadas	15
Duração processamento	1h 35m42s Taxa de processamento de quadros: 4 fps
Erros de classificação	0
Motivos dos erros	

<p>Quadro original</p>	
<p>Cam1</p> <ul style="list-style-type: none"> • 1.562 quadros classificados corretamente; • 0 classificados errados; 	
<p>Cam2</p> <ul style="list-style-type: none"> • 1.563 quadros classificados corretamente; • 0 classificados errados; 	
<p>Cam3</p> <ul style="list-style-type: none"> • 1.563 quadros classificados corretamente; • 0 classificados errados; 	

<p>Cam4</p> <ul style="list-style-type: none">• 1.562 quadros classificados corretamente;• 0 classificados errados;	 <p>01764 00:00:29.412</p>
<p>Cam5</p> <ul style="list-style-type: none">• 3.125 quadros classificados corretamente;• 0 classificados errados;	 <p>02443 00:00:40.740</p> <p>02453 00:00:40.907</p>
<p>Cam6</p> <ul style="list-style-type: none">• 1.563 quadros classificados corretamente;• 0 classificados errados;	 <p>02902 00:00:48.398</p>
<p>Cam7</p> <ul style="list-style-type: none">• 1.563 quadros classificados corretamente;• 0 classificados errados;	 <p>03175 00:00:52.952</p>

<p>Cam8</p> <ul style="list-style-type: none"> • 1.563 quadros classificados corretamente; • 0 classificados errados; 	
<p>Cam9</p> <ul style="list-style-type: none"> • 1.563 quadros classificados corretamente; • 0 classificados errados; 	
<p>Cam10</p> <ul style="list-style-type: none"> • 1.563 quadros classificados corretamente; • 0 classificados errados; 	
<p>Cam11</p> <ul style="list-style-type: none"> • 1.562 quadros classificados corretamente; • 0 classificados errados; 	
<p>Cam12</p> <ul style="list-style-type: none"> • 1.562 quadros classificados corretamente; • 0 classificados errados; 	

<p>Cam13</p> <ul style="list-style-type: none">• 1.562 quadros classificados corretamente;• 0 classificados errados;	
<p>Cam14</p> <ul style="list-style-type: none">• 1.562 quadros classificados corretamente;• 0 classificados errados;	
<p>Cam15</p> <ul style="list-style-type: none">• 1.562 quadros classificados corretamente;• 0 classificados errados;	

Tabela 6.2 – Arquivo anzol_teste.avi

Arquivos	anzol_teste.avi
Quadros	4.500
Resolução	640x480
Fps	29,97
Duração	0h02m30,15s
Taxa de dados	11.764 kbps
Espaço de cor	Y8 (256 tons de cinza)
Qualidade imagens	Quadros com boa nitidez, contraste e brilho
Pré-processamento	Exclusão de trecho com estática e falta de sincronismo (VirtualDub)
	Desentrelaçamento com duplicação de campo (VirtualDub)
	Inserção de caracteres de número e instante nos quadros (Avisynth)
	Codificação Ligos Indeo XP Vídeo 5.1 – IV50 (VirtualDub)
Duração pré-processamento	Taxa de pré-processamento de quadros: 10fps
Câmeras de fato	8
Câmeras detectadas	8
Duração processamento	0h29min45seg Taxa de processamento de quadros: 3 fps
Erros de classificação	0
Motivos dos erros	
Quadro original	

Cam1

- 610 quadros classificados corretamente;
- 0 classificados errados;

**Cam2**

- 466 quadros classificados corretamente;
- 0 classificados errados;



Cam3

- 673 quadros classificados corretamente;
- 0 classificados errados;

**Cam4**

- 571 quadros classificados corretamente;
- 0 classificados errados;



Cam5

- 448 quadros classificados corretamente;
- 0 classificados errados;

**Cam6**

- 503 quadros classificados corretamente;
- 0 classificados errados



Cam7

- 685 quadros classificados corretamente;
- 0 classificados errados;



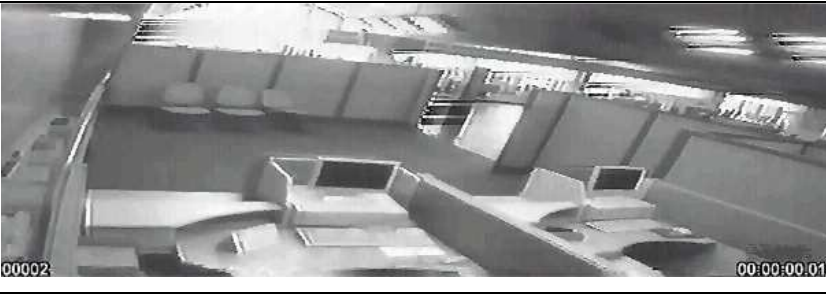

**Cam8**






- 544 quadros classificados corretamente;
- 0 classificados errados;








Tabela 6.3 – Arquivo entrelaçado rápido.avi

Arquivos	entrelaçado rápido.avi
Quadros	56.832
Resolução	612x200
Fps	59,94
Duração	0h15m48,14s
Taxa de dados	12.646 kbps
Espaço de cor	Y8 (256 tons de cinza)
Qualidade imagens	Quadros com regular nitidez, regular contraste e regular brilho
Pré-processamento	Exclusão de trecho com estática e falta de sincronismo (VirtualDub)
	Exclusão de bordas superiores, inferiores e laterais
	Desentrelaçamento (Avisynth)
	Inserção de caracteres de número e instante nos quadros (Avisynth)
	Codificação Ligos Indeo XP Vídeo 5.2 – IV50 (VirtualDub)
Duração pré-processamento	Taxa de pré-processamento de quadros: 10fps
Câmeras de fato	15 (2 câmeras desligadas)
Câmeras detectadas	14 (2 câmeras desligadas detectadas como uma só)
Duração processamento	2h36m01s Taxa de processamento de quadros: 6 fps
Erros de classificação	0
Motivos dos erros	

Quadros original	
Cam1 <ul style="list-style-type: none">• 7.567 quadros classificados corretamente;• 0 classificados errados;	
Cam2 <ul style="list-style-type: none">• 3.783 quadros classificados corretamente;• 0 classificados errados;	
Cam3 <ul style="list-style-type: none">• 3.781 quadros classificados corretamente;• 0 classificados errados;	

<p>Cam4</p> <ul style="list-style-type: none">• 3.788 quadros classificados corretamente;• 0 classificados errados;	
<p>Cam5</p> <ul style="list-style-type: none">• 3.797 quadros classificados corretamente;• 0 classificados errados;	
<p>Cam6</p> <ul style="list-style-type: none">• 3.795 quadros classificados corretamente;• 0 classificados errados;	
<p>Cam7</p> <ul style="list-style-type: none">• 3.793 quadros classificados corretamente;• 0 classificados errados;	
<p>Cam8</p> <ul style="list-style-type: none">• 3.793 quadros classificados corretamente;• 0 classificados errados;	

<p>Cam9</p> <ul style="list-style-type: none">• 3.792 quadros classificados corretamente;• 0 classificados errados;	 <p>00011 00:00:00:166</p>
<p>Cam10</p> <ul style="list-style-type: none">• 3.791 quadros classificados corretamente;• 0 classificados errados;	 <p>00012 00:00:00:183</p>
<p>Cam11</p> <ul style="list-style-type: none">• 3.788 quadros classificados corretamente;• 0 classificados errados;	 <p>00013 00:00:00:200</p>
<p>Cam12</p> <ul style="list-style-type: none">• 3.788 quadros classificados corretamente;• 0 quadros classificados errados;	 <p>00014 00:00:00:216</p>
<p>Cam13</p> <ul style="list-style-type: none">• 3.793 quadros classificados corretamente;• 0 classificados errados;	 <p>00015 00:00:00:233</p>

Cam14



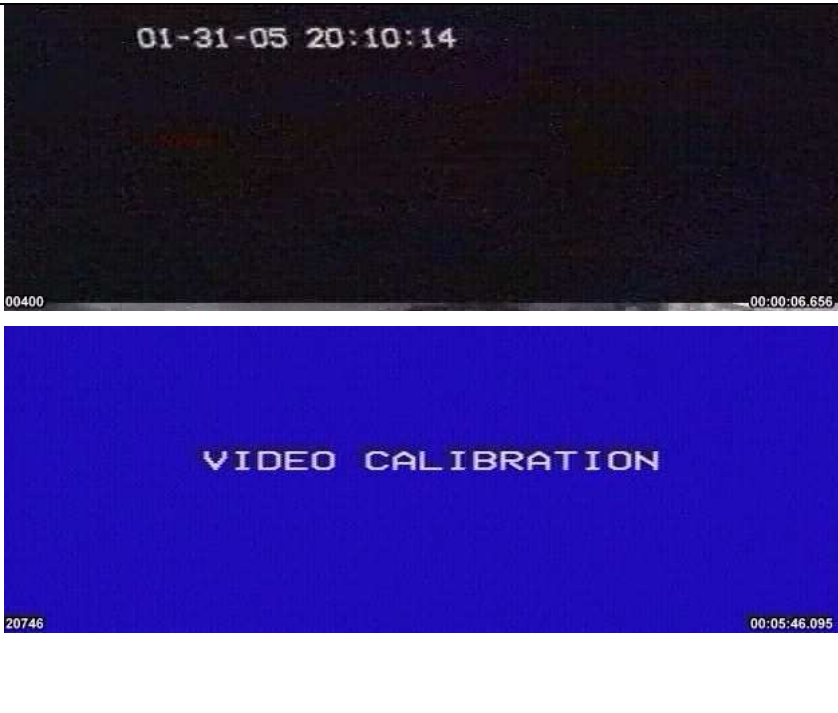
- 3.783 quadros classificados corretamente;
- 0 classificados errados;



Tabela 6.4 – Arquivo Of_1220_08.avi

Arquivos	Of_1220_08.avi
Quadros	14.583
Resolução	640x480
Fps	29,97
Duração	0h08m06,58s
Taxa de dados	23.997 kbps
Espaço de cor	TRUECOLOR (24bits)
Qualidade imagens	Quadros coloridos com boa nitidez, bom contraste e bom brilho
Pré-processamento	Exclusão de trecho com estática e falta de sincronismo (VirtualDub)
	Exclusão da região de retraço com informações do multiplexador (Avisynth)
	Exclusão de bordas superiores, inferiores e laterais
	Desentrelaçamento (Avisynth)
	Inserção de caracteres de número e instante nos quadros (Avisynth)
	Codificação Ligos Indeo XP Vídeo 5.2 – IV50 (VirtualDub)
Duração pré-processamento	0h45m24s Taxa de pré-processamento de quadros: 11 <i>fps</i>
Câmeras de fato	11
Câmeras detectadas	28
Duração processamento	1h45m41s Taxa de processamento de quadros: 5 <i>fps</i>
Erros de classificação	
Motivos dos erros	Perda de sincronia das cabeças dos videocassetes / mau funcionamento do equipamento multiplexador – estática, contaminação de campos que apresentavam imagens de duas câmeras / limitações do algoritmo




<p>Amostra de quadro original</p>	
<p>Cam1 (correta)</p> <ul style="list-style-type: none"> • 2.399 quadros classificados corretamente; • 0 classificados errados; • Obs: adaptou-se ao desligamento das luzes 	
<p>Cam2 (correta)</p> <ul style="list-style-type: none"> • 1.378 quadros classificados corretamente; • 0 classificados errados; • Obs: adaptou-se ao desligamento das luzes e saída dos funcionários; 	

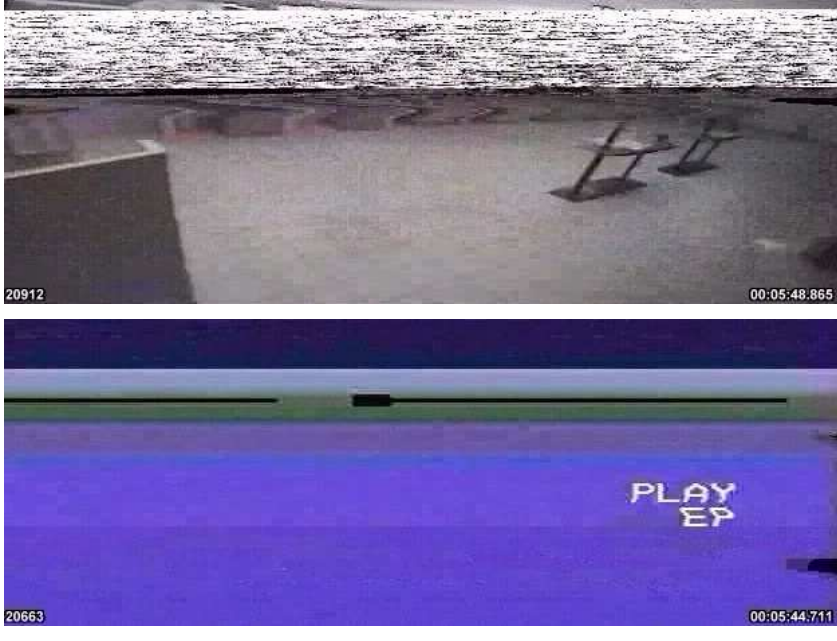
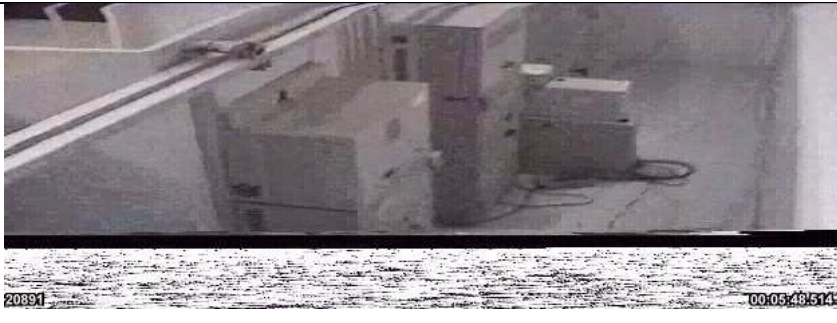


	
<p>Cam3 (correta)</p> <ul style="list-style-type: none"> • 4.798 quadros classificados corretamente; • 0 classificados errados; • Obs: adaptou-se ao acendimento das luzes; 	
<p>Cam4 (correta)</p> <ul style="list-style-type: none"> • 4.741 quadros classificados corretamente; • 41 classificados errados próximos a trechos de perda de sincronismo. <p>Causas: 1) Quadros de baixa luminância contaminados de estática; 2) Quadros de baixíssima luminância;</p> <ul style="list-style-type: none"> • Obs 1: câmera desligada; • Obs 2: classificou a tela azul de perda de sincronismo como 	

<p>pertencente à câmara 4;</p>	 <p>Exemplo de quadro classificado errado – problema associado ao limiar de luminância</p> <p>Exemplo de quadro classificado errado – problema associado ao limiar de luminância e contaminação por estática (parte inferior da imagem)</p>
<p>Cam5</p> <ul style="list-style-type: none"> • 2.379 quadros classificados corretamente; • 0 classificados errados; 	
<p>Cam6</p> <ul style="list-style-type: none"> • 2.404 quadros classificados corretamente; • 0 classificados errados • Obs: adaptou-se ao desligamento das luzes; 	

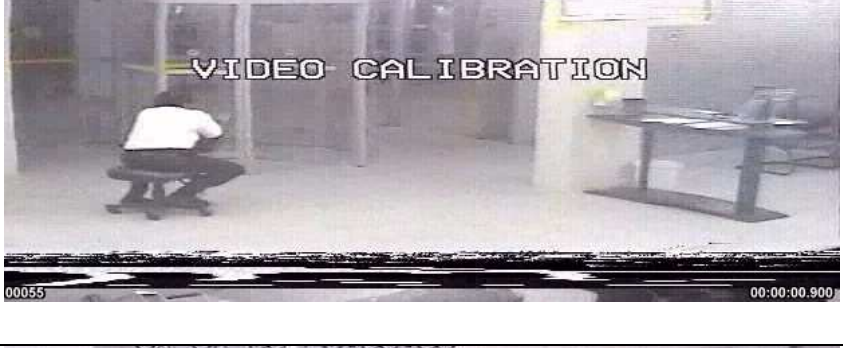



	
<p>Cam7</p> <ul style="list-style-type: none"> • 2.379 quadros classificados corretamente; • 0 classificados errados; 	
<p>Cam8</p> <ul style="list-style-type: none"> • 1.318 quadros classificados corretamente; • 1 classificado errados. Causa: quadro com perda de sincronismo, cromaticidade e luminância alterados; • Obs: adaptou-se a duas aberturas da porta do cofre; 	   <p>Exemplo de quadro classificado errado – problema associado a</p>

	perda de sincronismo da imagem
<p>Cam9</p> <ul style="list-style-type: none"> • 2.380 quadros classificados corretamente; • 0 classificados errados; 	
<p>Cam10</p> <ul style="list-style-type: none"> • 3.446 quadros classificados*; • *Quadros de duas câmeras classificados como pertencentes a uma só a partir do instante 04:36.893. Causa: ao apagar das luzes duas câmeras passaram a apresentar imagens muito semelhantes; 	 <p>Exemplo de quadro classificado errado – problema associado a grande semelhança entre as imagens</p>

<p>Cam11</p> <ul style="list-style-type: none"> • 1.363 quadros classificados corretamente; • 0 classificados errados; 	
<p>Cam12</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 2 quadros classificados errados. Causa: contaminação por estática na parte inferior. Conteúdo que deveria ter sido classificado na Cam11; 	
<p>Cam13</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 3 classificados errados. Causa1: 2 dos quadros por limitação do algoritmo – luminância média fora do limiar. Causa 2: o 3º quadro por perda de sincronismo; • Conteúdo deveria ter sido classificado nas Cam1 e 2; 	

<p>Cam14</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 8 classificados errados. Causa: contaminação por estática e perda de sincronismo; 	
<p>Cam15</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 24 classificados errados. Causa: contaminação por estática; 	
<p>Cam16</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 9 classificados errados. Causa: contaminação por estática; 	
<p>Cam17</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 1 classificado errados. Causa: limitação do algoritmo – luminância média fora do limiar; 	

<p>Cam18</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 1 classificado errados. Causa: contaminação por estática e limitação do algoritmo – luminância média fora do limiar; 	
<p>Cam19</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 1 classificados errados. Causa: perda de sincronismo e contaminação por estática; 	
<p>Cam20</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 1 classificados errados. Causa: perda de sincronismo e contaminação por estática; 	
<p>Cam21</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 6 classificados errados. Causa: perda de sincronismo e contaminação por estática; 	

<p>Cam22</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 3 classificados errados. Causa: perda de sincronismo e contaminação por estática; 	
<p>Cam23</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 3 classificados errados. Causa: perda de sincronismo; 	
<p>Cam24</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 1 classificados errados. Causa: perda de sincronismo; 	
<p>Cam25</p> <ul style="list-style-type: none"> • 0 quadros classificados corretamente; • 1 classificados errados. Causa: perda de sincronismo; 	

Capítulo 7

Conclusões e Trabalhos Futuros

Com base no vídeo apresentado e utilizando as técnicas descritas, o processo de demultiplexação com base na luminância média dos quadros mostrou-se adequado. Combinado com outras técnicas, como os quadros referência baseados em média móvel, o algoritmo demonstrou ser capaz de se adaptar a grande parte das variações normais como a passagem de clientes vestindo roupas das mais diversas cores e a variação da iluminação natural presentes em vídeos de vigilância. Em muitos casos, mostrou-se capaz também de se adaptar a variações incomuns como a abertura de portas e o desligamento da iluminação. Várias imagens do processamento do vídeo do caso da “pescaria” e outros similares também resolvidos com a técnica apresentada foram apresentados no Capítulo 6. Por fim, tendo em vista a frequente apresentação de vídeos multiplexados à perícia para análise, acreditamos que muito ainda poderia ser feito para tornar o algoritmo ainda mais eficiente e sobretudo acelerar o processamento dos quadros

Buscaram-se por outras soluções para que pudéssemos realizar comparações de eficiência com o algoritmo desenvolvido. Verificou-se que não há muitas referências sobre o assunto e que pesquisas sobre o mesmo nos levam, invariavelmente, a patentes ou *softwares* comerciais voltados para o problema. Um deles, o *Video Investigator* da empresa Cognitech, chegou a ser analisado pelo Instituto Nacional de Criminalística, que à época dos testes optou por não o adquirir.

7.1 Sugestões para trabalhos futuros

Como o intuito de dar continuidade ao trabalho que acreditamos ainda pode ser muito refinado, expandido a capacidade de adaptação do algoritmo, sugerimos alguns pontos que podem servir de sugestão para outros trabalhos.

Talvez, uma das maiores contribuições que possa ser dada ao algoritmo seria automatizar completamente a demultiplexação do vídeo. Atualmente, alguns parâmetros são ajustados manualmente, sobretudo o limiar de semelhança utilizado na comparação dos quadros em análise com os quadros-referência das câmeras. Apesar de ter um valor único e, ao que se percebeu

durante a utilização do programa para outros vídeos, a variação necessária para o perfeito funcionamento do programa para outros vídeos é pequena. Baseando-nos em outras implementações semelhantes existentes na literatura [16], [17] e [18], talvez um caminho a seguir no sentido da automatização dessa variável seria realizar uma análise estatística dos valores de luminância de uma amostra representativa de quadros do vídeo multiplexado antes de processar todo o vídeo. Com base nos gráficos analisados de luminância média dos quadros observamos haver concentração de luminância em determinadas faixas. A primeira tarefa seria identificar os valores médios e desvios padrões para a luminância de cada uma dessas faixas. Em seguida, com base na distância relativa entre essas faixas, seria possível identificar qual o valor ótimo para o limiar de semelhança que apresentasse uma folga segura para realizar a classificação. Utilizando esse valor ótimo a probabilidade de se realizar uma classificação errada estaria minimizada embora não descartada.

Uma outra possibilidade seria utilizar um limiar de semelhança variável ou estudar alguma outra característica mais eficiente e que exigisse menor esforço computacional para ser calculado no intuito de identificar os quadros.

Um refinamento muito desejável também seria agregar ao *script* uma interface gráfica amigável e compilar o programa em código C ou outra linguagem de programação que agilize a execução do programa.

Referências

- [1] ALVARES, MARCELO FELIPE MAIA HOR-MEYLL, **Informação Técnica nº 010/2005 do Setor Técnico Científico da Superintendência Regional da Polícia Federal em Pernambuco (SETEC/SR/DPF/PE)**, Pernambuco: Recife, 2005;
- [2] ALVARES, MARCELO FELIPE MAIA HOR-MEYLL, **Laudo Pericial nº 980/2006 do Setor Técnico Científico da Superintendência Regional da Polícia Federal em Pernambuco (SETEC/SR/DPF/PE)**, Pernambuco: Recife, 2006;
- [3] ALVARES, MARCELO FELIPE MAIA HOR-MEYLL, **Laudo Pericial nº 1.236/2006 do Setor Técnico Científico da Superintendência Regional da Polícia Federal em Pernambuco (SETEC/SR/DPF/PE)**, Pernambuco: Recife, 2006;
- [4] GROB, BERNARD, **Basic Television and Video Systems**, 2ª edição, EUA: New York, McGraw-Hill Education, 1999;
- [5] sítio Internet sobre processamento de vídeo: <http://www.100fps.com/>, último acesso em 10/03/2009;
- [6] RUDIN, LEONID I.; YU, Ping; MOREL, Jean-Michel, **Sampling theory for digital video acquisition: the guide for the perplexed user**, *SPIE (The International Society for Optical Engineering)*. Proc. SPIE Vol. 4709, p. 4-15, *Investigative Video Processing*, 07/2002;
- [7] POYNTON, CHARLES A., **Digital video and HDTV algorithms and interfaces**, 1ª edição, EUA: San Francisco, Morgan Kaufmann, 2005;
- [8] ITU BT.601-6, **Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios**, última revisão em 01/2007, situação: em vigência, <http://www.itu.int/rec/R-REC-BT.601/en> ;
- [9] ITU BT 709-5, **Parameter values for the HDTV standards for production and international programme exchange**, última revisão em 04/2002, situação: em vigência, <http://www.itu.int/rec/R-REC-BT.709/en> ;
- [10] ITU-R BT 470-7, **Conventional analog television systems**, última revisão em 02/2005, situação: em vigência, <http://www.itu.int/rec/R-REC-BT.470/en> ;
- [11] ITU BT.656-5, **Interfaces for digital component video signals in 525-line and 625-line television systems operating at the 4:2:2 level of recommendation ITU-R BT.601 (part A)** , última revisão em 12/2007, situação: em vigência, <http://www.itu.int/rec/R-REC-BT.656/en> ;
- [12] AVREY, LEE, **sítio AviSynth** disponível em <http://www.avisynth.org> ;
- [13] AVREY, LEE, **sítio VirtualDub**, disponível em <http://www.virtualdub.org> ;

- [14] MATLAB Central, **fórum de usuários do programa**, disponível em <http://www.mathworks.com/matlabcentral/> ;
- [15] WIKIPEDIA, **S-Vídeo**, disponível em <http://pt.wikipedia.org/wiki/S-VIDEO> ;
- [16] LISANI, J.; RUDIN, LEONID I., MONASSE, P.; MOREL, Jean-Michel; YU, Ping; *Meaningful Automatic Video Demultiplexing with Unknown Number of Cameras, Contrast Changes, and Motion*, EUA: Cognitech Inc., Pasadena, CA, **Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference**, 15/09/2005, <http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/10535/33340/01577337.pdf?arnumber=1577337> ;
- [17] GUICHARD, Frederic; LITZ, Alexander; RUDIN, LEONID I.; YU, Ping; *Software-based universal demultiplexing: threshold-free energy minimization approach*, SPIE (The International Society for Optical Engineering). Proc. SPIE Vol. 4232, p. 513-520, *Enabling Technologies for Law Enforcement and Security*, 02/2001, <http://adsabs.harvard.edu/abs/2001SPIE.4232..513G> ;
- [18] COLMENAREZ, Antonio J.; GUTTA, Srinivas, V., R.; TRAJKOVIC, Miroslav; *Video monitoring and surveillance systems capable of handling asynchronously multiplexed video*, United States Patent 6,999,613, 14/02/2006, Maracaibo, Venezuela, <http://www.wipo.int/pctdb/en/wo.jsp?wo=2003056527> ;
- [19] SENCORE, *Understanding The IVPP Composite Video Signal*, Sencore Tech Tip #148, Sioux Fall, South Dakota, EUA;
- [20] sítios Internet do fabricante do programa MATLAB, Mathworks de suporte aos usuários descrevendo o problema: <http://www.mathworks.com/support/solutions/data/1-1ZD67J.html?solution=1-1ZD67J>; http://www.mathworks.de/matlabcentral/newsreader/view_thread/159467 último acesso em 10/03/2009;
- [21] GOODRIDGE, Steven G., *Content based software demultiplexing of surveillance video*, Proc. SPIE Vol. 4232, p. 489-496, 05/11/2000, <http://adsabs.harvard.edu/abs/2001SPIE.4232..489G> ;
- [22] sítio Internet de banco de informações de circuitos integrados: <http://www.datasheetarchive.com/Bt848-datasheet.html> último acesso em 11/03/2009;
- [23] BRAMBLE, Simon K.; CARAPEZZA, Edward M.; RUDIN, Leonid I., *Enabling Technologies for Law Enforcement and Security*, Proc. SPIE Vol. 423, 05/11/2000, <http://adsabs.harvard.edu/abs/2001SPIE.4232.....B> ;
- [24] sítio Internet da empresa Ligos Corporation: <http://www.ligos.com/index.php/home/products/indeo/> último acesso em 10/03/2009;
- [25] sítio Internet: <http://www.digitalpreservation.gov/formats/fdd/fdd000065.shtml> http://ligos.com/pdf_docs/Indeo_doc.pdf último acesso em 10/03/2009;

- [26] sítio Internet: <http://patft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnethtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=4698675.PN.&OS=PN/4698675&RS=PN/4698675> último acesso em 10/03/2009;
- [27] sítio Internet: http://www.avchduser.com/articles/watching_interlaced_video.jsp último acesso em 10/03/2009;
- [28] sítio Internet: <http://www.microsoft.com/brasil/technet/sysinternals/utilities/processexplorer.msp>, último acesso em 13/03/2009;
- [29] ROSIN, P. *Thresholding for change detection*, *Computer Vision and Image Understanding*, vol. 86, no. 2, pp. 79-95, 2002, <http://users.cs.cf.ac.uk/Paul.Rosin/thresh/thresh.html> ;
- [30] sítio Internet sobre programa computacional comercial para demultiplexação de vídeo http://www.oceansystems.com/dtective/dplex/dplex_index.html ;
- [31] BOVICK, Al; **Handbook of Image and Video Processing**, 1^a edição, EUA, Texas, Academic Press, 2000;
- [32] THYAAGARAJAN, K. S.; **Digital Image Processing with Application to Digital Cinema**, 1^a edição, EUA, San Diego, Focal Press, 2006;

APÊNDICE I – ALGORITMO DE DEMULTIPLEXAÇÃO DE VÍDEO

```

%%% Programa para demultiplexação de frames
%%%
%%%
%%% PCF Marcelo Hor-Meyll
%%%
%%% SETEC/SR/DPF/PE
%%%
%%% 29/09/2006 - Le arquivo AVI em formato RGB
%%%              - Cria o número de câmeras dinamicamente de acordo com o
%%%              conteúdo do vídeo
%%%              - Classifica frames por camera se a diferença entre o frame
%%%              padrão e o atual for menor do que um valor fixo (6)
%%%
%%% 03/10/2006 - Le arquivo AVI em pacotes de tamanho windowSize
%%%              - Ajusta o tamanho da última janela de acordo com o tamanho do
%%%              arquivo
%%%              - Cria arquivos de saída para cada camera
%%%              - Cada câmera é uma estrutura com:
%%%                número de frames
%%%                frame default
%%%                nome do arquivo de saída
%%%                posição dos frames da camera no arquivo original
%%%              - Antes de cada rodada o programa mostra o intervalo de
%%%              frames no qual está trabalhando no formato:
%%%                início do intervalo
%%%                fim do intervalo
%%%                total de frames do arquivo
%%%              Resolver problema de estouro de memoria
%%%              Solucao esvaziar todas as matrizes depois de escrever os
%%%              dados do windowSize para liberar memoria
%%%
%%% 15/11/2006 - Pre-processamento é essencial e deve:
%%%              * remover frames com ruído do início do vídeo
%%%              * desentrelacar os frames para evitar a criação de
%%%              cameras contendo os frames de transição
%%%
%%% 16/11/2006 - Novas regras para a criação de cameras
%%%              * se erro < 6 classifica em uma camera no loop do
%%%              programa
%%%              * se passou por todas as cameras e não classificou
%%%              verifica se erro < 8, se for classifica em uma
%%%              das cameras
%%%              * se ainda não classificou verifica se o frame
%%%              ainda está no início do vídeo (<150frames) e se a
%%%              diferença é pequena (erro<15) ou se o frame fica
%%%              no meio do vídeo daí classifica em uma delas
%%%              * se ainda não classificou e está no início do
%%%              vídeo daí cria nova camera
%%%
%%% 18/04/2008 - Criação de parâmetros de medidas dos erros
%%%
%%%
%%%

```



```

clc;
clear all;
j=1;
i=1;
%frameCount=[];
tempErroEntr = [];
ncam=0;
defaultOutputPath = 'C:\teste_cam\';
inputFile = 'entrelacado_rapido_desentrelacado_avisynth_indeo32.avi';
Files\MATLAB\R2006\work\4500_frames_bons_desentrelacados.avi';
windowSize = 300;
offset=0;
%0.299*R+0.5870*G+0.1140*B
%colormap(gray);
grayScaleVideo = 1;
videoInfo=aviinfo(inputFile); % lê informações do arquivo
disp(['Percentual concluído: ', int2str(100*offset/videoInfo.NumFrames)]);

%numRodadas=videoInfo.NumFrames;

tic;

%for (offSet=1>windowSize:videoInfo.NumFrames)
while(offset<videoInfo.NumFrames) % loop para todas as janelas do vídeo
    if((videoInfo.NumFrames/(offset>windowSize))<1) % verifica se a janela nao
vai além do numero total de frames do arquivo
        windowSize=rem(videoInfo.NumFrames>windowSize); % ajusta o tamanho da
última/única janela para concluir o total de frames
    end
    clear entrada;
    %pack;
    entrada=aviread(inputFile,[offset+1:(offset>windowSize)]); % lê o arquivo
de entrada em janelas e carrega na structure entrada

    %cdata(m,n,3) - m linhas, n colunas, 3 RGB. 3 matrizes m x n pixels
    %cada cor com 24 bits de profundidade

    tamanho_rgb =
((size(entrada(1).cdata,1))*(size(entrada(1).cdata,2))*(size(entrada(1).cdata,
3))); % pega o frame 1 e multiplica m x n x 3
    tamanho_gray = ((size(entrada(1).cdata,1))*(size(entrada(1).cdata,2))); %
pega o frame 1 e multiplica m x n x 1
    [numLinhas, numColunas,numDim] = size(entrada(1).cdata); % carrega a
largura, altura e RGB nas variáveis

    clc
    disp(['processando do frame ', offset, ' a ', offset>windowSize, ' de
', videoInfo.NumFrames]);
    %display('processando de')
    %offset
    %display('a')
    %offset>windowSize
    %display('total de frames')
    %videoInfo.NumFrames

```

```

disp ([int2str(100*offset/videoInfo.NumFrames) , '% concluido: ']);

for (i=1>windowSize) %loop para classificação de todos os frames da janela
    currentFrameIndex = offset + i;
    classificou = 0;
    erro = 255; % cria o vetor de erros de tamanho ncam, inicializa com
erro máximo

    %% rock %%matriz_erro(offset+i,1)= offset+i; %%

    for k=1:ncam %loop para calculo do erro do frame i em relação ao
frame padrão de cada camera
        if (classificou == 0)
            if grayScaleVideo % se for escala de cinza
                %% funciona p arqu laudo %%cam(k).tempMatrizErro =
abs(cam(k).df_frame.cdata(:, :,1)- entrada(i).cdata(:, :,1)); % calcula matriz
diferença entre o frame atual e os padrões de todas as câmeras
                %%cam(k).tempMatrizErro = abs(int16(dummy2(:, :,1))-
int16(dummy1(:, :,1))); % calcula matriz diferença entre o frame atual e os
padrões de todas as câmeras
                cam(k).tempMatrizErro =
abs(int16(cam(k).df_frame.cdata(:, :,1))- int16(entrada(i).cdata(:, :,1))); %
calcula matriz diferença entre o frame atual e os padrões de todas as câmeras
                %erro(k)=sum(sum(cam(k).df_frame.cdata(:, :,1)-
entrada(i).cdata(:, :,1)))/tamanho_gray; %erro = média da luminancia do frame
das diferencas entre o frame i e o frame padrao da camera k
                erro(k)=sum(sum(cam(k).tempMatrizErro))/tamanho_gray;
%erro = média da luminancia do frame das diferencas entre o frame i e o frame
padrao da camera k, soma linhas depois colunas divide pelo número de pixels do
frame, reduz a um número o frame
                %%erro(k)=abs((sum(sum(cam(k).df_frame.cdata(:, :,1))-
sum(sum(entrada(i).cdata(:, :,1)))))/tamanho_gray);
                %% rock %%lum_media(offset+i)=
sum(sum(entrada(i).cdata(:, :,1)))/tamanho_gray;
            else % se for colorido
                erro(k)=sum(sum(sum(cam(k).df_frame.cdata-
entrada(i).cdata)))/tamanho_rgb; %erro = média da luminancia do frame das
diferencas entre o frame i e o frame padrao da camera k
            end

            %% rock %%matriz_erro(offset+i,k+2)=erro(k); %%

            if(erro(k)<6) % se o erro(k,i) for pequeno o frame i pertence
à camera k
                %frameCount(k)=frameCount(k)+1; % incrementa contador de
frames da camera k
                cam(k).numFrames=cam(k).numFrames+1;
                %cam(k).frames(frameCount(k))=entrada(i); % adiciona o
frame i à camera k
                cam(k).frames(cam(k).numFrames)=entrada(i); % adiciona o
frame i à camera k
                cam(k).df_frame.cdata = 0.9*cam(k).df_frame.cdata +
0.1*(entrada(i).cdata); % atualiza o frame padrao da camera k
                cam(k).matrizErro = cam(k).matrizErro +
abs(double(cam(k).tempMatrizErro)/255);
                %cam(k).frameOldPosition(frameCount(k))=i+offset; %
armazena a posição original do frame classificado

```

```

        classificou = 1;

        %%% rock
%%erro_df(i)=sum(sum(cam(k).tempMatrizErro)/tamanho_gray);

        %%% rock %%%matriz_erro(offset+i,2)=k; %%%

        end % fim if(erro(k,i)<3)
    end % fim of (classificou == 0) TIREI PRA CALCULAR TODOS OS ERROS
18/04/08
    end % fim for k=1:ncam

    if classificou == 0
        if min(erro) < 28 % CLASSIFICA NA CÂMERA MAIS PARECIDA se não
classificou ainda, mas depois de ler todas as cameras o erro é menor do que 8
        [minValue,indexMin] = min (erro); % escolhe o menor erro
(min(erro(k))) e a respectiva câmera (k)
        %frameCount(indexMin)=frameCount(indexMin)+1; % incrementa
contador de frames da camera k
        cam(indexMin).numFrames=cam(indexMin).numFrames+1;
        %cam(indexMin).frames(frameCount(indexMin))=entrada(i); %
adiciona o frame i à camera k
        cam(indexMin).frames(cam(indexMin).numFrames)=entrada(i); %
adiciona o frame i à camera k
        cam(indexMin).df_frame.cdata =
0.9*cam(indexMin).df_frame.cdata + 0.1*entrada(i).cdata; % atualiza o frame
padrao da camera k com peso de 10% para a nova câmera
        cam(indexMin).matrizErro = cam(indexMin).matrizErro +
abs(double(cam(indexMin).tempMatrizErro)/255);
        %cam(k).frameOldPosition(frameCount(k))=i+offset; % armazena a
posição original do frame classificado
        classificou = 1;

        %%% rock
%%erro_df(i)=sum(sum(cam(indexMin).tempMatrizErro)/tamanho_gray);

        %%% rock %%%matriz_erro(offset+i,2)=indexMin; %%%

    else
        if (currentFrameIndex > 150 && min(erro) < 14) ||
(currentFrameIndex > 300) % CLASSIFICA NA CÂMERA MAIS PARECIDA se ainda está
no início e o erro é médio ou se não está mais no início do vídeo
        [minValue,indexMin] = min (erro); % escolhe o menor erro
(min(erro(k))) e a respectiva câmera (k)
        %frameCount(indexMin)=frameCount(indexMin)+1; % incrementa
contador de frames da camera k
        cam(indexMin).numFrames=cam(indexMin).numFrames+1;
        %cam(indexMin).frames(frameCount(indexMin))=entrada(i); %
adiciona o frame i à camera k
        cam(indexMin).frames(cam(indexMin).numFrames)=entrada(i);
% adiciona o frame i à camera k
        cam(indexMin).df_frame.cdata =
0.9*cam(indexMin).df_frame.cdata + 0.1*entrada(i).cdata; % atualiza o frame
padrao da camera k com peso de 10% para a nova câmera
        cam(indexMin).matrizErro = cam(indexMin).matrizErro +
abs(double(cam(indexMin).tempMatrizErro)/255);

```

```

        %cam(k).frameOldPosition(frameCount(k))=i+offset; %
armazena a posição original do frame classificado
        classificou = 1;

        %%% rock
%%erro_df(i)=sum(sum(cam(indexMin).tempMatrizErro)/tamanho_gray);

        %%% rock %%%matriz_erro(offset+i,2)=indexMin; %%%

        else % CRIA NOVA CÂMERA, nao conseguiu classificar
ncam=ncam+1; % incrementa numero de câmeras
        %frameCount(ncam)=1; % cria contador de frames da camera k
cam(ncam).numFrames=1; % cria contador de frames da camera
k
        %cam(ncam).frames(frameCount(ncam))=entrada(i); %
classifica o frame i como pertencente à camera k
cam(ncam).frames(cam(ncam).numFrames)=entrada(i); %
adiciona o frame i à camera k
cam(ncam).df_frame = entrada(i); % cria o frame padrao da
camera k
        cam(ncam).OutputFile = avifile([defaultOutputPath,
'cam_',int2str(ncam),'.avi'], 'fps', videoInfo.FramesPerSecond, 'quality',
videoInfo.Quality); % cria arquivo de saída da câmera k e deixa aberto para
append
        cam(ncam).matrizErro = double (zeros(numLinhas,
numColunas,1)); % cria matriz de erros da câmera k
cam(ncam).tempMatrizErro = double (zeros(numLinhas,
numColunas,1)); % cria matriz temporária de erros da câmera k

        %%% rock %%%erro_df(i)=0;

        %%% rock %%%matriz_erro(offset+i,2)=ncam; %%%

        end % fim if (currentFrameIndex > 150 && min(erro) < 12)
        end % fim if min(erro)<8
        end % fim if classificou==0
end % fim for i=1>windowSize
offset=offset>windowSize;

        for (k=1:ncam) % grava todos os frames nos respectivos arquivos das
câmeras (acesso ao HD - LENTO!!!)
cam(k).OutputFile = addframe(cam(k).OutputFile, cam(k).frames);
cam(k).frames = cam(k).df_frame; % nao funcionou o clear
cam(k).frames;
cam(k).numFrames=0;
        end

end %while(offset<videoInfo.NumFrames)

        for (k=1:ncam) % fecha todos os arquivos das câmeras
cam(k).OutputFile = close(cam(k).OutputFile);
        end

disp (['Número de câmeras detectadas ' , ncam]);

%for(k=1:ncam)

```

```

%      nframes=nframes+ size(cam(k).frames,2);
%end
%nframes

t=toc;
disp(['Tempo de compilação de ' , int2str(videoInfo.NumFrames) , ' quadros: '
, int2str((t-rem(t,3600))/3600) , 'h ' , int2str(rem(t,3600)/60) , 'min ' ,
int2str(rem(rem(t,3600),60)) , 'seg ' ]);
clear all

%%% rock %%%
%{
%%% PLOTA OS ERROS E LUMINANCIAS DOS FRAMES
figure(1);
stem(lum_media);
axis([0 100 40 180]);
xlabel('frames');
ylabel('luminância média do frame');
figure(2);
for (k=1:ncam)
    lum_med_df(k)=sum(sum(cam(k).df_frame.cdata(:,:,1))/tamanho_gray);
    stem(lum_med_df);
end
axis([1 ncam 40 180]);
xlabel('câmera');
ylabel('luminância média dos frames default');

for (z=1:ncam)
    for(k=1:cam(z).numFrames)
        lum_por_cam(z,k)=sum(sum(cam(z).frames(k).cdata(:,:,1))/tamanho_gray);
    end
end
figure(3);
stem(lum_por_cam(1,:),ones(size(lum_por_cam(1,:))),'-or');
hold on
stem(lum_por_cam(2,:),ones(size(lum_por_cam(2,:))),'-og');
stem(lum_por_cam(3,:),ones(size(lum_por_cam(3,:))),'-ob');
stem(lum_por_cam(4,:),ones(size(lum_por_cam(4,:))),'-oc');
stem(lum_por_cam(5,:),ones(size(lum_por_cam(5,:))),'-om');
stem(lum_por_cam(6,:),ones(size(lum_por_cam(6,:))),'-oy');
stem(lum_por_cam(7,:),ones(size(lum_por_cam(7,:))),'-ok');
stem(lum_por_cam(8,:),ones(size(lum_por_cam(8,:))),'-o','Color',[.39 .56
.33]);

axis([120 170 0 1]);
xlabel('luminância dos 300 frames classificados');
ylabel('vm-1 vd-2 az-3 cy-4 mg-5 am-6 pt-7 bc-8');

saida=fopen('tabela.txt','w');
ha=fprintf(saida, '%6.0f %4.2f %4.2f %4.2f %4.2f %4.2f %4.2f %4.2f %4.2f %4.2f
%4.2f %4.2f %4.2f %4.2f %4.2f %4.2f %4.2f %6.0f\n', matriz_erro);
fclose(saida);
%}

```

**APÊNDICE II – TABELA DE DIFERENÇA DE LUMINÂNCIA
(CRIAÇÃO DE REPOSITÓRIOS DE CÂMERAS)**

Quadro	Diferença de Luminância entre o quadro em análise e o quadro padrão das câmeras:								Câmera Classif.
	cam 01	cam 02	cam 03	cam 04	cam 05	cam 06	cam 07	cam 08	
1	0	0	0	0	0	0	0	0	1
2	0,9254	0	0	0	0	0	0	0	1
3	1,3829	0	0	0	0	0	0	0	1
4	1,5984	0	0	0	0	0	0	0	1
5	1,5929	0	0	0	0	0	0	0	1
6	1,6994	0	0	0	0	0	0	0	1
7	11,944	0	0	0	0	0	0	0	2
8	11,9897	0,7463	0	0	0	0	0	0	2
9	11,9733	0,9759	0	0	0	0	0	0	2
10	11,9624	1,0943	0	0	0	0	0	0	2
11	11,9621	1,0584	0	0	0	0	0	0	2
12	11,9543	1,0872	0	0	0	0	0	0	2
13	18,9374	29,6987	0	0	0	0	0	0	3
14	18,8325	29,6703	0,916	0	0	0	0	0	3
15	18,8238	29,7043	1,2929	0	0	0	0	0	3
16	18,8385	29,7121	1,4818	0	0	0	0	0	3
17	18,8571	29,7545	1,4101	0	0	0	0	0	3
18	18,891	29,8101	1,4768	0	0	0	0	0	3
19	8,8438	14,544	19,1246	0	0	0	0	0	4
20	8,8359	14,5865	19,2835	1,0712	0	0	0	0	4
21	8,9058	14,7236	19,479	1,7055	0	0	0	0	4
22	8,9418	14,7711	19,5439	1,8613	0	0	0	0	4
23	8,9221	14,7509	19,5093	1,7601	0	0	0	0	4
24	10,3391	16,3891	18,636	19,6833	0	0	0	0	5
25	10,2582	16,3023	18,5083	19,7636	0,9214	0	0	0	5
26	10,2291	16,2825	18,4761	19,7807	1,291	0	0	0	5
27	10,1906	16,2206	18,4238	19,7431	1,34	0	0	0	5
28	10,2024	16,2474	18,4697	19,7558	1,3705	0	0	0	5
29	10,1763	16,2128	18,4047	19,7209	1,336	0	0	0	5
30	10,2141	16,2597	18,4498	19,7735	1,3693	0	0	0	5
31	21,4045	34,6893	30,3384	38,4654	40,1537	0	0	0	6
32	21,3228	34,6241	30,2649	38,4151	40,1795	0,9055	0	0	6
33	21,3315	34,6357	30,2745	38,4394	40,2461	1,3656	0	0	6
34	21,3972	34,7187	30,3364	38,5337	40,3496	1,5579	0	0	6
35	21,4549	34,7876	30,3977	38,6084	40,4363	1,595	0	0	6
36	21,5005	34,8401	30,4559	38,661	40,4882	1,6361	0	0	6

37	14,0514	20,3183	21,0951	25,0961	25,7069	15,3516	0	0	7
38	13,9288	20,1845	20,9443	24,9236	25,5456	15,3489	0,7316	0	7
39	13,8091	20,0452	20,8147	24,7588	25,3908	15,2662	0,946	0	7
40	13,7849	20,0083	20,7795	24,729	25,3579	15,2417	1,051	0	7
41	13,7691	19,9865	20,7615	24,7038	25,3328	15,2199	1,0268	0	7
42	10,4928	16,488	18,5084	19,9146	2,823	13,7785	23,8861	0	5
43	10,4696	16,4516	18,4536	19,8705	2,3754	13,7673	23,9527	0	5
44	10,3917	16,3524	18,3725	19,7961	2,1872	13,6932	23,8915	0	5
45	10,4096	16,3952	18,3583	19,8381	2,1228	13,7272	23,9161	0	5
46	10,398	16,3752	18,3443	19,812	2,0936	13,6949	23,8881	0	5
47	10,4141	16,3803	18,3339	19,8174	1,9804	13,7037	23,8887	0	5
48	10,4443	16,4099	18,3607	19,8166	1,9379	13,7331	23,9049	0	5
49	21,5768	34,6116	30,5083	38,7343	40,2749	3,3084	40,2944	0	6
50	21,4747	34,5081	30,4053	38,6187	40,2644	2,915	40,1828	0	6
51	21,5197	34,5952	30,4634	38,7106	40,3724	2,8217	40,2653	0	6
52	21,6034	34,6887	30,5304	38,8236	40,4881	2,8152	40,3849	0	6
53	21,5868	34,6666	30,5171	38,8091	40,4865	2,6339	40,3343	0	6
54	21,6522	34,724	30,5587	38,8698	40,5277	2,6022	40,4049	0	6
55	5,8526	35,8814	35,729	45,5194	46,1317	30,2157	48,1566	0	1
56	5,076	35,6862	35,5348	45,3098	45,9194	30,1783	47,9519	0	1
57	4,6276	35,6144	35,457	45,232	45,8363	30,1491	47,8655	0	1
58	4,3806	35,6667	35,4914	45,2749	45,9119	30,1849	47,8916	0	1
59	4,185	35,6986	35,5346	45,3333	45,9538	30,2266	47,9319	0	1
60	4,0361	35,7791	35,5983	45,4058	46,0332	30,2907	47,9995	0	1
61	14,2734	21,019	21,6969	25,9113	26,4973	15,9836	3,1434	0	7
62	14,2539	20,8983	21,5587	25,7346	26,345	15,8485	2,5995	0	7
63	14,1934	20,7681	21,4367	25,5834	26,1976	15,7502	2,2874	0	7
64	14,1385	20,6873	21,3782	25,4845	26,1157	15,6904	2,1001	0	7
65	14,1037	20,6652	21,3339	25,4579	26,079	15,6525	2,0022	0	7
66	14,0871	20,6273	21,3043	25,426	26,0511	15,6511	1,9793	0	7
67	22,1975	36,9206	31,6981	40,4296	42,0506	5,6745	41,5147	0	6
68	22,1266	36,8203	31,6034	40,3397	41,935	5,0034	41,5251	0	6
69	22,2013	36,9041	31,6628	40,4357	42,0133	4,6995	41,639	0	6
70	22,2622	36,9864	31,7265	40,5282	42,1171	4,4778	41,7017	0	6
71	22,2465	36,9616	31,7166	40,5165	42,0949	4,1975	41,709	0	6
72	22,2747	36,9851	31,7411	40,5282	42,1027	3,987	41,7231	0	6
73	5,1217	35,1866	34,9375	44,1947	45,0513	29,0957	47,3213	0	1
74	4,5691	35,0416	34,765	44,0126	44,8581	29,0449	47,1709	0	1
75	4,2163	35,0132	34,7307	43,9685	44,788	29,0167	47,1233	0	1
76	3,977	35,0072	34,7604	43,9923	44,8276	29,0566	47,1551	0	1
77	3,7357	35,0532	34,7924	44,0449	44,8595	29,0835	47,1997	0	1

78	3,3956	34,8838	34,6631	43,8548	44,6872	28,9604	47,0566	0	1
79	17,5194	28,5961	2,6429	37,3197	35,6915	20,9796	36,7845	0	3
80	17,482	28,4154	2,1595	37,1227	35,4604	20,8269	36,5786	0	3
81	17,4381	28,2868	1,9694	36,9989	35,3375	20,7515	36,4496	0	3
82	17,4065	28,2668	1,888	36,9909	35,3188	20,7436	36,4153	0	3
83	17,4056	28,2693	1,7531	36,98	35,3004	20,7253	36,405	0	3
84	17,4097	28,2378	1,7184	36,9652	35,2973	20,7128	36,3861	0	3
85	8,1336	14,2353	19,2793	2,8245	18,4538	11,1167	22,1159	0	4
86	8,0172	14,0725	19,2344	2,375	18,2273	11,0083	21,954	0	4
87	7,9823	13,9926	19,2553	2,1817	18,1413	10,9761	21,896	0	4
88	7,9435	13,9384	19,2026	2,0486	18,0891	10,9552	21,8322	0	4
89	7,9253	13,897	19,1651	1,925	18,0415	10,9321	21,7876	0	4
90	7,9064	13,8657	19,1225	1,8301	18,01	10,9058	21,7559	0	4
91	12,6284	18,7583	19,8312	23,4387	23,9428	14,0759	0,4135	0	7
92	12,5653	18,6941	19,7185	23,4694	23,8371	13,9777	0,2864	0	7
93	12,5033	18,6161	19,6402	23,4244	23,7409	13,8892	0,274	0	7
94	12,4879	18,5894	19,611	23,3829	23,7094	13,8659	0,2819	0	7
95	12,4559	18,5533	19,5861	23,3547	23,6647	13,8381	0,2976	0	7
96	9,782	17,5087	20,8017	17,7995	22,9557	11,1012	26,7415	0	8
97	9,7238	17,4113	20,7452	17,7216	22,9082	11,0893	26,8058	0,9974	8
98	9,719	17,4044	20,736	17,7116	22,8958	11,0649	26,8773	1,3052	8
99	9,7035	17,4268	20,7404	17,7256	22,9204	11,0746	26,8938	1,509	8
100	9,7283	17,4537	20,7703	17,7485	22,9167	11,0982	26,9374	1,4979	8
101	9,7197	17,4344	20,7688	17,7094	22,9051	11,0869	26,9105	1,391	8
102	9,7172	17,4214	20,7473	17,711	22,9078	11,0895	26,8832	1,4812	8
103	4,0789	34,5795	35,1444	44,1722	44,3921	28,5084	47,3974	43,467	1
104	3,6295	34,5585	35,108	44,1932	44,3623	28,4441	47,36	43,5803	1
105	3,4213	34,6635	35,1818	44,3082	44,4974	28,5138	47,4571	43,727	1
106	3,3186	34,8078	35,2983	44,4774	44,6383	28,6223	47,5963	43,8896	1
107	3,1756	34,853	35,3691	44,5253	44,7007	28,652	47,6448	43,933	1
108	3,1478	34,9899	35,5069	44,6577	44,849	28,7573	47,7897	44,0359	1
109	11,8707	18,1171	18,917	22,9279	23,2812	13,4911	0,7869	24,7879	7
110	11,8507	17,9982	18,777	22,7606	23,1072	13,3744	0,5736	24,5924	7
111	11,8241	17,9101	18,6897	22,6361	23,0181	13,2951	0,531	24,4952	7
112	11,7771	17,8526	18,6384	22,5574	22,9548	13,2494	0,5004	24,428	7
113	11,7415	17,8123	18,5865	22,5364	22,8963	13,2207	0,49	24,3865	7
114	11,7259	17,7682	18,5385	22,4856	22,8662	13,1999	0,5022	24,3444	7
115	11,2283	2,0171	22,7528	27,3868	27,5917	19,8278	31,5414	28,3807	2
116	11,1543	1,6948	22,6306	27,2722	27,48	19,7543	31,505	28,2868	2
117	11,1595	1,6156	22,6159	27,2599	27,4804	19,7544	31,555	28,2828	2
118	11,2014	1,6356	22,6729	27,3313	27,5623	19,7897	31,6173	28,3211	2

119	11,2302	1,6094	22,7089	27,3782	27,591	19,8206	31,6358	28,3742	2
120	11,2251	1,5945	22,7043	27,3782	27,5749	19,8155	31,6241	28,3968	2
121	16,3802	26,6931	1,6787	35,8774	33,8514	19,6788	36,9446	35,2505	3
122	16,2705	26,6589	1,4605	35,7574	33,6874	19,5851	36,771	35,1346	3
123	16,2788	26,7147	1,3471	35,7657	33,6693	19,5902	36,7632	35,1368	3
124	16,3118	26,785	1,317	35,8355	33,7555	19,6412	36,8377	35,2074	3
125	16,3414	26,8111	1,2445	35,8686	33,7796	19,6717	36,876	35,2579	3
126	16,3252	26,7954	1,2461	35,8556	33,7639	19,6625	36,8479	35,2292	3
127	8,734	15,7042	20,8605	4,9856	19,9245	10,9664	25,2029	16,4453	4
128	8,6637	15,6376	20,891	4,4111	19,821	10,8772	25,1143	16,3273	4
129	8,622	15,5717	20,8812	4,0116	19,756	10,8339	25,0685	16,2171	4
130	8,6261	15,5618	20,8608	3,7385	19,756	10,84	25,0347	16,213	4
131	8,5972	15,5211	20,8059	3,4473	19,7101	10,8326	24,9797	16,1703	4
132	8,6123	15,509	20,8007	3,233	19,7155	10,8256	24,9739	16,1989	4
133	11,6223	17,6822	19,1869	21,9258	22,711	13,1791	1,2984	24,163	7
134	11,5342	17,5761	19,0532	21,8909	22,5787	13,0418	1,0323	24,0512	7
135	11,4717	17,4898	18,9577	21,8248	22,4719	12,9623	0,9357	23,9448	7
136	11,445	17,4466	18,9287	21,7782	22,4397	12,9311	0,8961	23,8869	7
137	11,4696	17,4784	18,9525	21,8148	22,4673	12,9522	0,8692	23,9305	7
138	10,2856	17,7304	19,5325	20,8674	5,689	14,0185	27,4773	22,9729	5
139	10,2533	17,6672	19,4541	20,7848	5,0352	13,9703	27,4699	22,8957	5
140	10,2058	17,6206	19,3774	20,7311	4,5865	13,9107	27,4564	22,8647	5
141	10,2296	17,6524	19,3826	20,7638	4,2726	13,9304	27,478	22,9035	5
142	10,2042	17,6293	19,3696	20,7419	3,9306	13,9177	27,46	22,8728	5
143	10,2216	17,6403	19,4043	20,7534	3,6649	13,9381	27,4863	22,9016	5
144	10,234	17,6679	19,4101	20,7797	3,4235	13,9584	27,5018	22,9255	5
145	21,2267	36,1902	31,8222	39,3779	40,5564	4,083	43,9455	37,8792	6
146	21,1181	36,0755	31,6817	39,2612	40,4895	3,5964	43,8394	37,7617	6
147	21,1387	36,0932	31,6832	39,3093	40,5414	3,3919	43,8579	37,8261	6
148	21,2061	36,1967	31,7594	39,4388	40,6577	3,3021	43,9752	37,9437	6
149	21,2411	36,2491	31,8075	39,4911	40,7143	3,1712	44,0265	38,0118	6
150	21,2705	36,2808	31,8473	39,5299	40,7625	3,0661	44,0656	38,0317	6
151	11,2737	39,4712	40,4324	49,0279	50,2388	32,7579	54,8809	49,1489	1
152	9,9742	39,2802	40,2659	48,7916	50,0361	32,7007	54,6752	48,9358	1
153	9,1654	39,2881	40,2677	48,8116	50,0352	32,7412	54,6924	48,9618	1
154	8,5231	39,4023	40,3729	48,9522	50,1551	32,814	54,7921	49,0603	1
155	9,5321	18,5838	22,396	18,2479	24,0238	11,6232	29,7561	3,1354	8
156	9,6683	18,655	22,4842	18,3397	24,1046	11,6487	29,8482	2,9032	8
157	9,7894	18,8108	22,6481	18,5414	24,2628	11,728	30,0421	3,0002	8
158	9,7968	18,8512	22,7002	18,607	24,2986	11,7539	30,0848	3,0283	8
159	9,7522	18,7555	22,6346	18,4956	24,2019	11,7087	29,9861	2,7642	8

160	9,7422	18,7335	22,6123	18,4708	24,1831	11,6817	29,9744	2,6924	8
161	9,6992	18,6568	22,5367	18,379	24,0963	11,6436	29,8872	2,507	8
162	15,4095	27,2016	2,5214	35,7235	34,1756	19,7413	38,2181	35,2964	3
163	15,3256	27,068	2,0703	35,5975	34,0447	19,6766	38,0508	35,271	3
164	15,2575	26,963	1,878	35,4607	33,8861	19,5907	37,914	35,1933	3
165	15,2674	26,9876	1,8002	35,4887	33,9245	19,6091	37,9495	35,2146	3
166	15,283	26,992	1,7279	35,5213	33,9561	19,6048	37,9706	35,2362	3
167	10,1559	17,5125	18,2792	20,0781	4,5865	13,083	26,8259	21,73	5
168	10,1495	17,5505	18,4125	20,1033	4,102	13,076	26,8457	21,7539	5
169	10,2246	17,6476	18,5585	20,24	3,8706	13,1519	26,9767	21,8719	5
170	10,2357	17,6918	18,6163	20,305	3,6037	13,1845	27,0587	21,9283	5
171	10,2164	17,6457	18,5556	20,2262	3,3251	13,1566	26,9885	21,854	5
172	10,2113	17,6343	18,5719	20,2311	3,0711	13,1419	26,9769	21,8601	5
173	10,2338	17,6525	18,5772	20,2292	2,8835	13,1733	26,9862	21,8656	5
174	19,486	34,4947	31,2794	37,9609	39,4549	2,7559	42,5945	36,1047	6
175	19,4347	34,4279	31,1902	37,9001	39,4377	2,4057	42,5106	36,0443	6
176	19,4549	34,4656	31,2114	37,9512	39,5167	2,2556	42,562	36,0987	6
177	19,5151	34,5662	31,302	38,0537	39,6401	2,2106	42,6712	36,2167	6
178	19,5624	34,6402	31,3613	38,1492	39,7306	2,1389	42,7605	36,2999	6
179	19,6009	34,6729	31,4065	38,1915	39,7615	2,0864	42,7838	36,3355	6
180	8,5847	38,8901	40,0523	48,5736	50,2145	32,8167	54,2402	48,3172	1
181	7,6105	38,7462	39,8914	48,3961	50,0233	32,7864	54,0796	48,1596	1
182	6,9881	38,7302	39,8841	48,3604	50,0147	32,799	54,066	48,114	1
183	6,4959	38,7734	39,9254	48,416	50,0709	32,8283	54,1118	48,1771	1
184	6,0121	38,7767	39,924	48,4157	50,0711	32,8241	54,1122	48,1724	1
185	5,6906	38,845	39,9968	48,4953	50,1236	32,8824	54,1714	48,2442	1
186	11,405	18,5305	20,3359	23,0026	24,3569	13,8246	1,6225	24,8168	7
187	11,3389	18,3595	20,1635	22,7951	24,1563	13,6648	1,1592	24,6046	7
188	11,2772	18,2378	20,0651	22,6635	24,0221	13,5665	1,0353	24,4781	7
189	11,2271	18,1639	19,9987	22,5962	23,9558	13,5238	0,9708	24,4088	7
190	11,2196	18,1354	19,9597	22,5484	23,9138	13,4992	0,9393	24,3741	7
191	9,5486	18,3948	22,4196	18,0046	23,8276	11,5585	29,3219	2,213	8
192	9,4712	18,2493	22,2903	17,8282	23,6663	11,4789	29,3353	1,8672	8
193	9,4535	18,2248	22,2367	17,7882	23,6047	11,4657	29,3803	1,8486	8
194	9,4312	18,2118	22,2129	17,7872	23,6139	11,4539	29,3772	1,8564	8
195	9,4067	18,1787	22,1783	17,7414	23,5662	11,4384	29,3464	1,7714	8
196	9,424	18,1769	22,1837	17,7463	23,5795	11,4445	29,3361	1,7996	8
197	9,4203	18,1649	22,1871	17,7203	23,567	11,4449	29,3342	1,7617	8
198	15,2771	27,6148	2,4389	36,0108	34,3213	20,2693	38,5605	35,7002	3
199	15,1949	27,4733	1,9443	35,8673	34,1816	20,1757	38,3787	35,6403	3
200	15,2017	27,4759	1,8666	35,9044	34,2271	20,2032	38,3744	35,7104	3

201	15,2156	27,483	1,8241	35,9083	34,2492	20,2176	38,3854	35,7422	3
202	15,2372	27,5149	1,8025	35,9273	34,2899	20,229	38,44	35,7462	3
203	15,2328	27,4967	1,7223	35,9268	34,2736	20,2244	38,4131	35,7433	3
204	7,4188	14,2593	19,6304	3,2325	19,4954	11,0839	24,7771	15,6153	4
205	7,347	14,1648	19,6717	2,7592	19,3721	11,0212	24,6774	15,4739	4
206	7,3128	14,1154	19,6665	2,5586	19,2907	10,9761	24,6297	15,4015	4
207	7,2815	14,0852	19,6266	2,3965	19,2494	10,9382	24,5742	15,3531	4
208	7,2708	14,0703	19,5884	2,2926	19,2382	10,9066	24,5425	15,3693	4
209	10,6459	18,135	18,5365	20,1279	3,8774	13,3522	27,0995	21,9687	5
210	10,5342	17,9718	18,4405	20,1569	3,2534	13,2769	26,9813	21,849	5
211	10,5395	17,9661	18,4205	20,2102	2,9728	13,2663	26,9645	21,8261	5
212	10,5176	17,9335	18,4363	20,2142	2,7373	13,2694	26,9507	21,8444	5
213	10,5113	17,9272	18,4326	20,2107	2,5332	13,2728	26,9376	21,8404	5
214	10,5259	17,9273	18,3942	20,1941	2,3719	13,2653	26,9136	21,8234	5
215	10,4936	17,8843	18,3643	20,1718	2,2199	13,253	26,8823	21,8105	5
216	19,127	34,0374	31,0662	37,9034	39,6621	2,2263	42,1596	35,8621	6
217	19,0941	34,0085	31,0328	37,8813	39,7031	1,931	42,1288	35,8452	6
218	19,1292	34,052	31,0716	37,9204	39,7605	1,8335	42,159	35,8885	6
219	19,1758	34,126	31,1523	37,9866	39,8277	1,7982	42,2273	35,9515	6
220	19,236	34,1817	31,1996	38,0605	39,8957	1,7514	42,2934	36,0331	6
221	19,2593	34,2204	31,2186	38,0961	39,9322	1,7214	42,3215	36,0641	6
222	5,5319	37,4734	38,7542	46,9484	49,2134	31,6904	52,999	46,776	1
223	4,8768	37,3079	38,5998	46,7455	49,0309	31,649	52,823	46,6044	1
224	4,501	37,3029	38,5784	46,7352	49,0091	31,6539	52,7903	46,5787	1
225	4,217	37,3605	38,6523	46,8112	49,0881	31,6763	52,8583	46,636	1
226	4,0033	37,428	38,6922	46,8914	49,1493	31,7265	52,918	46,7218	1
227	10,7116	17,8556	18,4327	20,5557	4,3478	13,7531	27,0134	21,9692	5
228	10,7352	17,8292	18,424	20,5226	3,8427	13,7183	27,015	21,9578	5
229	10,8102	17,9363	18,5477	20,682	3,6188	13,7892	27,1878	22,1156	5
230	10,8432	17,996	18,6148	20,7539	3,4076	13,8336	27,2615	22,1737	5
231	10,7854	17,9316	18,5412	20,6695	3,0846	13,7675	27,1605	22,0872	5
232	10,7725	17,932	18,5247	20,6413	2,8801	13,7619	27,1545	22,0564	5
233	10,7903	17,9226	18,5169	20,6581	2,7073	13,7968	27,1493	22,0743	5
234	19,6847	34,4371	31,1753	38,2735	40,3774	2,7418	42,5649	36,2449	6
235	19,6021	34,3473	31,0814	38,2035	40,3579	2,3226	42,4618	36,1786	6
236	19,6502	34,4208	31,1316	38,2959	40,4534	2,2289	42,5478	36,2743	6
237	19,7366	34,5301	31,2273	38,4025	40,5709	2,1909	42,657	36,3813	6
238	19,7305	34,541	31,2473	38,4093	40,594	2,1043	42,6605	36,4032	6
239	19,7157	34,5008	31,2131	38,3507	40,5368	2,023	42,6273	36,3523	6
240	5,0885	37,8962	39,2136	47,5067	50,0961	31,8711	53,7371	47,0725	1
241	4,5106	37,8345	39,1446	47,4281	50,0135	31,912	53,6368	47,0176	1

242	4,1925	37,7934	39,13	47,3894	49,9983	31,9264	53,6049	46,9691	1
243	4,0031	37,8402	39,1383	47,4502	50,044	31,9589	53,6448	47,0266	1
244	3,7946	37,856	39,1479	47,4801	50,0538	31,9609	53,6348	47,0413	1
245	9,9025	17,1282	18,027	19,6892	3,436	13,28	26,7009	21,2826	5
246	10,018	17,1947	18,0822	19,7747	3,1118	13,3076	26,8179	21,3497	5
247	10,0798	17,2783	18,2159	19,9317	2,9819	13,375	27,0102	21,4755	5
248	10,0929	17,2894	18,239	19,9544	2,8036	13,3779	27,0325	21,4644	5
249	10,0457	17,218	18,1844	19,8516	2,527	13,3309	26,9368	21,3999	5
250	10,0598	17,2333	18,1831	19,8597	2,3827	13,3406	26,9237	21,4127	5
251	10,0285	17,1978	18,1324	19,7833	2,2111	13,3017	26,8726	21,3365	5
252	19,9972	34,9919	31,3078	38,7188	41,187	2,9142	42,8979	36,7707	6
253	19,9146	34,8715	31,1917	38,6213	41,1459	2,4848	42,7875	36,676	6
254	19,8794	34,8678	31,1905	38,6306	41,2008	2,3719	42,8017	36,684	6
255	19,9646	34,9612	31,2458	38,7439	41,2985	2,3521	42,9023	36,7891	6
256	19,9745	34,9921	31,2819	38,775	41,3304	2,2941	42,9249	36,8164	6
257	20,0127	35,0662	31,3137	38,8487	41,3835	2,2758	42,991	36,8828	6
258	10,2158	1,4326	23,4031	26,2848	29,7809	19,4487	32,2456	27,4468	2
259	10,1538	1,1072	23,2916	26,126	29,6249	19,4477	32,1219	27,3052	2
260	10,1032	1,0678	23,236	26,0631	29,5665	19,433	32,0709	27,2306	2
261	10,0915	1,0788	23,2208	26,0601	29,5374	19,4303	32,0756	27,2262	2
262	10,0974	1,0889	23,2087	26,0454	29,5188	19,4324	32,0654	27,2303	2
263	10,0608	1,0416	23,1604	25,9805	29,445	19,3899	31,9888	27,169	2
264	13,1901	20,0299	22,2197	25,0173	27,3076	15,2724	4,7796	26,5666	7
265	13,0776	20,0065	22,1011	24,8195	27,1514	15,1073	3,9723	26,3776	7
266	13,0067	19,9749	22,0395	24,7273	27,0677	15,0213	3,6933	26,2928	7
267	12,996	19,9717	22,0153	24,7092	27,053	14,9985	3,5432	26,273	7
268	12,9986	19,9717	22,0253	24,7154	27,0451	15,0127	3,444	26,2739	7
269	10,4467	19,1458	23,0537	19,2366	25,5374	12,2649	29,673	3,795	8
270	10,3288	18,9762	22,9363	19,079	25,3704	12,1962	29,6805	3,3166	8
271	10,3006	18,9201	22,91	19,0515	25,3129	12,1938	29,7355	3,1762	8
272	10,3232	18,9419	22,9305	19,0754	25,3487	12,2023	29,745	3,1759	8
273	10,3406	18,9719	22,939	19,0782	25,3792	12,2107	29,7537	3,1202	8
274	10,3723	19,0171	22,9759	19,1375	25,3935	12,2245	29,7904	3,1021	8
275	10,3462	18,9829	22,9509	19,0924	25,3504	12,2166	29,7713	2,9928	8
276	17,0262	29,7199	4,8095	38,1008	37,3004	21,9162	39,7236	37,24	3
277	16,923	29,5663	4,0698	37,9717	37,1588	21,816	39,5434	37,1784	3
278	16,895	29,4971	3,7799	37,9033	37,053	21,7702	39,4762	37,1462	3
279	16,9032	29,5293	3,7141	37,9261	37,0832	21,7851	39,4993	37,1643	3
280	16,9222	29,5799	3,689	37,9719	37,1415	21,8183	39,5323	37,2141	3
281	16,973	29,636	3,6603	38,0142	37,1929	21,853	39,5983	37,2648	3
282	7,5898	14,3115	19,6599	3,1059	19,6326	10,8276	23,8676	14,7605	4

283	7,5381	14,2237	19,7231	2,6575	19,538	10,7742	23,8002	14,6384	4
284	7,4905	14,1769	19,7301	2,5072	19,4854	10,7487	23,7618	14,5564	4
285	7,4853	14,1699	19,6999	2,4395	19,4485	10,7196	23,734	14,5601	4
286	7,4577	14,143	19,6927	2,3134	19,4589	10,7168	23,7227	14,5364	4
287	9,8321	16,8136	19,9872	20,1524	6,914	14,0166	26,2805	21,8366	5
288	9,7182	16,6815	19,8417	20,185	5,9225	13,9135	26,1835	21,7583	5
289	9,6992	16,6498	19,7688	20,2139	5,4657	13,8934	26,1432	21,7547	5
290	9,6857	16,6426	19,7754	20,2148	5,1394	13,8848	26,168	21,753	5
291	9,7043	16,6543	19,7827	20,2087	4,8056	13,9045	26,1568	21,7421	5
292	9,7161	16,6599	19,7913	20,2202	4,5502	13,9181	26,1753	21,7543	5
293	9,6651	16,6237	19,7469	20,1809	4,2704	13,8915	26,1289	21,7105	5
294	19,8962	35,1443	31,2841	39,3085	41,4886	3,8142	42,4085	36,8523	6
295	19,7829	35,0082	31,1493	39,1711	41,4192	3,269	42,2741	36,7287	6
296	19,7926	35,042	31,1588	39,2284	41,4777	3,1717	42,3113	36,7722	6
297	19,874	35,1712	31,2588	39,3529	41,6073	3,2007	42,4214	36,9175	6
298	19,9176	35,2192	31,2945	39,4207	41,6837	3,1753	42,4765	36,9619	6
299	19,9896	35,319	31,3735	39,5209	41,7822	3,2054	42,5742	37,0529	6

APÊNDICE III - ARTIGO A SER SUBMETIDO A EVENTO INTERNACIONAL

AUTOMATIC ASSINCHRONOUS VIDEO DEMULTIPLEXING

Marcelo Hor-Meyll Alvares
Brazilian Federal Police
João Pessoa, PB, Brazil
hormeyll.mfmha@dpf.gov.br

Rafael Dueire Lins
Universidade Federal de Pernambuco
Recife, PE, Brazil
rdl@ufpe.br

ABSTRACT

The work on real forensics videos involving banking images lead to the method presented here: a software approach to the problem of demultiplexing time multiplexed security videos. This approach is of particular interest in cases in which, for some reason, the regular demultiplexing hardware/software is not available. Based on comparisons of the mean frame luminance and having low frequency characteristics of the scenes as references, this approach is being used with good results by video experts in Brazil to help locating sequences of particular interest in surveillance time multiplexed videos recorded in magnetic video cassettes.

Index Terms— demultiplexing, video, forensics.

1. INTRODUCTION

Security surveillance systems are everywhere as a result of the feeling of insecurity and mistrust which grows amongst the people of good will, reinforced by the cost drop of equipments and systems as a result of mass production. Normally designed to develop remote surveillance of wide areas, the video provided by security systems are commonly presented to the police forces in the case of incidents. Many surveillance systems employ low cost, low resolution sensors that make use of distorting spherical lenses to survey wide areas, combined with multiplexing equipments as a way to save storage space. The lack of standardization in the multiplexing techniques makes the problem even worse once there are many proprietary incompatible techniques that make the demultiplexing process sometimes very complicated.

As a result of all the aforementioned problems, a growing number of videotapes containing inadequate images is presented to forensic experts to work on.

Conceived to demultiplex a surveillance system videotape which recorded a robbery at a bank agency, the demultiplexing solution presented here, was tested in several videos yielding satisfactory results.

2. VIDEO CHARACTERISTICS

The video used as starting point for this work is a gray scale interlaced video with images of a bank agency. The frames were recorded on a magnetic VHS videotape by a time multiplexed CFTV system with an unknown number of cameras. Witnesses affirmed that a certain crime was committed at the agency in a location covered by the cameras. The video was recorded in ELP mode (8hrs) in that tape, but in an unknown position. The video had multiplexation information in the retrace lines and the demultiplexing hardware was unavailable. An initial frame-by-frame analysis revealed that the multiplexation did not follow a regular order but, somehow, it tried to sample the 8 cameras of the system uniformly as depicted in scheme shown in Figure 1.

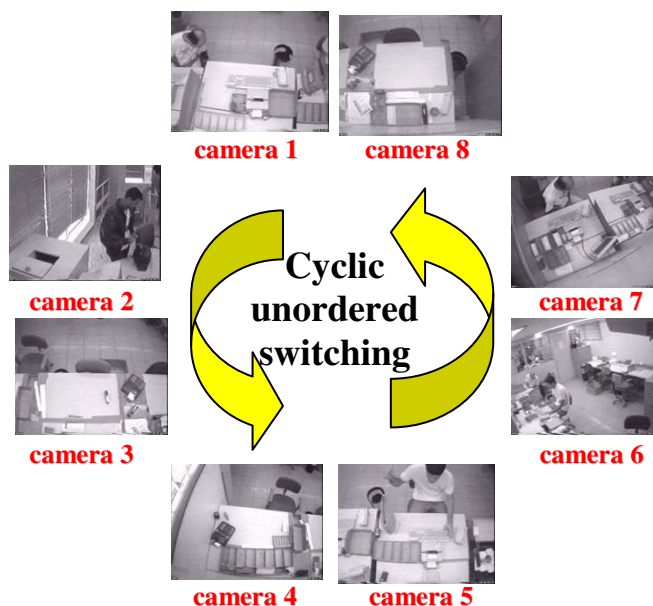


Figure 1 - Multiplexation scheme of the analyzed video

Once there was no way to demultiplex the video using the specific hardware, a software approach was the solution adopted to the problem.

3. PRE-PROCESSING THE VIDEO

As already mentioned, the video under analysis was in VHS thus, it had to be digitized before being enhanced and processed. This section focuses in the digitalization process, frame deinterlacing and finally noise removal and frame identification process.

3.1. Capturing video

Beginning by the digitalization of the video recorded in the videotape, it was necessary to pre-process the frames prior to trying the classification algorithm. As long as the analysis of video by the forensics experts aims, in general, the identification of suspects, image quality preservation is a priority. Thus, to find the adequate resolution to capture de video, the following data were analyzed: Once the video was an NTSC interlaced analog one, disregarding the 30% resolution reduction caused by the Kell effect, one has to sum up the 242.5 vertical active lines of the odd fields with the 240.5 [1] one from the even field. This way, the vertical digital resolution (system based) should be at least 483 lines. To determine the minimum horizontal resolution (source based), one should first consider the active line interval of the NTSC system 52,6 μ s [2] (figure 2)

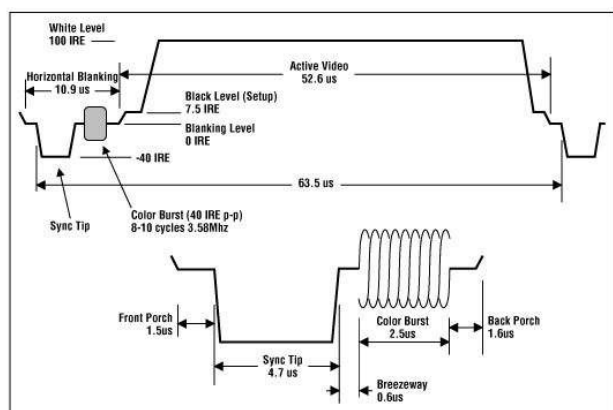


Figure 2 – Video line signal timings (ITU-R BT.470-7)

Based on this interval, it is possible to calculate the number of oscillations of a 4.2MHz video signal line:

$$\text{horizontal resolution} = \frac{4.2 \cdot 10^6 \frac{\text{oscillatio ns}}{s}}{1} = 220.92 \frac{\text{oscillatio ns}}{\text{line}}$$

$$\frac{4.2 \cdot 10^6 \frac{\text{oscillatio ns}}{s}}{52.6 \cdot 10^{-6} \frac{s}{\text{line}}}$$

Using Shannon's sampling theory, one obtains 442 samples/line for a lossless horizontal representation. This way, a 442x485 pixels resolution should be enough not to lose any analog detail if it was not for the aspect ratio. A 442x483 pixels resolution would do a 0,91 aspect ratio, once only 640x480 (4/3 or 1.33), 720x480 (16/9 or 1.50) or more rarely 720x486 (1,48) are commercially available. A 16x9

aspect ratio capture card would add two 40x480 rows making a no information frame to the acquired video. So, as concluded by [3], this work also concluded that a 640x480 pixels frame grabber would be the best choice and do the job suitably for forensics applications.

3.2. Deinterlacion

Once the time multiplexation is field based, it is common to see in security videos, hybrid frames that would introduce noise in the classification process. Thus, prior to starting any classification, deinterlacing the video is a must. Normally the deinterlacion process is associated with discarding of one of the fields or merging them together. In forensic applications, once the frames will be analyzed statically, the apparent division by two of the vertical resolution caused by the frames unfolding process will bring no harm to the problem addressed herein. This way, the best option is to perform a simple field unfolding, not losing the image sharpness or resolution and separating both fields.

3.3. Noise removal and frame identification

Due to tracking misalignment of the recording heads of the VCRs used, one often finds, not only for the settling time, but sometimes for all the video, static noise and visible retrace lines in the active video interval. These problems can be introduced by various sources, but more frequently by some mechanical malfunction or magnetic tape defect. These problems are often found because VCR security systems are normally switched on 24/24hs, using continuously reutilized tapes. The results are dirty VCR heads and low quality tapes causing loss of track and synchronization problems.

These contaminated frames can introduce noise in the classification algorithm, sometimes taking it out of a recovering region. That is the reason why they should be removed in a preceding phase called here as pre-processing. Using Avisynth [4] combined with Virtualdub [5] both open source freeware software, the frames are cropped and static contaminated frames are manually analyzed and eventually discarded. Original frames sequence numbers and time-stamps are added to the frames in some quiet region of the video to allow eventual videos appending. One last step in the pre-processing phase is performed: the video codification using Indeo 5.1 codec to fulfill the input video requirements to be processed by MATLAB.

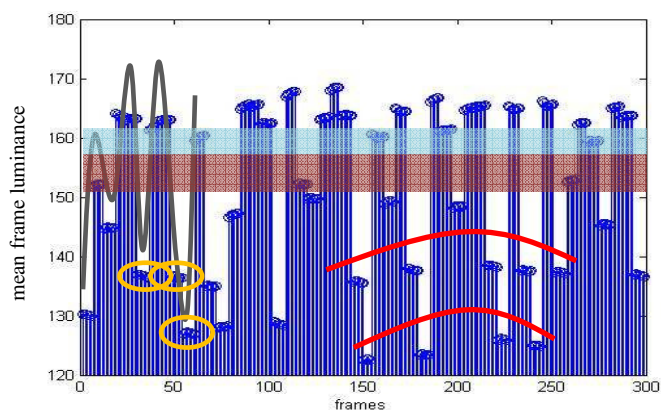
4. PROCESSING THE VIDEO

4.1. Platform and frames behavior

As already mentioned, MATLAB was chosen to process the images. The main reason for this choice was that all the demultiplexing solution was developed during a real case in which there were arrested people and time was rushing out.

MATLAB offers some really interesting functions that speeded up the process.

The initial frame-by-frame analysis showed some characteristics of the frames behavior as can be seen in the following graph (Figure 3):



CHARACTERISTICS SUMMARY





-  1-well defined horizontal levels
-  2-sharpen frame-to-frame changes
-  3-neighbor frames with almost no luminance change
-  4-frames from the same camera have slow luminance variations

Figure 3 – Graph of the mean luminance of a 300 frames time multiplexed video test file

Based on these characteristics, a MATLAB script was developed to perform the time demultiplexation, transforming a single multi-camera video file into N single-camera video files, where N represents the total number of cameras in the system.

4.2. Classification and frame comparison

The comparison variable chosen was the normalized mean luminance of the frame. The main idea is to group the frames as long as they show little difference in luminance, actually smaller than a pre-defined threshold empirically found. When the mean luminance differences increases beyond a limit, the algorithm assumes that that frame belongs to a different camera. The process works automatically creating a “new camera”, that in fact is a structure with various information, when the comparison between the current frame and all the “existing cameras” is greater than the threshold. For comparison means, each camera is represented by a reference frame, obtained by a low-pass filtering of a limited window containing the last

frames classified as belonging to that camera. One example of this filtering is shown in Figure 4.



Figure 4 – Example of some camera frames (up) and the respective camera reference frame

As can be seen above, the frames show a bank entrance revolving door. The noise caused by the constant flux of people wearing clothes of different colors is a typical example of how this filtering technique can be valuable. The filter time constant has to be slow enough to reject the unwanted noise but not too slow not to be able, for example, to adapt to the dimming daylight.

Besides the filtering one, other techniques showed to be useful whenever processing multiplexed videos. In the current version, for example, a camera creation inertia, is employed to help in the camera creation process. The idea was to introduce a progressive difficulty in the creation of new cameras: easier in the beginning of the video and making it more difficult as the video continues.

5. RESULTS

Some real cases have already been solved using the presented technique. Despite MATLAB 7 limitation in processing video files not bigger than 2GB, TRUECOLOR videos and Y8 grayscale ones have been tested with good results. The following table summarizes the tests performed.

CPU used	
Processor	Intel Core 2 Duo 6300, <i>clock</i> 1.86GHz
MB	Intel DG965SS – <i>chipset</i> Intel G965
RAM	Kingston KVR667D2N5/1G – 1GB
HD	Samsung HD160JJ – 160GB/7200RPM/8MB <i>cache</i>
Processing data	
Mean pre-processing frame rate (deinterlacing, cropping, insertion of frame number and time-stamp, and Indeo 5.1 re-encoding) – Avisynth and Virtualdub	10-11fps
Cameras in the video	7-15
Mean processing frame rate -	3-6fps

MATLAB	
Classification efficiency	98-100%
Number of frames processed	12,500-57,000

Some classification problems were identified in the videos processed, normally they are associated to the loss of synchronization (Figure 5), the static contamination comprising considerable portion of the frame (Figure 6) or extremely low illumination in the scene (Figures 7 and 8).



Figure 5 – Loss of synchronization



Figure 6 – Static contamination



Figures 7 and 8 – Similar low illumination frames classified as belonging to the same camera

Contrariwise to what was concluded by the Cognitech team in references [6] and [7], the experiments performed in this work found that the mean luminance can represent quite well the frame content, as long as other techniques are associated with the luminance information. The solution developed was used in several other forensic videos under analysis at the Federal Police of Brazil (Polícia Federal do Brasil). If this paper were accepted, the executable code for the solution described herein may be freely obtained by contacting the first author of this paper, provided it is for

non-commercial use. Besides that, the solution described may also be demonstrated at ICIP 2009.

6. REFERENCES

- [1] GROB, BERNARD, Basic Television, EUA: New York, 1964;
- [2] ITU-R BT 470-7, Conventional analog television systems;
- [3] RUDIN, LEONID I. *et al*, Sampling theory for digital video acquisition: the guide for the perplexed user;
- [4] AVREY, LEE, AviSynth available at <http://www.avisynth.org> ;
- [5] AVREY, LEE, VirtualDub, available at <http://www.virtualdub.org>;
- [6] RUDIN, LEONID I. *et al*, Meaningful Automatic Video Demultiplexing with Unknown Number of Cameras, Contrast Changes, and Motion, EUA: Cognitech Inc., Pasadena, CA, Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference, Publication Date: 15-16 Sept. 2005;
- [7] RUDIN, LEONID I. *et al*, Software-based universal demultiplexing: threshold-free energy minimization approach, SPIE (The International Society for Optical Engineering). Proc. SPIE Vol. 4232, p. 513-520, Enabling Technologies for Law Enforcement and Security, Publication Date: 2/2001;
- [8] GOODRIDGE, STEVEN G., Content-based software demultiplexing of surveillance video, Proc. SPIE Vol. 4232, p. 489-496, Enabling Technologies for Law Enforcement and Security, Simon K. Bramble; Edward M. Carapezza; Leonid I. Rudin; Eds.
- [9] COLMENAREZ, ANTONIO J.*et al*, Video monitoring and surveillance systems capable of handling asynchronously multiplexed video, United States Patent 6,999,613, February 14, 2006, Maracaibo, Venezuela;

ANEXO I – DVD