Universidade de São Paulo

Escola de Engenharia de São Carlos
Departamento de Engenharia Elétrica

André Calheiros Silvestre

Estabilização digital em tempo real de imagens em seqüência de vídeos.

USP – São Carlos

2007

André Calheiros Silvestre

Estabilização digital em tempo real de imagens em seqüência de vídeos.

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo para obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Processamento de Sinais e Instrumentação
Orientador: Prof. Dr. Valentin Obac Roda

USP – São Carlos

2007

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

S587e

Silvestre, André Calheiros
Estabilização digital em tempo real de imagens em seqüência de vídeos / André Calheiros Silvestre ; orientador Valentin Obac Roda. -- São Carlos, 2007.

Dissertação (Mestrado-Programa de Pós-Graduação em Engenharia Elétrica e Área de Concentração em Processamentos de Sinais e Instrumentação) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2007.

1. Processamento digital de imagens. 2. Estabilização de imagens em tempo real. 3. Estimação de movimento. 4. Compensação de movimento. I. Título.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **ANDRÉ CALHEIROS SILVESTRE**

Dissertação defendida e julgada em 10/05/2007 perante a Comissão Julgadora:

APROVADO

Prof. Associado **VALENTIN OBAC RODA (Orientador)**
(Escola de Engenharia de São Carlos/USP)

APROVADO

Prof. Associado **ADILSON GONZAGA**
(Escola de Engenharia de São Carlos/USP)

APROVADO.

Prof. Dr. **JOÃO DO ESPÍRITO SANTO BATISTA NETO**
(Instituto de Ciências Matemáticas e de Computação/USP)

Prof. Associado **GERALDO ROBERTO MARTINS DA COSTA**
Coordenador do Programa de Pós-Graduação em
em Engenharia Elétrica e
Presidente da Comissão de Pós-Graduação da EESC

Ao meu pai, Jurandir, minha mãe, Lucila, demais familiares e amigos, pela ajuda, dedicação e encorajamento.

AGRADECIMENTOS

Agradeço ao Prof. Dr. Valentin Obac Roda pela oportunidade, confiança e apoio na realização deste trabalho e pela significativa contribuição para a minha formação científica e intelectual.

Aos demais professores da área de Processamento de Sinais e Instrumentação do departamento de Engenharia Elétrica pela cooperação e apoio.

Aos colegas do Laboratório de Instrumentação Virtual e Microprocessada pela amizade e cooperação.

Aos funcionários e técnicos do Departamento de Engenharia Elétrica.

RESUMO

SILVESTRE, A.C. **Estabilização digital em tempo real de imagens em seqüência de vídeos.** 2006. 109 f. Dissertação (Mestrado) – Departamento de Engenharia Elétrica, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006.

Podemos afirmar que deslocamentos da imagem em quadros consecutivos de uma seqüência de vídeo são causados por pequenas vibrações da câmera e/ou movimentos desejados pelo operador da câmera.

A estabilização de imagem consiste no processo de remoção de pequenas vibrações que caracterizam movimentos indesejados de uma seqüência de imagens. Com este propósito, atualmente técnicas de processamento digital de vídeo vêm sendo comumente aplicadas na indústria eletrônica. No processo digital de estabilização de imagens são necessários métodos computacionais de estimação, de suavização e de correção de movimento, para os quais, existe uma grande variedade de técnicas de processamento. O emprego de uma técnica específica de processamento é determinado conforme o tipo de aplicação. Técnicas para a estimação de movimento como casamento de blocos (CB), e para a suavização de movimento como filtro de freqüência passa baixa, são freqüentemente encontradas na literatura.

Este trabalho apresenta um sistema de estabilização digital de imagens em tempo real capturadas por uma câmera digital, estimando e compensando movimentos translacionais e rotacionais indesejados.

PALAVRAS CHAVE: Estabilização de imagens em tempo real, estimação de movimento, processamento digital de imagens e compensação de movimento.

ABSTRACT

SILVESTRE, A.C. **Real time digital image stabilization in videos sequences.** 2006. 109 f. Dissertation (Master of Science Degree) – Departamento de Engenharia Elétrica, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006.

Undesirable shakes or jiggles, object motion within image or desirable motions caused by the camera operator causes image differences in consecutive frames of video sequences. The image stabilization consists of the process of removing inevitable and undesirable fluctuations, shakes and jiggles; with this purpose, nowadays digital processing techniques have been commonly applied in the electronic industry. On the digital processing of image stabilization, computational methods of estimation, smoothing and motion correction are necessary. In the literature various digital processing techniques for image stabilization are described, the most suitable technique should be chosen according to the kind of application. Techniques such as "Block matching" used in motion estimation and low-pass filters used in motion smoothing are found in a great number of papers.

This work presents a real time digital image stabilization system capable of stabilizing video sequences with undesirable translational and rotational displacements between frames.

KEY WORDS: Real time image stabilization, digital image processing, motion estimation and smoothing.

Índice de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ITU-T-H.263 – Padrão de codificação de vídeo publicado pela International Telecomunication Union

MPEG – *Moving Picture Expert Group*

OIS – *Optical Image Stabilizer*

EIS – *Eletronic Image Stabilizer*

CB – Casamento de Blocos

VLSI – *Very Large System Integration*

H.261 – Padrão de codificação de vídeo publicado pela *International telecomunication Union*

EMQ – Mínimo erro médio quadrático

MDA – Mínima diferença absoluta

MCP – Máxima combinação por contagem

GC BPM – Gray coded bit plane matching

VFE – Vision Front End

OpenCV – Open source computer vision library

USB – Universal serial bus

RGB – Sistema de cor de imagens (vermelho, verde e azul)

Fps – Frames por segundo

YCrCb – Sistema de cor de imagens (Luminância, Crominância vermelha, Crominância azul)

FPGA – *Field Programable Gate Arrays*

Sumário

Capítulo 1

Introdução

1.1 Motivação

Os movimentos indesejados realizados por uma câmera de vídeo, usando como exemplo uma pessoa que está realizando uma filmagem através de uma câmera portátil ou uma câmera acoplada a um veículo, fazem com que as imagens obtidas apresentem pequenas interferências inevitáveis, como vibrações ou pequenas flutuações, principalmente quando os movimentos com a câmera são rápidos. Esses pequenos deslocamentos presentes na seqüência de imagens são capazes de interferir de forma negativa em processos de codificação e compactação de vídeo e em processos de detecção de movimentos de objetos no campo de ação da câmera. (GUESTRIN, C.; COZMAN, F.; KROTKOV, E., 1998)

Na codificação de vídeo, pequenas vibrações indesejadas presentes na imagem causam aumentos na taxa de bits. O estudo realizado em Engelsberg e Schmidt (1999) analisa a relação entre as vibrações e as variações na taxa de bits de uma codificação ITU-T-H.263, em que um típico cenário de vídeo-fone, com uma pessoa (cabeça e ombro) em primeiro plano, é artificialmente agitado por movimentos de características aproximadamente senoidais e horizontais, com diferentes amplitudes e freqüências.

O comportamento obtido na análise entre a taxa de bits de codificação e a freqüência de vibração, no estudo de Engelsberg e Schmidt é apresentado pela figura

1.1, sendo possível observar um aumento significativo na taxa de bits da codificação, mostrando com isso, a importância da retirada dos deslocamentos indesejados entre os quadros de um vídeo para este processo.

Em processos de Compactação de vídeo, como a codificação MPEG-4, pequenos deslocamentos podem contribuir para a diminuição da taxa de compressão, prejudicando desta forma, a eficiência do processo.

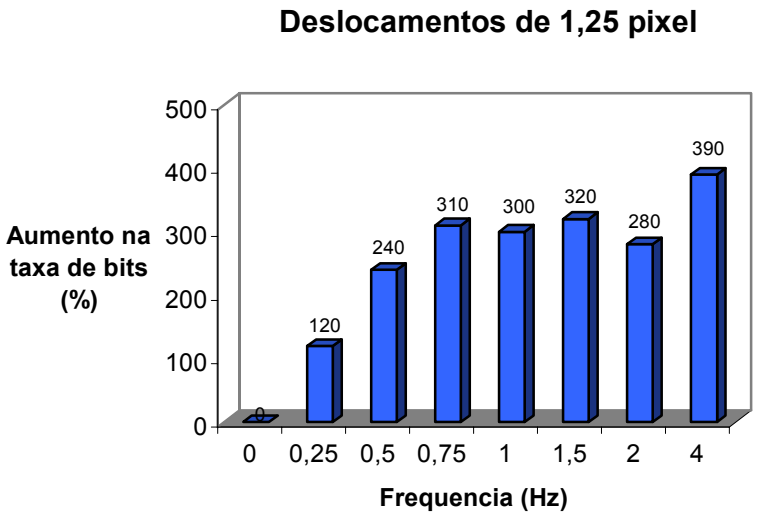


Figura1.1 – Taxa de bits x Freqüência de vibração. (ENGELSBERG, A.; SCHMIDT, G., 1999).

Nos telescópios usados na astronomia são utilizados comumente, técnicas de suavização de movimentos entre as imagens para compensar efeitos de baixa ordem da vista atmosférica e instabilidades locais dos índices de refrações próximos ao telescópio. (TEARE, S.W. et. al., 2004).

Em produtos eletrônicos comerciais, tais como câmeras portáteis de vídeo, telefones celulares de última geração e câmeras digitais, necessitam de mecanismos ou processos capazes de remover as vibrações e flutuações nas imagens obtidas durante a captação do vídeo ou imagem. (CHANG, J.Y. et. al., 2002).

As câmeras compactas comerciais, equipadas com poderosas ações de “zoom”, sofrem grandes problemas com flutuações na imagem. Vários sistemas de compensação de deslocamentos de imagens vêm sendo desenvolvidos para obter câmeras que tirem fotos sem degradações causadas por pequenos movimentos. (KO, S.J. et. al., 1999).

A remoção de movimentos indesejados em seqüências de imagens também possui um importante papel em muitas aplicações de visão computacional, como tele-operações, navegação automática, navegação de robôs, modelagem de cenas, entre outras.

1.2 Objetivos

O objetivo deste trabalho foi o de desenvolver um sistema digital de estabilização de imagens em tempo real, isto é, um sistema capaz de fornecer em tempo real imagens estáveis, originadas de uma seqüência de imagens instáveis, fornecidas por uma câmera em movimento. A estabilidade é encontrada corrigindo a imagem a fim de que se diminua ou anule as variações de movimento rotacional e translacional entre os quadros da seqüência de imagens instáveis de entrada. Técnicas de processamento digital de imagem, como casamento de blocos e filtros de freqüência passa baixa, foram utilizadas para a estimação e suavização de movimento. O sistema desenvolvido procurou ter as seguintes características:

- Operação em tempo real, permitindo a correção da imagem capturada e apresentação da imagem processada;

- Aplicação em diferentes situações de captura de imagens;

- Robustez.

Foi também efetuada uma análise sobre a aplicação em hardware do sistema desenvolvido.

1.3 Organização da monografia.

Esta monografia é constituída por sete capítulos:

- Neste primeiro, foi realizada a introdução ao tema, destacando a importância e aplicabilidade do mesmo. Foram apresentados também os objetivos principais deste trabalho;

- No capítulo 2 são apresentados os diferentes métodos de estabilização de imagens;

- No capitulo 3 é apresentado, detalhadamente, o método de estabilização de imagem digital: a estrutura básica, técnicas de processamento, principais características e exemplos encontrados na literatura;

- No capitulo 4 é apresentado o sistema de estabilização de imagem digital desenvolvido;

- No capitulo 5 são apresentados os testes e resultados obtidos pelo sistema de estabilização desenvolvido;

- No capitulo 6 é apresentada a análise realizada para a implementação em hardware do sistema desenvolvido;

- No capítulo 7 são apresentadas sugestões para trabalhos futuros e algumas considerações finais.

Neste trabalho, há também dois apêndices contendo o programa desenvolvido para a implementação do sistema de estabilização de imagem e informações sobre a biblioteca OpenCV utilizada no projeto.

Capítulo 2

Sistemas de Estabilização de imagens

Neste capítulo são descritos alguns dos diferentes métodos de estabilização de imagens encontrados em câmeras comerciais e em trabalhos na literatura. Os principais métodos utilizados, atualmente, para estabilização de imagens podem ser classificados como: métodos Mecânicos, no qual a estimação e compensação dos movimentos são realizadas por equipamentos mecânicos; métodos Ópticos que realizam o processo de estabilização de imagem através de sensores e sistemas ópticos; métodos Mecânicos – Digitais, em que a estimação dos movimentos é realizada por sensores mecânicos, e ópticos, porém a compensação é realizada de forma digital e métodos Digitais, no qual a estimação e a compensação dos movimentos são realizadas totalmente por processamento digital. (ENGELSBERG, A.; SCHMIDT, G., 1999)

2.1 Métodos Mecânicos

Os métodos de estabilização de imagem totalmente mecânicos são mais encontrados atuando externamente à câmera e são chamados de estabilizadores de câmera, pois buscam estabilizar o equipamento de captura de imagem e não propriamente a imagem. O método mecânico mais simples é conhecido como suporte fixo ou tripé. Sistemas de estabilização de imagens mecânicos mais elaborados, conhecidos como “Steadicam” ou “GlideCam”, usados principalmente em cinematografia, utilizam o centro de gravidade da câmera para a estabilização, isolando-a do operador evitando que as vibrações do mesmo sejam transmitidos a câmera. Além

disso, estes sistemas aumentam a inércia da câmera, deslocando seu centro de gravidade de modo que o operador possa manipulá-la facilmente. A figura 2.1 apresenta um sistema “Steadicam” sendo manipulado por um cinegrafista. (HANDBOOK: SteadiCam Manual, 2001)

Figura 2.1 – Sistema SteadiCam. (HANDBOOK: SteadiCam Manual, 2001).

Os sistemas de estabilização mecânicos de imagem de alto desempenho tendem a ser volumosos e caros.

2.2 Métodos Ópticos

Os métodos ópticos (OIS) são sistemas internos ao equipamento de captura das imagens que manipulam as mesmas antes que cheguem ao sensor da câmera. Microsensores, que têm como princípio de funcionamento o giroscópio (dispositivo capaz de detectar movimento em todas as direções), são colocados próximos às lentes

da câmera para transformar as vibrações em sinais elétricos. Estes sinais elétricos são enviados a um microprocessador capaz de processar os dados recebidos e acionar micromotores situados no conjunto de lentes da câmera. Os micromotores efetuam pequenas alterações nos elementos ópticos, que deslocam a projeção da imagem sobre o elemento sensor diminuindo ou anulando as vibrações da câmera. Métodos ópticos desta configuração são encontrados em diversas câmeras comercias.

Um outro sistema óptico, também empregado em câmeras comerciais, utiliza pequenas placas de vidro, formando um conjunto equivalente a um prisma que, ao se mover, desvia os raios de luz da imagem de entrada. A luz é desviada de forma que, ao passar pelas lentes do sistema, os movimentos indesejados da câmera são compensados. O movimento deste prisma é controlado por um microprocessador que recebe informações dos sensores ópticos de movimento. (BAPTISTA, E., 2000).

Os métodos ópticos de estabilização de imagem apresentam bom desempenho na correção de instabilidades, porém são sistemas altamente sofisticados e de grande custo, sendo encontrados somente em câmeras de uso profissional.

2.3 Métodos Mecânicos – Digitais

No método Mecânico - Digital a estabilização da imagem é realizada através de sensores ópticos e mecânicos e por um processamento digital da imagem. Sensores ópticos são utilizados de forma a detectar a movimentação da câmera capturando e enviando as informações necessárias ao processador digital que estimará e compensará

os movimentos indesejados realizados pela câmera, por meio de um processamento digital da imagem. (ENGELSBERG, A.; SCHMIDT, G., 1999)

2.4 Método Digital

O método Digital, também conhecido como EIS, é altamente utilizado em câmeras comerciais de utilidade amadora e profissional, em processos de codificação e em sistemas de compactação de imagem.

Nesta abordagem, a estimação, suavização e compensação dos movimentos entre os quadros de um vídeo, são realizadas por processamento digital. Na literatura, são encontrados diversos trabalhos que empregam diferentes técnicas de processamento digital para realizar a estabilização da imagem. Alguns dos trabalhos pesquisados serão apresentados e estudados no próximo capítulo.

2.5 Conclusão

Os principais métodos de estabilização de imagens utilizados atualmente podem ser descritos como: Mecânicos, Ópticos, Mecânicos - Digitais e Digitais. O método Digital, devido a sua maior facilidade de emprego em equipamentos, sem a necessidade de acessórios, tem sido muito aplicado em câmeras comerciais, sendo objeto de estudo e discussão em diversos trabalhos.

Já os Métodos Mecânicos, Ópticos e Mecânicos – Digitais são empregados, principalmente, em câmeras de uso profissional utilizadas por cinegrafistas ou

fotógrafos de alto nível técnico. Geralmente, requerem experiência do operador e têm um custo elevado.

Capítulo 3

Estabilização digital de imagens

O objetivo de qualquer processo de estabilização de imagem é retirar de uma seqüência de imagens os movimentos indesejados (vibrações) entre uma imagem e outra. No processo digital de estabilização, toda a informação necessária e as correções estipuladas para a estabilização dos movimentos entre quadros devem ser estimadas através de técnicas de processamento digital.

3.1 Estrutura básica de estabilização de imagens digital

O processo de estabilização de imagem pode ser dividido em três procedimentos básicos de processamento, ilustrados na figura 3.1 e descritos a seguir:

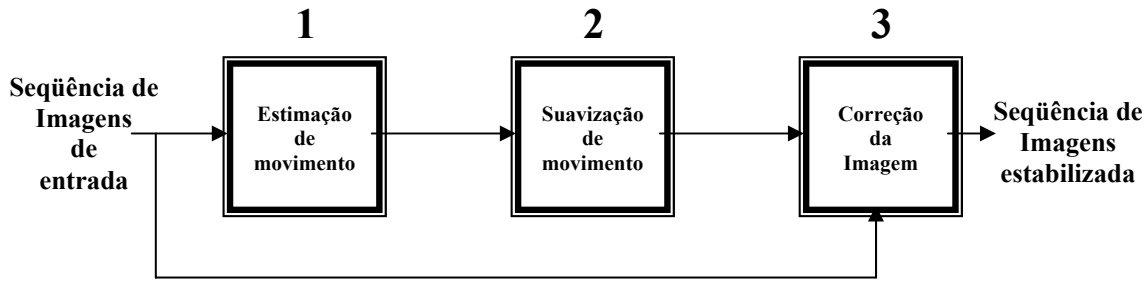


Figura 3.1 – Estrutura básica de estabilização de imagem

1- Estimação de movimento: Cálculo dos vetores de movimento que representam os deslocamentos entre as imagens presentes na seqüência de vídeo;

2- Suavização de movimento: Através dos vetores de movimento estipulados estima a correção necessária, para a suavização dos movimentos indesejados entre as imagens da seqüência de vídeo;

3- Correção de movimento: Com as correções estimadas, atua sobre a imagem original resultando numa imagem estabilizada.

No processo de estabilização para imagens coloridas, frequentemente é utilizado o sistema de cor YCrCb (Luminância, Crominância vermelha, Crominância azul), sendo o canal de Luminância, aplicado ao sistema como imagem de entrada para os processos de estimação e suavização de movimento. Desta forma, todos os cálculos e estimação dos movimentos desejados e indesejados são baseados apenas no canal de Luminância, enquanto a correção de movimento é aplicada a todos os canais da imagem YCrCb.

Para cada bloco de processamento são encontradas, na literatura, diferentes técnicas e abordagens de execução. Em seguida, algumas destas técnicas e abordagens serão descritas.

3.2 Estimação de movimento

Para a estimação de movimento (que determina as variações de movimento rotacional e translacional entre uma imagem e outra) foram desenvolvidas e aplicadas comercialmente diversas técnicas em que o tipo de aplicação é um fator importante na escolha da técnica a ser utilizada.

As principais e mais utilizadas técnicas de estimação de movimento dimensionam os deslocamentos translacionais entre as imagens, através de vetores bidimensionais chamados de vetores de movimento. Movimentos rotacionais são estimados através de modelos de deslocamentos, geralmente formados por um vetor de movimento translacional e uma matriz de rotação. Algumas das técnicas para a estimação do movimento serão detalhadas a seguir.

3.2.1 Movimentos translacionais

3.2.1.1 Casamento de Blocos ("Block Matching")

A técnica Casamento de blocos (CB) pode ser considerada a mais popular para estimação de movimento translacional, principalmente devido a sua baixa complexidade de hardware. Como resultado de sua larga disposição em VLSI, muitos códigos como H.261 e MPEG 1 e 2 utilizam o CB para estimar o movimento entre duas imagens de um vídeo. Neste procedimento, a estimação de movimento é realizada por um processo de busca no domínio do pixel. (TEKALP,A.M. ,1995).

A idéia básica do CB (figura 3.2) é determinar o deslocamento de um pixel de coordenadas (n_1, n_2) da imagem k em relação à imagem k+1, considerando um bloco de tamanho N_1 x N_2 centralizado em (n_1, n_2). A busca de um bloco de mesmo tamanho e melhor combinação, é normalmente limitada para redução de custos computacionais através de uma janela de busca ("search window") M_1 x M_2.

Os algoritmos de CB se diferenciam em três pontos:

- Critério de combinação: máxima correlação cruzada, mínimo erro quadrático, etc;
- Estratégia de busca: busca completa, busca por três passos, busca cruzada, etc;
- Determinação do tamanho de bloco: adaptativo, hierárquico, etc.

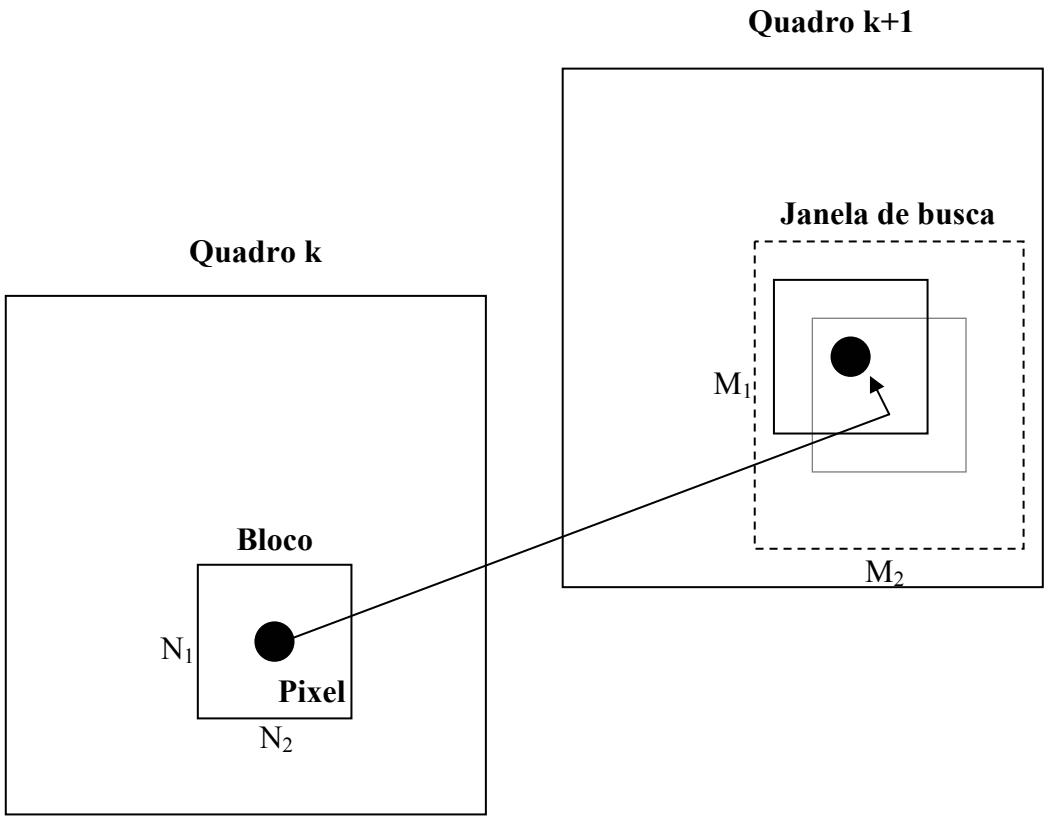


Figura 3.2 – Block Matching

- **Critério de combinação**

A combinação entre os blocos das imagens pode ser quantizada de acordo com vários critérios, incluindo máxima correlação cruzada, minimização do erro médio quadrático (EMQ), minimização da média da diferença absoluta (MDA) e máxima combinação por contagem de pixel (MCP). Os critérios EMQ e MDA serão estudados em seguida.

O critério de EMQ é definido pela equação (1).

$$\mathrm{EMQ}(d_1,d_2)=\frac{1}{N_1N_2}\sum_{(n_1,n_2)\in B}[s(n_1,n_2,k)-s(n_1+d_1,n_2+d_2,k+1)]^2 \qquad (1)$$

Na qual *B* denota o bloco N_1 x N_2 para o conjunto de possíveis vetores de movimentos (d_1, d_2) do bloco, K representa o índice da imagem e *S* é o valor em nível de cinzas do pixel. O vetor de movimento estimado é obtido pelo par (d_1, d_2) que minimiza o EMQ, equação (2).

$$[d_1,d_2]^T=\arg\min_{(d_1,d_2)} EMQ(d_1,d_2) \qquad (2)$$

O critério de minimização do EMQ pode ser visto como a imposição do fluxo óptico para todos os pixeis do bloco, sendo o fluxo óptico definido como um campo vetorial de velocidades que associa cada pixel a um único vetor de velocidade. (TEKALP,A.M. ,1995)

O critério de minimização MDA pode ser definido na equação (3):

$$MDA(d_1,d_2)=\frac{1}{N_1N_2}\sum_{(n_1,n_2)\in B}[s(n_1,n_2,k)-s(n_1+d_1,n_2+d_2,k+1)] \qquad (3)$$

Este critério é o mais utilizado para aplicações em VLSI. A estimação de deslocamentos é dada pelo par (d_1, d_2) que minimiza a equação (3).

$$[d_1, d_2]^T = \arg\min_{(d_1,d_2)} MDA(d_1, d_2) \quad (4)$$

A minimização MDA apresenta baixo desempenho em casos de grandes janelas de busca, devido à presença de vários mínimos locais. (TEKALP,A.M. ,1995).

Este critério apresenta uma maior sensibilidade às variações de luminosidades ou mudanças profundas nas imagens, resultando, nestes casos, em grandes perturbações e erros de estimação dos vetores de movimento nestes casos.

Dos critérios estudados, pode-se observar que o critério EMQ apresenta operações mais complexas em relação ao critério MDA e, consequentemente, possui uma implementação em hardware mais complexa.

- **Estratégia de busca**

Encontrar o melhor bloco para a estimação de movimento requer otimizar o critério de combinação para um possível vetor de deslocamento em cada pixel (n_1, n_2). A estratégia de busca, conhecida como busca completa, calcula o critério de combinação para todas as distâncias da janela de busca, isto é, o aplica a todos os pixeis da área de busca. Este tipo de busca resulta num grande consumo de tempo e de memória computacional.

Uma estratégia comum, utilizada para reduzir o custo computacional, é estimar os vetores de movimento para pixeis espaçados (geralmente de oito pixeis), usando um bloco de 16x16 pixeis. (TEKALP,A.M. ,1995)

No entanto, em alguns casos, estratégias de busca mais rápidas e com menor custo computacional são necessárias. Desta forma, estratégias como busca por três passos e busca cruzada são utilizadas.

A figura 3.3 realiza um processo de busca, utilizando a estratégia dos três passos. Uma janela de busca de M_1 x M_2, na qual $M_1 = M_2 = 15$ é utilizada. Para o tamanho do bloco podemos considerar $N_1 = N_2 = 1$. O pixel marcado com "0" indica o pixel da imagem de busca (referência) relacionado ao pixel da imagem atual. No primeiro passo o critério de combinação é aplicado em nove pixeis, o pixel "0" e os pixeis "1". Se o menor critério for encontrado para o pixel "0", não há movimento entre as imagens. No segundo passo, o critério de combinação é aplicado a oito pixeis, os pixeis marcados com "2" e o pixel marcado com "1" (em negrito), sendo este o escolhido como melhor combinação pelo passo anterior. O vetor de movimento é encontrado no terceiro passo, em que os pixeis utilizados para a busca estão na vizinhança de 8 conectados. Portanto, ao aplicar o critério de combinação para cada pixel desta janela, o processo será finalizado. (TEKALP,A.M. ,1995)

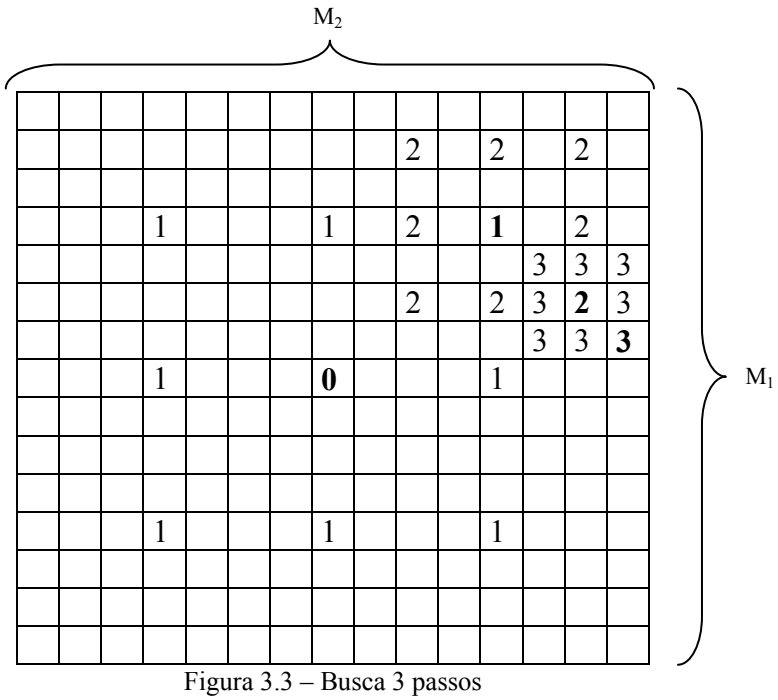


Figura 3.3 – Busca 3 passos

Ao generalizarmos este procedimento a outros tamanhos de janela de busca, serão realizados passos a mais ou a menos. Estes procedimentos, que utilizam diferentes tipos de janela de busca, recebem o nome de n passos ou log-D.

A busca cruzada (figura 3.4) é uma outra estratégia rápida de estimação de movimento. Nela a busca é realizada em quatro locais (pixeis) por etapa. Os pixeis são determinados pela figura de uma cruz em "×" ou "+" pré-estabelecida, em que o centro da figura representa o pixel inicial da busca ou o escolhido no passo anterior e as extremidades, os outros locais de busca.

A distância entre os pixeis de busca, isto é, das pontas da "cruz", é reduzida se o pixel central apresenta o melhor critério de combinação ou se estamos trabalhando na borda da janela de busca. Muitas variações desta estratégia de busca são encontradas na literatura.

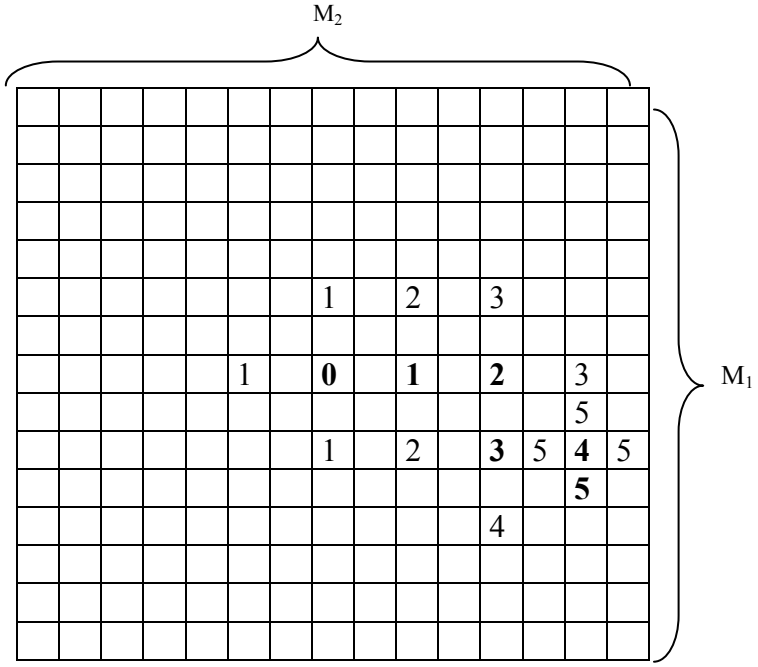


Figura 3.4 – Busca cruzada

O CB também pode ser generalizado para figuras de janela de busca com formas variadas, tais como figuras curvilíneas, utilizando diferentes estratégias de busca heurísticas.

- **Determinação do tamanho do bloco de busca**

A seleção do tamanho do bloco de busca (N_1, N_2) é essencial para qualquer algoritmo de estimação CB. Se o bloco for muito pequeno, a combinação pode ser estabelecida devido à similaridade de níveis de cinza entre os blocos e não com relação ao seu movimento. Por outro lado, para grandes blocos de busca, o vetor de movimento pode variar dentro do bloco, violando a hipótese de apenas um vetor por bloco. (TEKALP, A.M., 1995)

Com o objetivo de contornar este problema, estratégias com estimativa hierárquica do movimento foram desenvolvidas.

A estratégia de busca hierárquica representa a imagem na forma de pirâmide pela transformada de Laplace ou de Wavelet dividindo-a em vários níveis de resolução, e pode ser usada nos métodos estimação de movimento correlação de fase e CB, com a finalidade de aumentar o desempenho de estimação dos vetores de movimento.

A idéia básica da estratégia de busca hierárquica é aplicar a estimação de movimento para cada nível da pirâmide de Laplace ou Wavelet, sucessivamente, iniciando do nível de menor resolução. Os níveis de baixa resolução determinam uma estimação grosseira dos deslocamentos, usando blocos de grande tamanho. Observa-se que, o “tamanho relativo do bloco” pode ser determinado relacionando-se o tamanho de

bloco normalizado com o tamanho da imagem num nível particular de resolução (TEKALP, A.M., 1995). A estimação do vetor de movimento dos níveis de baixa resolução é então passada aos níveis de resolução maior como uma estimação inicial. Assim, os níveis de alta resolução trabalham como um ajuste fino da estimação de movimento, utilizando tamanhos de blocos menores.

Apesar dos diferentes critérios de combinação, estratégias de busca e variações da técnica de estimação CB, o desempenho deste método é comprometido em casos de grandes movimentos rotacionais, profundas mudanças nas imagens e borramento das imagens, podendo apresentar erros na estimação dos vetores de movimento.

3.2.1.2 Correlação de Fase para movimentos translacionais (Phase– Correlation)

O método de correlação de fase estima o deslocamento entre dois blocos de imagem pela normalização da função de correlação cruzada computada, no domínio da freqüência (transformada de Fourier 2-D). Baseia-se no princípio de que deslocamentos no domínio do espaço resultam em termos de fase linear no domínio de Fourier. (TEKALP,A.M., 1995)

A função de correlação cruzada pode ser definida na equação (5),

$$c_{k,k+1}(n_1, n_2) = s(n_1, n_2, k+1) ** s(-n_1, -n_2, k) \tag{5}$$

em que, ** denota a operação de convolução 2-D.

Aplicando a transformada de Fourier para ambos os lados e normalizando a expressão obtida temos.

$$\widetilde{C}_{k,k+1}(f_1, f_2) = \frac{S_{k+1}(f_1, f_2) S_k^{*}(f_1, f_2)}{\left| S_{k+1}(f_1, f_2) S_k^{*}(f_1, f_2) \right|} \tag{6}$$

Utilizando o modelo discreto de movimento, definido na equação (7), sua transformada de Fourier 2-D na equação (8) e assumindo somente movimentos translacionais, podemos reescrever a equação (6) na equação (9):

$$s(n_1, n_2, k) = s(n_1 + d_1, n_2 + d_2, k + 1) \tag{7}$$

$$S_k(f_1, f_2) = S_{k+1}(f_1, f_2) \exp\{j2\pi(d_1 f_1 + d_2 f_2)\} \tag{8}$$

$$\widetilde{C}_{k,k+1}(f_1, f_2) = \exp\{-j2\pi(f_1 d_1 + f_2 d_2)\} \tag{9}$$

Obtendo a transformada inversa, temos a função de correlação de fase.

$$\widetilde{c}_{k,k+1}(n_1, n_2) = \delta(n_1 - d_1, n_2 - d_2) \tag{10}$$

Observe que, a função de correlação de fase consiste num impulso em que sua localização, dada pelo par (d_1, d_2), gera o vetor de movimento.

O emprego deste método requer substituir a transformada 2-D de Fourier pela transformada discreta 2-D de Fourier (DFT 2-D). Na prática, a função de correlação de fase obtida apresenta vários picos principalmente devido a discretização empregada pela DFT – 2D, a presença de objetos se movendo dentro do bloco e a presença de ruídos. O vetor de movimento pode ser encontrado verificando-se a magnitude de cada pico da função de correlação. (TEKALP, A.M., 1995)

A correlação de fase é relativamente insensível à mudanças na iluminação, pois deslocamentos no valor médio ou multiplicação por constantes não alteram a fase de Fourier.

Os métodos de estimação dos vetores de movimento CB e Correlação de Fase, apresentados acima, determinam um vetor bidimensional para cada pixel ou bloco no qual foi realizada a busca. Cada vetor de movimento é chamado de vetor de movimento local, pois representa o deslocamento de um determinado pixel ou bloco. Para estimar o vetor de movimento global, que represente o deslocamento geral entre as imagens, são empregadas técnicas como o cálculo da média ou integração dos vetores de movimento local.

3.2.2 Movimento rotacionais

Para estimar ou modelar os movimentos rotacionais entre as imagens de um vídeo, outros modelos ou técnicas de estimação de movimento precisam ser empregadas separadamente ou em conjunto com a estimação de movimentos translacionais.

Em Chang (2002), os movimentos rotacionais são representados em um modelo dinâmico de movimento, cujos parâmetros são obtidos através das velocidades estimadas de cada pixel. Para estimar as velocidades, emprega-se a técnica de fluxo óptico.

Um modelo comumente empregado para a estimação e correção de movimentos rotacionais em sistemas de estabilização de imagem é o modelo “Affine”, apresentado abaixo na equação (11).

$$\begin{bmatrix} X \\ Y \end{bmatrix} = S\begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \tag{11}$$

$$\alpha = \cos(\theta),\ \beta = sin(\theta),$$

$$S = \frac{\sqrt{\Delta x_2 + \Delta y_2}}{\sqrt{\Delta x_1 + \Delta y_1}}\ .$$

O termo *S* (fator de escala) possibilita que variações de escala entre as imagens possam também ser modeladas e corrigidas por este modelo. Δx_1, Δx_2, Δy_1, Δy_2 são as diferenças entre as coordenadas de dois pontos quaisquer, entre duas imagens. θ é o ângulo de rotação entre as imagens.

Em Chen (2000), o modelo “Affine” é empregado em conjunto com a técnica de estimação de movimento CB.

Morimoto e Chellapa (1996) utilizam pontos de referência para a estimação dos parâmetros do modelo de movimento “Affine”. Comumente, aplicam-se regras heurísticas para a seleção dos pontos de referência em imagem de campo aberto, sendo o horizonte freqüentemente empregado.

3.3 Suavização de movimento

Os algoritmos de estabilização de imagens necessitam diferenciar os movimentos desejados realizado pela câmera, daqueles que são indesejados. Na literatura são encontradas diversas propostas.

Em Chen (2000), assume-se que movimentos intencionais caracterizam-se por apresentarem variações suaves entre uma imagem e outra, de um vídeo. Assim, componentes de altas freqüências na variação dos vetores de movimento são consideradas efeitos de movimentos indesejados da câmera. Portanto, filtrando-se estas componentes, tem-se a distinção entre os movimentos desejados e indesejados.

É comum a utilização de um filtro de freqüência passa baixa digital, como filtro Gaussiano ou média de movimentos ("Moving Average"), permitindo, desta forma, somente a passagem das componentes de baixa freqüência da variação dos vetores de movimento.

Outros sistemas de estabilização de imagem, como descrito em Ko et. al.(1999), utilizam processos de integração para permitir que somente os movimentos indesejados estimados entre as imagens sejam corrigidos. Os vetores de movimento local são integrados com um coeficiente de amortecimento D_1. O vetor resultante da integração é utilizado como vetor de correção do movimento da imagem. O coeficiente de amortecimento tem seu valor no intervalo de $0<D_1<1$. Sua função é realizar a convergência do vetor de movimento global a zero, quando não há movimento da câmera.

Uma outra possível abordagem na determinação de movimentos desejados e indesejados é a construção de um modelo cinemático, no qual os parâmetros cinéticos do suporte da câmera e dos movimentos em que a câmera está sujeita sejam conhecidos. Como por exemplo, no caso de uma câmera instalada num veículo aos quais os parâmetros como, massa do veículo, características de suspensão, distâncias entre os eixos e centro de gravidade sejam possíveis de serem calculados.

O modelo construído a partir destes parâmetros é utilizado para separar os movimentos desejados das oscilações realizadas pelo veículo. Entretanto na maioria das aplicações não é possível o conhecimento de todos os parâmetros, o que torna esta abordagem freqüentemente inviável. Porém, um modelo genérico de movimento inercial utilizado como filtro para as vibrações de alta freqüência indesejadas é proposto em ZHU, Z. et. al. (1998). Este filtro combina métodos físicos e tem pequenos deslocamentos de fase numa grande faixa de freqüência, e necessitando somente de um conjunto de suposições da dinâmica de movimento. (ZHU, Z. et al., 1998).

Em Litvin et. al., é utilizado um filtro de Kalman (filtro adaptativo), para a detecção dos movimentos intencionais realizados pela câmera.

3.4 Correção de movimento

Após a estimação de movimento e a determinação dos deslocamentos desejados e indesejados, torna-se necessário empregar métodos de correção sobre as imagens.

A partir dos vetores de movimento desejados e indesejados é possível corrigir os movimentos, imagem a imagem, deslocando-as de maneira contrária aos vetores de movimento indesejados estimados. Um procedimento parecido a este é empregado em Chen (2000), em que a imagem estabilizada é obtida empregando-se o vetor de movimento suavizado (após filtragem passa baixa) sobre a imagem estabilizada ou a original anterior. A escolha entre utilizar a imagem anterior estabilizada ou original é estabelecida na estimação do erro de compensação propagado entre as imagens consecutivas. Neste caso, um quadro de referência (original ou estabilizado anterior) é escolhido num primeiro momento, em seguida, os quadros sucessivos são alinhados em relação a este através de um deslocamento baseado nos vetores de movimento estimados. Esta abordagem funciona satisfatoriamente se não houver mudanças profundas no campo de visão e/ou movimentos de grandes distâncias da câmera, pois cenas fora do campo de visão do quadro de referência não serão mostradas na imagem estabilizada, causando problemas nas bordas das mesmas.

Uma possível abordagem deste problema, freqüentemente utilizada em câmeras digitais comerciais, é a restrição do campo de visão para uma área menor que o sensor de luz CCD. Com isso, é possível estabilizar as imagens com as informações das bordas não presentes na imagem de referência.

3.5 Exemplos de estabilização digital de imagem

Na literatura são encontradas diversas propostas de sistemas de estabilização digital de imagem.

Em Hansen et. al. (1994), é apresentado um sistema de estabilização de imagem em tempo real baseado na construção de um mosaico de imagens. Utilizando um hardware específico de processamento de imagem chamado “Vision Front End” (VFE), este sistema busca estimar e corrigir movimentos de primeira ordem (modelo “Affine”) presentes entre sucessivas imagens de um vídeo, alinhando-as num sistema de coordenadas de referência e inserindo-as no mosaico de imagem na posição apropriada.

Para o alinhamento das imagens, uma transformação baseada no modelo “Affine” é aplicada, transportando a imagem atual para o sistema de coordenadas da imagem de referência. Um processo iterativo de multi resolução, operando em modo de baixa para alta resolução sobre a pirâmide de Laplace, é utilizado para estimar os parâmetros de movimento do modelo. Durante cada iteração, um campo de fluxo óptico é estimado entre as imagens através da análise de uma correlação cruzada local e um modelo de movimento “Affine” é gerado para o campo de fluxo óptico encontrado.

A representação por mosaico apresenta um melhor desempenho comparado a um simples alinhamento entre as imagens, já que o campo de visão é expandido, não se restringindo apenas ao da imagem de referência.

Uma técnica de estabilização baseada na codificação binária de níveis de cinza de uma imagem é apresentada em Ko et al. (1999). A estimação de movimento proposta realiza uma rápida combinação binária, usando a codificação de níveis de cinza dos planos de bit de uma imagem GC-BPM para determinar os vetores de movimento locais de quatro subimagens (figura 3.5), alocadas de forma apropriada na imagem em níveis de cinza. Cada vetor de movimento das subimagens é determinado pela medida da

correlação entre o GC-BPM das subimagens da imagem anterior e atual. Esta abordagem assume que todos os pixeis dentro das subimagens tenham movimento uniforme e que a faixa de movimento seja restrita a janela de busca. O vetor de movimento global do frame é encontrado através da média entre cada vetor de movimento local das subimagens e do vetor de movimento global anterior.

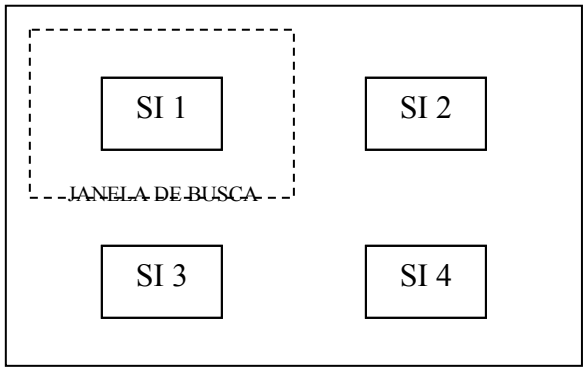


Figura 3.5 – Quatro subimagens

Após a estimação do vetor global de movimento, o sistema decide qual movimento da imagem é causado de forma intencional ou indesejada pela câmera. Com este objetivo, um processo de integração dos vetores de movimento estimados com coeficiente de amortecimento D_1 ($0<D_1<1$), é utilizado.

Em Engelsberg e Schmidt (1999) é apresentada uma técnica de estabilização de imagem cuja idéia básica é aumentar a imagem de entrada de forma a obtê-la com o maior tamanho possível e cortá-la em um menor, resultando em uma imagem com pouca ou nenhuma vibração indesejada. O processo de corte é controlado com base em uma imagem de referência pré-estabelecida. Uma visão geral do sistema é apresentada no diagrama de blocos da figura 3.6 (p. 30).

A imagem de entrada passa, basicamente, por 3 blocos:

1- Bloco de Memória: armazena as imagens recebidas para que as imagens anteriores sejam fornecidas ao bloco de estimação de movimento;

2- Bloco de Zoom: Aumenta a imagem de entrada que futuramente será recortada;

3- Bloco Estimação de movimento: utiliza as imagens passadas e presentes para estimar o vetor de movimento.

Para realizar a estimação de movimento, são escolhidas 3 regiões da imagem, determinadas com base no tipo de cenário da aplicação. Duas regiões são alocadas para estimar os movimentos relacionados ao plano de fundo (“background”) e uma região, para primeiro plano (“foreground”) da imagem. A técnica CB com pontos de referência é utilizada para a estimação de movimento de cada região.

Os vetores de movimento estimados são enviados ao bloco de validação, que calcula um vetor de movimento global para a imagem de entrada. Cada vetor de movimento de entrada é validado e testado para a estabilização do plano de fundo e de frente. A decisão sobre o plano de estabilização é realizada de acordo com o critério de detecção de movimento que, se baseia na análise dos movimentos dentro das regiões.

Com o objetivo de realizar a estabilização na seqüência de imagens, este sistema necessita de uma imagem de referência virtual, sobre a qual, a verdadeira correção será realizada. Esta propriedade é determinada pelo bloco de integração, que realiza uma integração entre os vetores de movimento globais passados e presente, ponderados por um fator de amortecimento. (ENGELSBERG, A.; SCHMIDT, G., 1999).

O último passo é realizado pelo bloco de recorte, no qual através do controle do vetor de movimento integrado, uma subimagem pertinente é recortada da imagem aumentada.

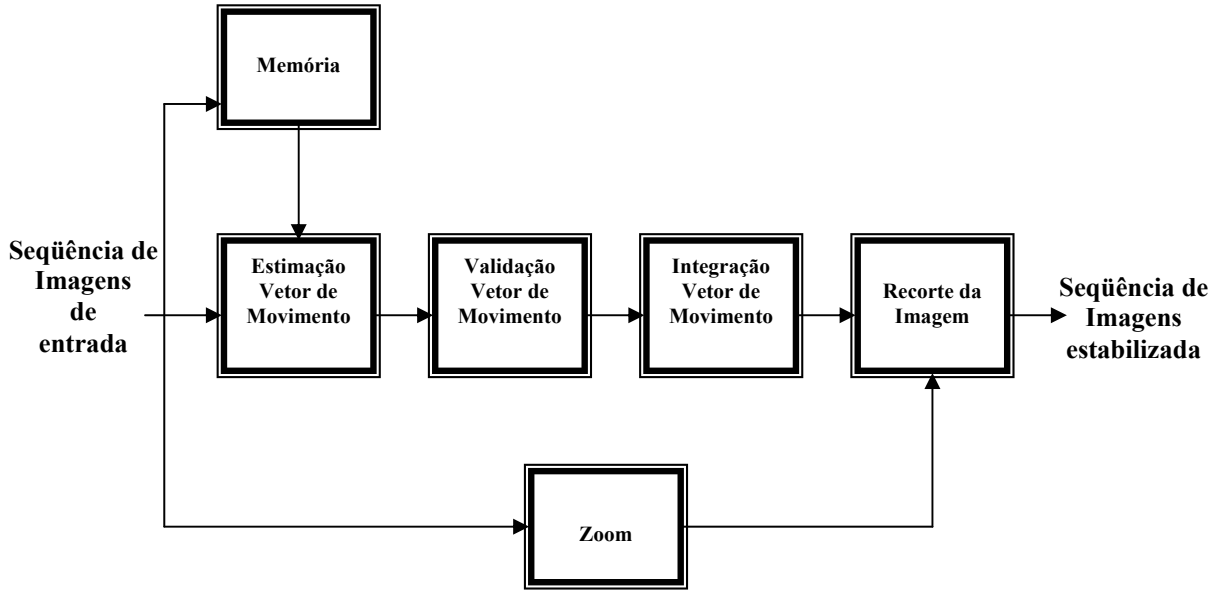


Figura 3.6 - Sistema de estabilização apresentado em Engelsberg e Schmidt (1999).

Chang (2002) apresenta uma abordagem de estabilização de imagem para movimentos translacionais e rotacionais, em que, a técnica de fluxo óptico é aplicada com o objetivo de determinar um vetor local de velocidade para cada pixel da imagem atual. Estes vetores de velocidade são usados com a finalidade de determinar o

movimento global de translação e rotação em termos de um modelo dinâmico de movimento, cujos parâmetros são encontrados por uma estimação de mínimos quadrados.

Em Chen (2000), a estimação de movimento é obtida através da técnica de estimação do CB busca completa. Esta abordagem trata movimentos translacionais e rotacionais separadamente, realizando uma análise dos vetores de movimentos encontrados. Um estudo apresentado neste artigo mostra que o conjunto dos vetores de movimentos estimados pelo CB, apresenta uma variância maior para movimentos rotacionais do que para movimentos translacionais. O modelo "Affine" é utilizado para a suavização e compensação dos movimentos rotacionais. São empregadas também, técnicas de estimação de erros baseadas na estimação do deslocamento, buscando uma melhoria no desempenho do sistema e na decisão sobre a imagem de referência.

3.6 Conclusão

Atualmente na literatura são encontrados diversos sistemas digitais de estabilização de imagens que, em sua maioria, mantém a estrutura básica (estimação, suavização e compensação de movimento), se diferenciando, principalmente, nas técnicas de processamento digital empregadas.

Em seguida, são exibidas as características principais dos sistemas estudados:

HANSEN et. al. (1994)

- Estabilização de imagem em tempo real;

- Correção de movimentos rotacionais e translacionais;
- Construção do modelo Affine;
- Representação por mosaico de imagens;
- Utilização de hardware específico para processamento de imagem.

Ko et. al. (1999)

- Estabilização de imagem por tempo real;
- Correção de movimentos translacionais;
- Estimação de movimento por codificação binária de níveis de cinza e busca por três passos.

Engelsberg e Schimidt (1999)

- Correção de movimentos translacionais;
- Estimação de movimento por casamento de blocos com pontos de referência;
- Correção de movimento através de cortes realizados na imagem original aumentada.

Chang (2002)

- Estimação de movimento através do cálculo do fluxo óptico;
- Correção de movimentos rotacionais e translacionais.

Chen (2000)

- Estimação de movimentos por casamento de blocos com busca completa;
- Construção do modelo Affine;
- Correção de movimentos translacionais e rotacionais.

Há um grande número de técnicas e estratégias utilizadas para o processo de estabilização de imagem. Entretanto, grande parte dos sistemas estudados, contendo as características do sistema desenvolvido neste projeto, isto é, operação em tempo real e correção de movimentos translacionais e rotacionais, necessitam de ferramentas em hardware específicas para processamento de imagem, capazes de realizar as operações com grande velocidade. Deste modo, durante o desenvolvimento deste trabalho buscou-se utilizar técnicas de processamento de vídeo e imagem, de baixo custo computacional e uma linguagem de programação rápida.

Capítulo 4

Sistema de Estabilização de Imagem Digital

O sistema digital de estabilização de imagens desenvolvido neste trabalho tem como objetivo principal, um bom desempenho em tempo real da estabilização de imagens, realizando correções de movimentos translacionais e rotacionais e buscando uma máxima praticidade na aplicação em hardware. Sua estrutura é baseada nos sistemas de estabilização de imagem discutidos anteriormente. Porém, técnicas de estimação de erro e detecção de movimento de objetos dentro da imagem foram empregadas na escolha do quadro de referência e na decisão de compensação de movimento respectivamente.

4.1 Estimação dos vetores de movimento

As características de trabalho e de aplicação do sistema desenvolvido levam à abordagem de dois pontos importantes quanto à escolha da técnica de estimação dos vetores de movimento:

- Custo computacional (memória computacional e tempo);
- Praticidade na implementação em hardware.

O sistema desenvolvido teve sua aplicação avaliada em hardware, portanto, a escolha de uma técnica de estimação de movimento de fácil emprego neste tipo de abordagem é importante.

Outra característica importante do sistema é a sua operabilidade em tempo real. O sistema deve estimar, suavizar e corrigir os movimentos das imagens de entrada sem prejudicar a velocidade de visualização das imagens originais e estabilizadas. Com isto o processo de estimação de movimento deve possuir baixo custo computacional e, conseqüentemente, apresentar um rápido processamento.

Visando as propriedades apresentadas e o tamanho escolhido do bloco de pixeis e da janela de busca, uma generalização do processo CB com estratégia de busca por três passos, chamado de CB n passos, foi utilizada.

A imagem de entrada (canal de Luminância) foi divida em blocos de 16 x 16 e uma janela de busca, também de 16 x 16, foi empregada. Como critério de combinação, primeiramente utilizou-se a minimização da média diferença absoluta (MDA), devido a sua simplicidade e facilidade de aplicação, porém, foi apresentada na simulação do sistema com o critério MDA, uma grande sensibilidade do sistema à variações de luminosidade, movimentos de objetos e deformações nas imagens. Após vários testes, escolheu-se o critério de minimização do erro médio quadrático (EMQ), já que este método apresentou um melhor desempenho nas simulações quanto à variação de luminosidade e deformações na imagem, embora tenha uma maior complexidade na implementação em hardware como discutido no capitulo dois.

4.1.1 Vetor de movimento global

Para realizar o processo de estabilização, devemos encontrar um vetor bidimensional que descreva o deslocamento geral de uma imagem em relação à outra. Aplicando-se a técnica CB n passos, encontram-se inicialmente, os vetores de movimento local ilustrados na figura 4.1, estimando-se o movimento para cada bloco 16 x 16. O vetor de estimação global de movimento (U, V) é calculado baseado nos vetores de movimento local estimados (u, v), através da equação (12):

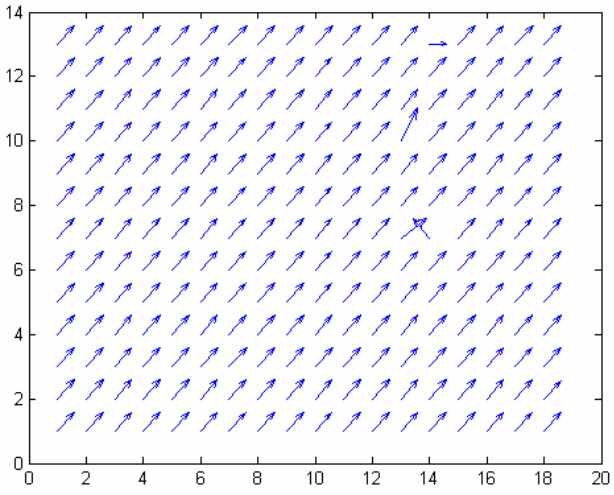
$$MV_{global} = (U, V)$$
$$U = \frac{1}{N}\sum_{i=1}^{N} u_i \qquad (12)$$
$$V = \frac{1}{N}\sum_{i=1}^{N} v_i$$

N = número de vetores locais para cada imagem.

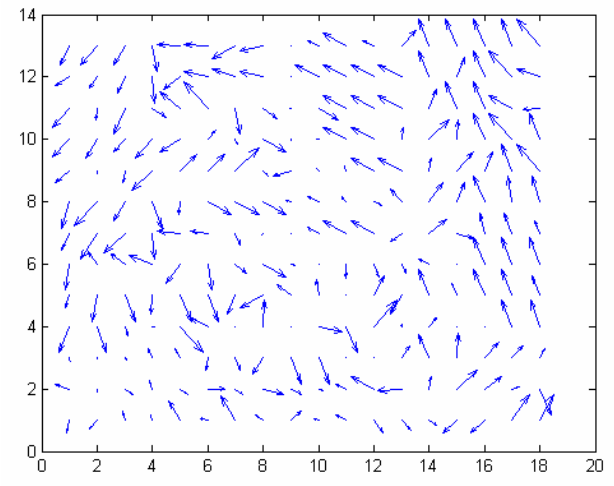
4.1.2 Movimentos rotacionais e translacionais

Os vetores de movimento estimados representam somente movimentos nas direções verticais e horizontais, pois a técnica casamento de blocos (CB n passos) empregada trata somente movimentos translacionais. Entretanto, em Chen (2000), um processo de classificação do movimento em rotacional e translacional, através dos vetores locais estimados pela CB, foi apresentado. Comparando-se a distribuição dos vetores de movimento translacionais e rotacionais, é possível classificar uma determinada distribuição de vetores em movimento de características translacionais ou movimento de características rotacionais.

Uma típica distribuição dos vetores de movimento estimados pelo CB n passos para movimentos rotacionais e translacionais é apresentada na figura 4.1.



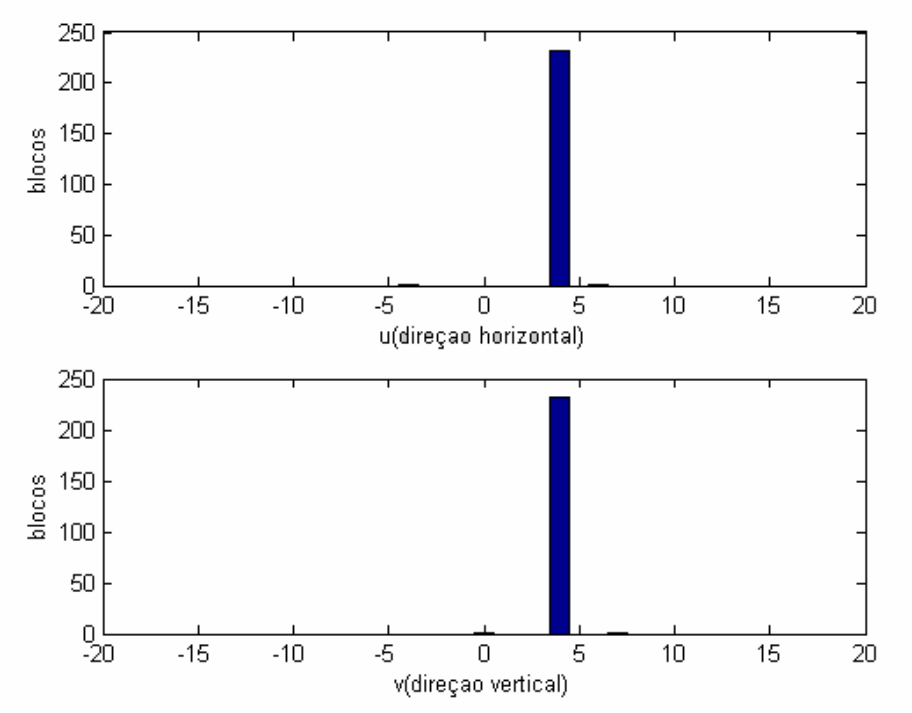
(a)



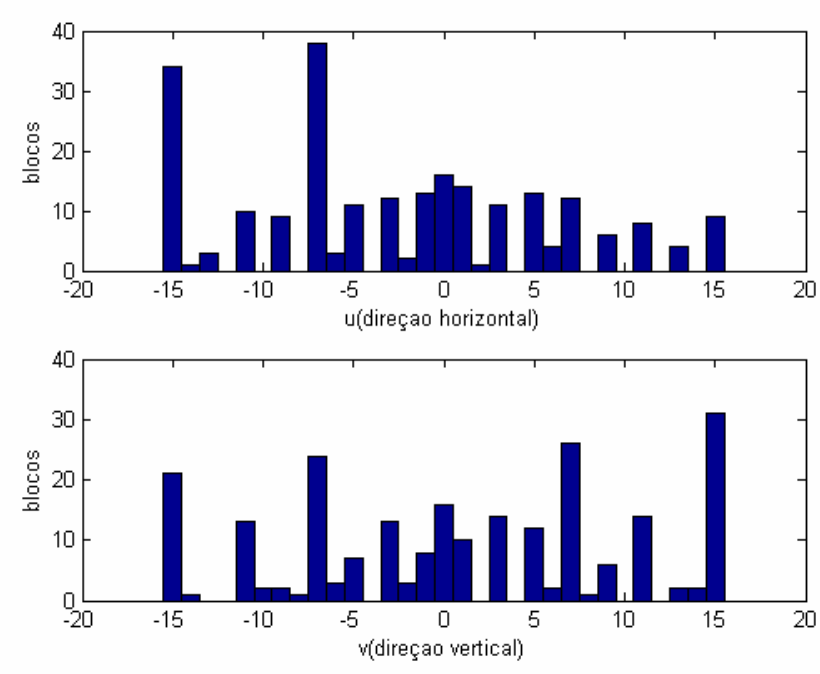
(b)

Figura 4.1 – (a) campo vetorial de movimento translacional, (b) campo vetorial de movimento rotacional

Nos histogramas de cada campo vetorial de movimento, figura 4.2, nota-se variações de direção mais significativas para movimentos rotacionais.



(a)

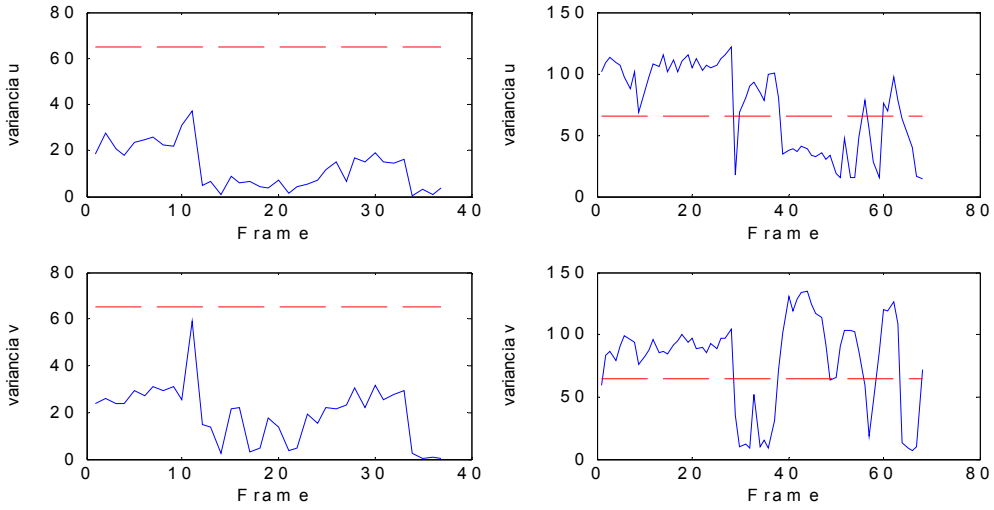


(b)

Figura 4.2 – (a) histograma para movimento translacional, (b) histograma para movimento rotacional

Através do cálculo da variância da distribuição dos vetores de movimento é possível estipular um limiar de classificação entre movimentos de características

rotacionais e translacionais. A figura 4.3 apresenta um gráfico da variância da distribuição dos vetores de movimento para um vídeo capturado por uma "WebCam" realizando movimentos puramente translacionais (a) e rotacionais(b) e o limiar de classificação empiricamente estipulado no valor de 65.



(a) (b)

Figura 4.3 – (a) variância das imagens com movimento translacional, (b) variância das imagens com movimento rotacional

4.1.2 Modelo "Affine"

Se, após a classificação do movimento entre as imagens, um movimento rotacional é determinado, o sistema necessita criar um modelo capaz de estimar e corrigir rotações entre as imagens. Com este objetivo um modelo "Affine" é obtido. Neste trabalho, variações de escala entre as imagens não serão consideradas.

Para determinar os parâmetros do modelo "Affine" de movimento, foram utilizados pontos correspondentes entre as imagens. Pontos centrais de cada bloco de

estimação de movimento foram selecionados. Deste modo, torna-se importante uma maior precisão da estimação dos vetores locais para os pontos escolhidos. Com este objetivo, uma ponderação que prioriza os pontos centrais em relação aos pontos da borda dos blocos é efetuada na estimação dos vetores de movimento local. Este processo foi utilizado em Chen (2000), no qual a busca ponderada foi chamada de ponderação do casamento de blocos ("Block Matching Weight"). A ponderação aplicada ao bloco de busca do sistema desenvolvido é apresentada na figura 4.4 e é descrita pela fórmula da gaussiana 2D (equação 13).

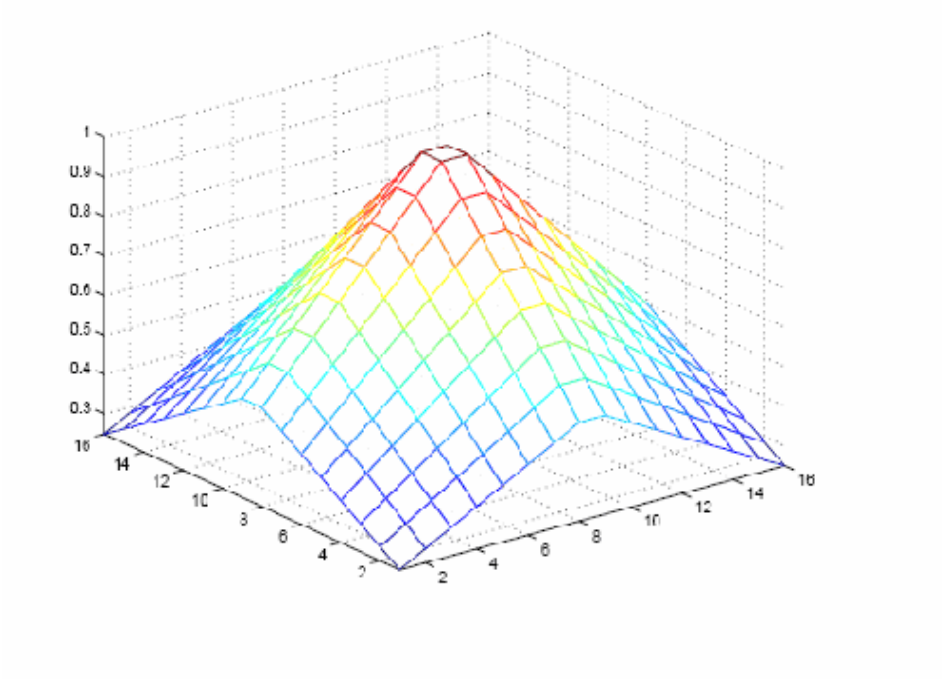


Figura 4.4 – Ponderação na estimação de movimento global. (Chen, 2000)

$$G(x,y) = e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \qquad (13)$$

Onde σ é o parâmetro que define a largura da função gaussiana.

Na sua aplicação ao sistema desenvolvido, esta ponderação foi utilizada de forma matricial, isto é, aplicou-se uma matriz 16x16 a cada bloco do processo de estimação de movimento.

Utilizando os vetores de movimento estimados para os pontos centrais dos blocos pode-se reescrever o modelo "Affine" na equação (14):

$$\begin{bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ \vdots \\ X_n \\ Y_n \end{bmatrix} = \begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \Delta x \\ \Delta y \end{bmatrix}, \tag{14}$$

$$X_i = x_i + u_i,$$
$$Y_i = y_i + v_i$$

A equação descreve um sistema de 2N equações e quatro incógnitas, em que os parâmetros do modelo podem ser encontrados aplicando-se uma resolução por mínimos quadrados.

4.1.3 Imagem de referência

Os vetores de movimento estimados para uma imagem dimensionam o deslocamento desta em relação a uma imagem anterior do vídeo chamada de imagem de

referência, portanto, a seleção correta da referência é de grande importância e atua diretamente no desempenho do sistema.

A escolha da imagem de referência no sistema desenvolvido foi realizada através da análise do vetor de movimento global estimado e pelo número de imagens que distanciam a imagem atual da de referência. Num estado inicial, a primeira imagem capturada torna-se de referência para os quadros seguintes. Nas próximas passagens, a distância euclidiana das imagens é calculada com base nos vetores de movimento global. Utilizando-se um limiar, classifica-se o deslocamento como pequeno ou grande.

Se um pequeno deslocamento é identificado, a imagem estabilizada passa a ser utilizada como referência, sendo adotada para estimar o deslocamento da imagem seguinte. Por outro lado, se um movimento de grande deslocamento ocorre, a imagem de referência anterior é mantida, evitando-se uma propagação de erro no sistema de estabilização, já que a imagem atual estabilizada, neste caso, apresenta uma grande diferença de campo de visão, podendo, desta forma, apresentar deformações consideradas nas suas bordas.

Além do deslocamento entre as imagens, a distância no tempo entre a imagem de referência e a atual deve ser controlada para que informações novas ou movimentos propositais realizados pela câmera não sejam perdidos. Com este objetivo, um contador de reinício (“reset”) foi implementado. Se o contador atingir um número maior que um limiar pré-estipulado, a imagem estabilizada seguinte passa a ser utilizada como referência.

4.2 Movimentos desejados e indesejados

Para que um sistema de estabilização de imagem comporte-se de forma satisfatória, é necessária a separação entre os movimentos desejados (realizados pelo operador da câmera ou pelo sistema de captação de imagem de forma proposital) das pequenas vibrações e variações indesejadas. Neste trabalho, foi suposto que movimentos indesejados são caracterizados por componentes de alta freqüência nos vetores de movimento, enquanto baixas freqüências correspondem a movimentos desejados. Esta relação de movimentos desejados e indesejados, em baixa e alta freqüência respectivamente, é amplamente utilizada nos trabalhos encontrados na literatura.

Os filtros digitais passa baixa, como média de movimentos ("Moving Average"), filtros de Chebychev e Butterworth ou os filtros adaptativos como Kalmam, podem ser utilizados para separar as componentes de freqüência. Aplicando-se estes filtros aos vetores de movimento global estimados e aos parâmetros do modelo "Affine", é possível separar os movimentos desejados corrigindo apenas as rotações, deslocamentos verticais e deslocamentos horizontais involuntários.

Neste trabalho, foram testados os filtros passa baixa Média de Movimento, Butterworth e Chebychev. As características e resultados de desempenho obtidos para cada filtro foram analisadas com relação ao custo computacional, ordem e capacidade de suavização de movimento. Após esta análise, o filtro de melhor comportamento foi utilizado no sistema de estabilização de imagem.

As características, resultados e análise dos filtros estudados são apresentados a seguir.

4.2.1 Filtro Média de Movimento

O filtro digital média de movimentos pode ser definido pela equação (15).

$$y(i) = \frac{1}{M} \sum_{j=0}^{M-1} x(i-j) \tag{15}$$

Sendo, y o sinal de saída e x o sinal de entrada. M descreve a ordem do filtro, para um filtro de quinta ordem, M=5.

A resposta em freqüência deste tipo de filtro de diferentes ordens é apresentada na figura 4.5:

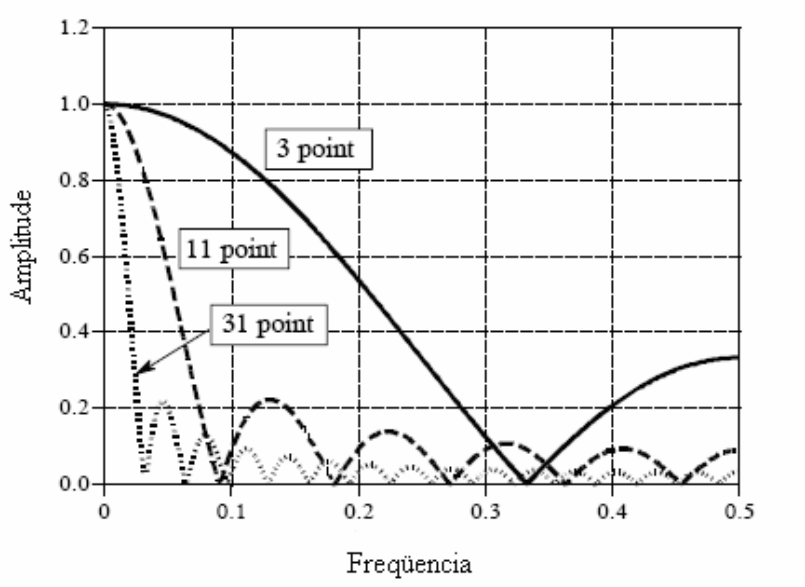


Figura 4.5 – Resposta em freqüência do filtro “Moving average” (Smith W., 1998).

Os filtros de média de movimento são comumente utilizados, devido a sua facilidade de utilização e entendimento. Requerem baixo custo computacional, sendo empregados com uma convolução no domínio do tempo.

Embora apresentem um pobre desempenho no domínio da freqüência (figura 4.5), pois possuem pouca habilidade em separar as faixas de freqüência, filtros de média de movimento são grandes "suavizadores", isto é, de bom desempenho no tempo, o que somado com seu rápido processamento, torna-os uma boa ferramenta na suavização dos vetores de movimento.

4.2.2 Filtros Passa Baixa Butterworth e Chebychev

Um Filtro Buterworth passa baixa também foi estudado para o processo de suavização de movimento e pode ser descrito pela equação (16):

$$a(1) * y(n) = b(1) * x(n) + b(2) * x(n-1) + \ldots + b(N+1)x(N)\ldots \\ - a(2) * y(n-1) - \ldots - a(N+1) * y(N) \quad (16)$$

em que;

N é a ordem do filtro, *y(n)* e *x(n)* são os sinais de saída e entrada respectivamente.

O filtro utilizado foi projetado através do software Matlab, sendo encontrado para N = 5 os vetores de coeficientes *a* e *b* apresentados a seguir:

a = [1 -2,9754 3,8060 -2,5453 0,8811 -0,1254]

b = [0,0013 0,0064 0,0128 0,0128 0,0064 0,0013]

A resposta em freqüência do filtro Butterworth de quinta ordem é apresentada a seguir na figura 4.6:

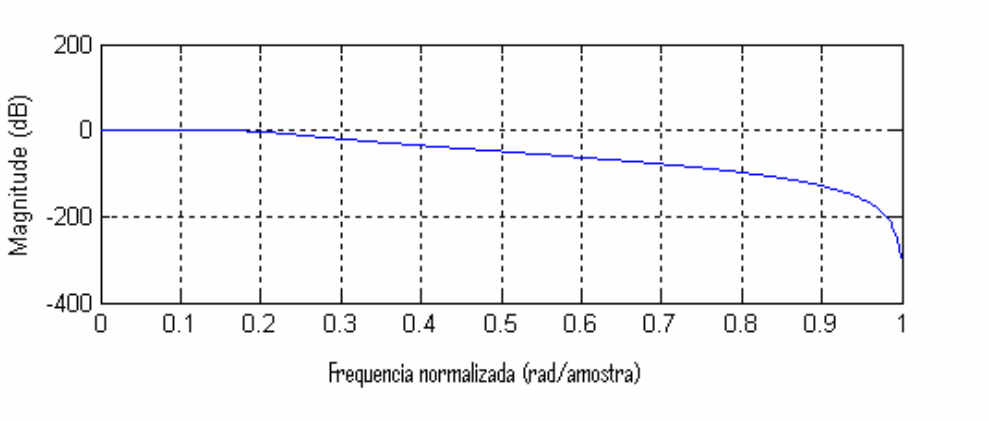


Figura 4.6 - Resposta em freqüência filtro Butterworth quinta ordem

Outro filtro estudado foi o filtro Chebychev passa baixa. Este filtro também pode ser descrito pela equação (16) e os vetores de coeficientes encontrados pelo Matlab são:

a = [1 -3,6925 5,7275 -4,7864 2,1123 -0,3927]

b = [0,0006 0,0029 0,0058 0,0058 0,0029 0,0006]

A resposta em freqüência é apresentada na figura 4.7, para um filtro de quinta ordem.

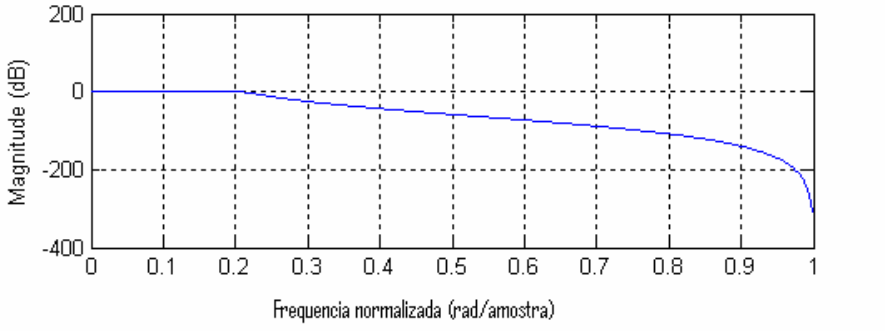


Figura 4.7 - Resposta em freqüência filtro Chebychev quinta ordem

Os filtros de Butterworth e Chebychev apresentam um melhor desempenho na resposta em freqüência em relação ao filtro Média de Movimento, porém apresentam uma maior complexidade na sua aplicação, tanto em hardware quanto em software.

4.2.3 Análise dos Filtros utilizados

Os filtros apresentados anteriormente foram aplicados ao vetor de movimentos horizontal, vertical e rotacional, estimados para um vídeo. Primeiramente filtros de quinta ordem foram empregados. Os resultados em relação às coordenadas horizontais dos vetores de movimento estimados são apresentados na figuras 4.8, 4.9 e 4.10:

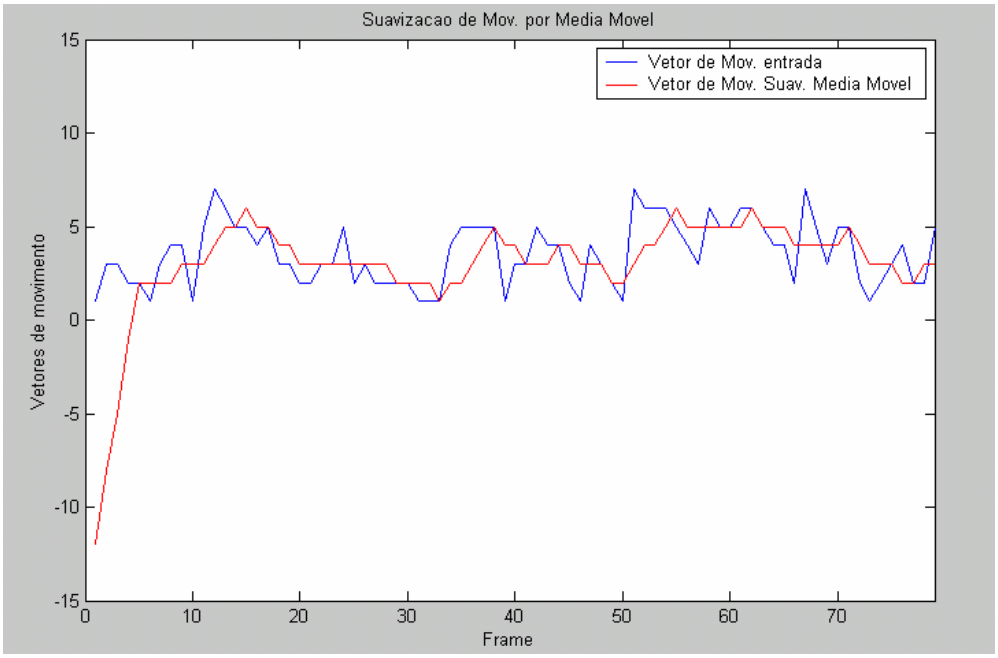


Figura 4.8 - Filtro Média de Movimento (quinta ordem)

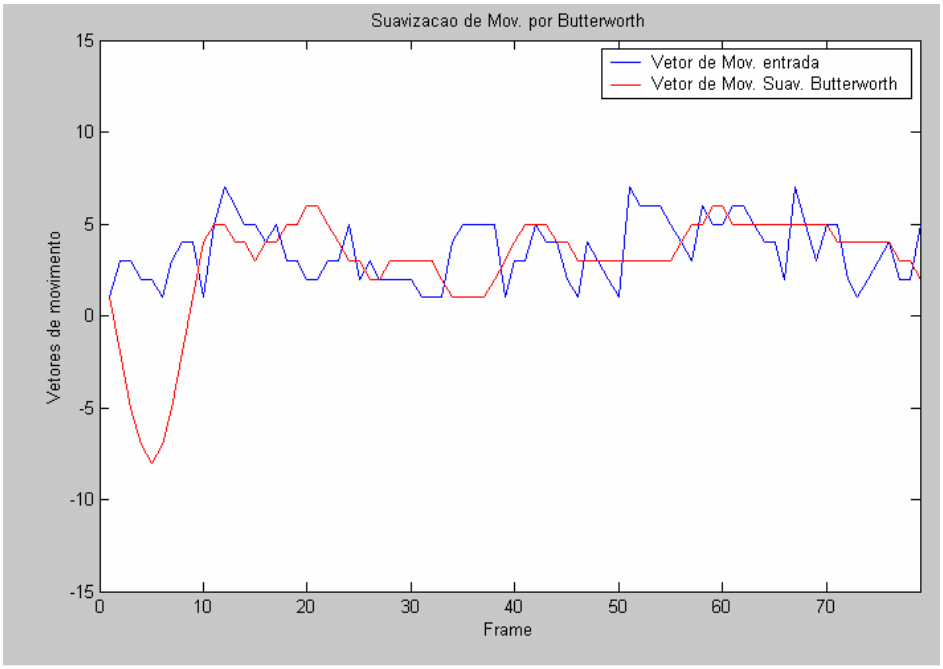


Figura 4.9 - Filtro Butterworth (quinta ordem)

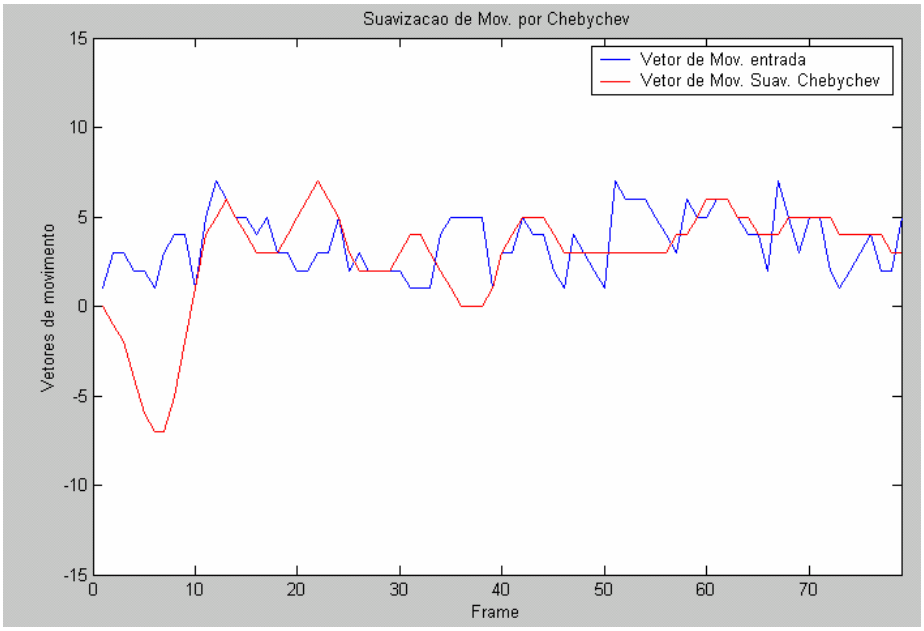


Figura 4.10 - Filtro Chebychev (quinta ordem)

É possível observar que, os filtros Chebychev e Butterworth, embora possuam uma melhor resposta em freqüência, não apresentam uma diferença significativa em relação ao filtro Média de Movimento na suavização do vetor de movimento.

Em seguida, filtros de décima ordem foram aplicados aos vetores de movimento estimados. Os resultados são apresentados nas figuras 4.11, 4.12 e 4.13:

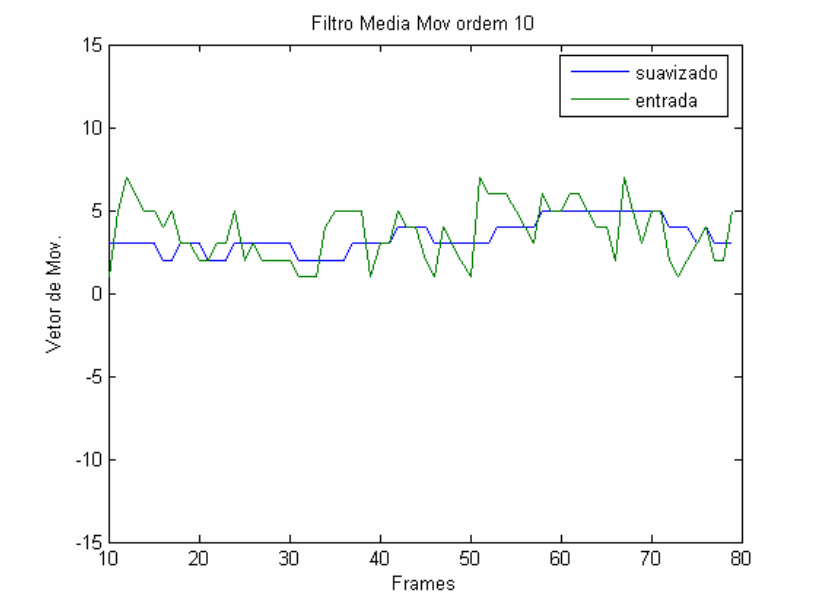


Figura 4.11 – Filtro Media Mov. Ordem 10

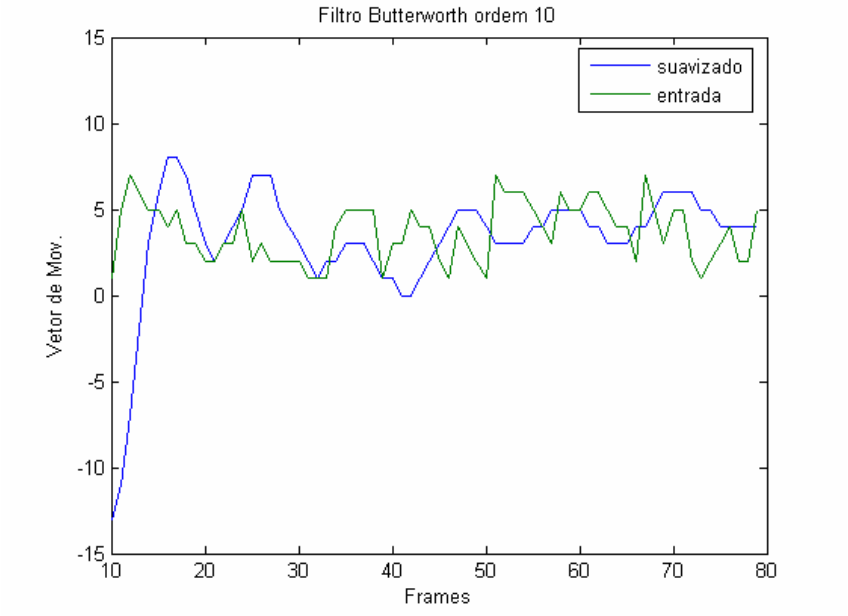


Figura 4.12 – Filtro Butterworth ordem 10

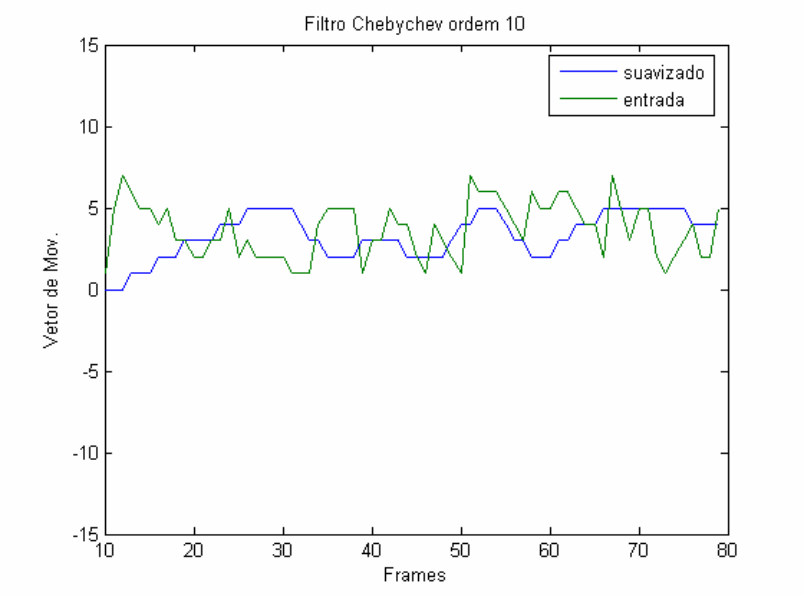


Figura 4.13 – Filtro Chebychev ordem 10

Através da análise dos gráficos dos vetores de movimento suavizados obtidos para cada tipo de filtro empregado, é possível observar que, o filtro Média de Movimento apresentou um desempenho semelhante ou superior aos outros filtros aplicados ao sistema. Isto pode ser quantizado pelo cálculo da variância dos vetores de movimento suavizados estimados. Os valores das variâncias são apresentados na tabela 4.1:

Ordem	**Média de Movimento**	**Butterworth**	**Chebychev**
5	***7,5***	***8,6***	***8,5***
10	***0,9***	***2,4***	***2,5***

Tabela 4.1-Variâncias dos vetores de movimento suavizados

Observa-se que, os valores das variâncias são menores para os vetores obtidos pelos filtros Média de Movimento de quinta e décima ordem. Devido a este fato, somado ao de os filtros Média de Movimento possuírem fácil implementação e

entendimento, optou-se por utilizar um filtro Média de Movimento no processo de suavização e separação de movimentos desejados e indesejados no sistema desenvolvido.

É possível notar também que, ao aumentarmos a ordem do filtro, obtemos uma maior suavização do movimento, porém uma maior suavização acarretará em um maior deslocamento da imagem e, conseqüentemente, uma maior perda nas bordas das mesmas após o processo de compensação de movimento. Este efeito foi observado nos testes realizados com filtros Média de Movimento de ordem maior que 10, como por exemplo, ordem 15, com os quais o sistema apresentou grandes deslocamentos e perdas nas bordas das imagens.

Portanto, para o funcionamento do sistema optou-se por um filtro Média de Movimento de ordem 10.

4.3 Compensação de movimentos

Após a estimação dos vetores de movimento, a classificação do movimento em rotacional ou translacional e a suavização e filtragem do vetor de movimento global ou dos parâmetros "Affine", podemos corrigir as imagens numa seqüência de vídeo, deformando-as ou deslocando-as de forma a diminuir ou anular as vibrações indesejadas.

No sistema desenvolvido, a correção foi realizada deslocando a imagem de forma contrária ao vetor de movimento indesejado, no caso de movimentos

translacionais. Os movimentos rotacionais estipulados são compensados, obtendo-se outro modelo contrário aos parâmetros de rotação e translação dos movimentos indesejados. A figura 4.14 ilustra o modo de correção de movimento realizado.

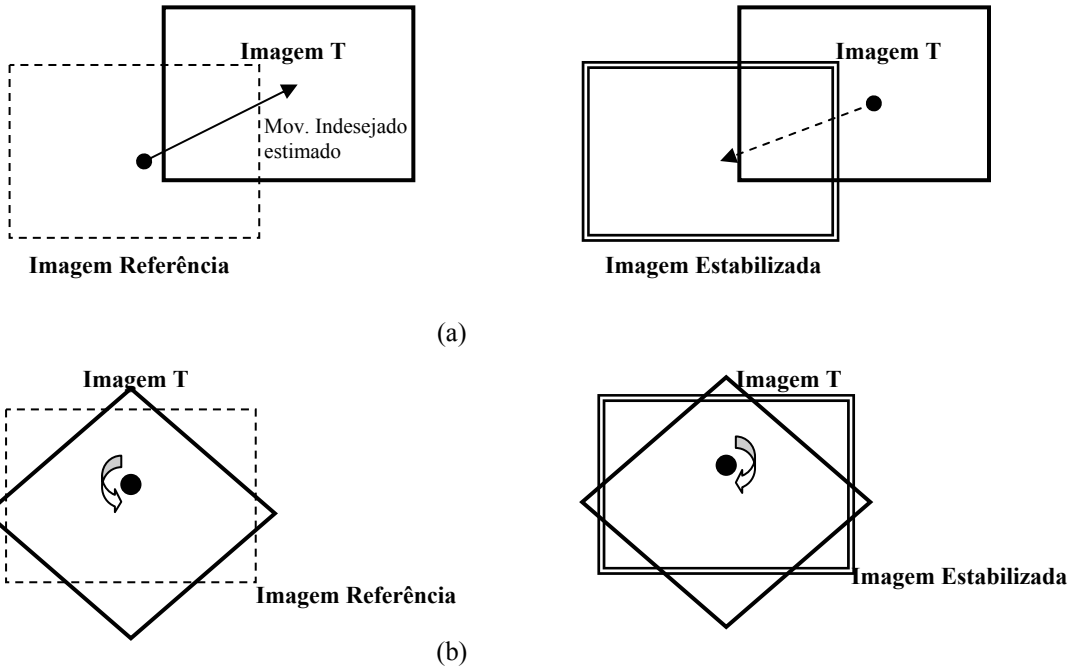


Figura 4.14 – (a) Compensação de movimento translacional, (b) Compensação movimento rotacional

4.3.1 Variações profundas nas imagens

Nas seqüências de imagens, nas quais o sistema de estabilização deve atuar, existem freqüentemente objetos se movimentando, pessoas passando em frente à câmera, imagens captadas com deformações e borramentos ou grandes ângulos de rotação entre as imagens. Estes efeitos causam mudanças profundas nas imagens que, interferem na estimação dos vetores de movimento local. Estas interferências nas estimações de movimento causam erros no processo de estabilização.

Embora o critério de combinação escolhido, EMQ, tenha apresentado um bom desempenho em relação ao critério MDA, problemas com grandes variações na luminosidade, na rotação e nos níveis de cinza são intrínsecos a técnica CB de estimação dos vetores de movimento.

Com o objetivo de diminuir este tipo de erro, utilizou-se um sistema de detecção de mudanças profundas nas imagens. Através de um algoritmo de detecção de movimento média móvel ponderada, apresentado em Gonzáles e Woods (1993), e de um limiar pré-estabelecido, foi possível detectar grandes alterações na seqüência das imagens, evitando que possíveis erros ocorressem e se propagassem no sistema de estabilização.

Na ocorrência de uma grande variação dos níveis de cinza entre as imagens, o processo de correção é desabilitado, não interferindo na imagem original de entrada, evitando desta forma, uma propagação de erro da estimação de movimento. O processo de correção é habilitado novamente quando uma nova imagem é capturada.

O algoritmo implementado na detecção de grandes variações das imagens calcula a diferença ponderada no tempo, em termos de níveis de cinza, entre a imagem diferença anterior ($\mathbf{I}_{med}^{n-1}$) e a atual ($\mathbf{I}_{i}^{n}$), mostrada na equação (16):

$$I_{med}^{n} = \alpha\, I_{i}^{n} + (1-\alpha).\, I_{med}^{n-1} \qquad (16)$$

Avaliando os pixeis da imagem diferença $\mathbf{I}_{med}$, quanto ao valor em nível de cinza e quanto à sua quantidade na imagem, podemos estipular uma classificação relacionada às variações nas imagens.

Dependendo da constante α escolhida para a ponderação, o sistema detectará somente variações bruscas entre as imagens, ao mesmo tempo em que ruídos não causarão grandes perturbações, pois informações das imagens antigas serão preservadas.

Para os testes e funcionamento do sistema desenvolvido, a constante α foi fixada no valor 0,7.

4.4 Parâmetros do sistema de estabilização

Em cada processo do sistema de estabilização desenvolvido são necessários parâmetros estimados através da observação dos dados capturados.

Para a classificação do movimento em rotacional ou translacional, um limiar da variância dos vetores de movimento local é estipulado. Nos testes realizados, foram utilizados vários valores de limiar, e o valor de 65 apresentou um melhor desempenho, porém, variações no tamanho da imagem e iluminação podem interferir na grandeza dos valores calculados. Desta forma, ajustes para diferentes equipamentos e locais de captura de imagem devem ser efetuados.

Na escolha da imagem de referência, dois parâmetros são necessários: limiar da distância euclidiana entre as imagens e a distância máxima no tempo entre a imagem de referência e a de entrada.

Novamente, vários testes foram realizados com o objetivo de se encontrar os valores de melhor desempenho para os parâmetros. Os efeitos causados por imagens capturadas em diferentes locais e situações de iluminação foram fatores importantes na escolha destes valores.

O limiar da distância euclidiana foi fixado em 5 pixeis e interfere no máximo deslocamento permitido da imagem. Com relação à distância no tempo entre as imagens, um cuidado maior deve ser tomado, pois informações importantes da seqüência de imagens podem ser perdidas. Nos testes, fixou-se uma distância máxima de 7 imagens entre a de referência e entrada.

Um último parâmetro é estipulado no algoritmo de detecção de movimento de objetos para determinar mudanças profundas nas imagens, evitando-se uma estimação de movimento errada. Na simulação, este parâmetro foi determinado em 1000 pixeis, presentes na imagem diferença acima do nível de cinza 250, para uma imagem de 320 x 240 pixeis e 256 níveis de cinza. Para esta escolha, também se aplicou vários valores a este parâmetro, observando o valor de melhor desempenho, isto é, capaz de evitar que grandes movimentos prejudiquem o sistema e que movimentos possíveis de estabilização sejam corrigidos.

4.5 Conclusão

Com a definição da estratégia de busca n passos e do processo CB para a estimação de movimento, foi possível realizar a estabilização de imagens para movimentos rotacionais e translacionais, sem grandes custos computacionais, viabilizando a operação do sistema em tempo real. Observando os resultados obtidos na simulação dos filtros passa baixa projetados, verificou-se um melhor comportamento do filtro Média de movimento, que juntamente com a sua facilidade de implementação, justifica a sua utilização no sistema desenvolvido. Foi necessário também, estabelecer uma relação entre a ordem do filtro e as perdas nas bordas da imagem após a estabilização, pois, filtros de ordem grande acarretam em grandes perdas. Neste sistema, foram utilizados quatro parâmetros que devem ser ajustados para um melhor funcionamento do processo de estabilização para diferentes tipos de iluminação, ambiente, etc.

Capítulo 5

Testes e Resultados

5.1 Teste do Sistema de Estabilização de Imagem

A estrutura do sistema desenvolvido é apresentada na figura 5.1.

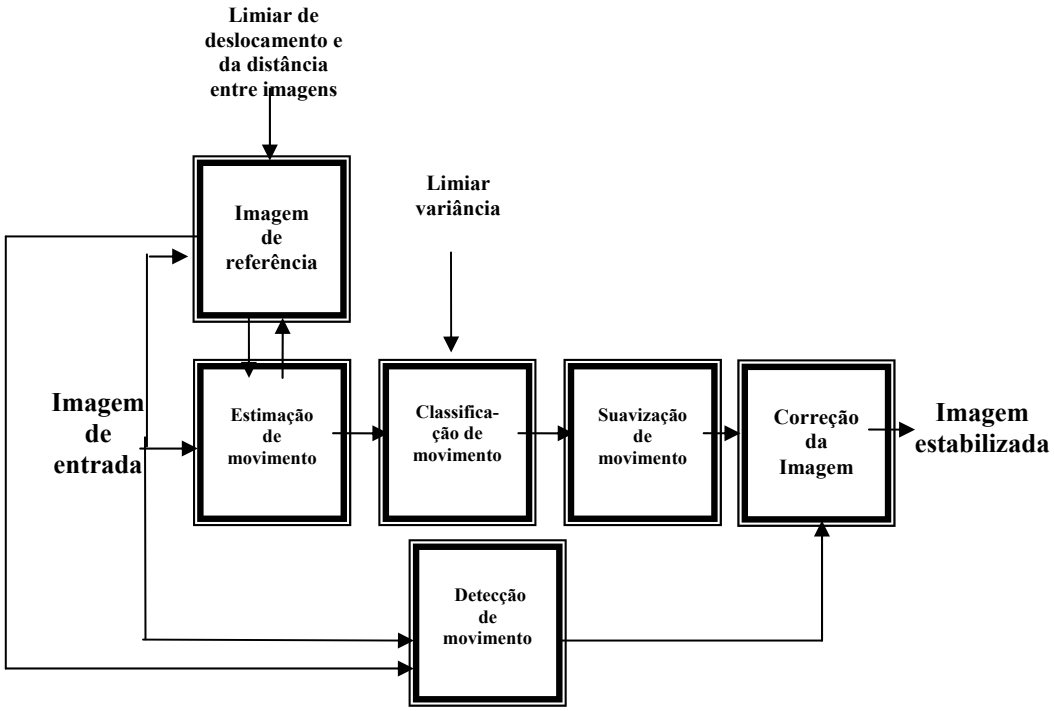


Figura 5.1 – Estrutura do sistema proposto

A implementação do sistema de estabilização apresentado, incorreu no desenvolvimento de um aplicativo na linguagem C++ (figura 5.2) na ferramenta de desenvolvimento Microsoft Visual C++ 6.0, utilizando a biblioteca OpenCV ("Open Source Computer Vision Library"), dedicada ao processamento de imagens. Uma imagem da tela do aplicativo desenvolvido está mostrada na figura 5.2:

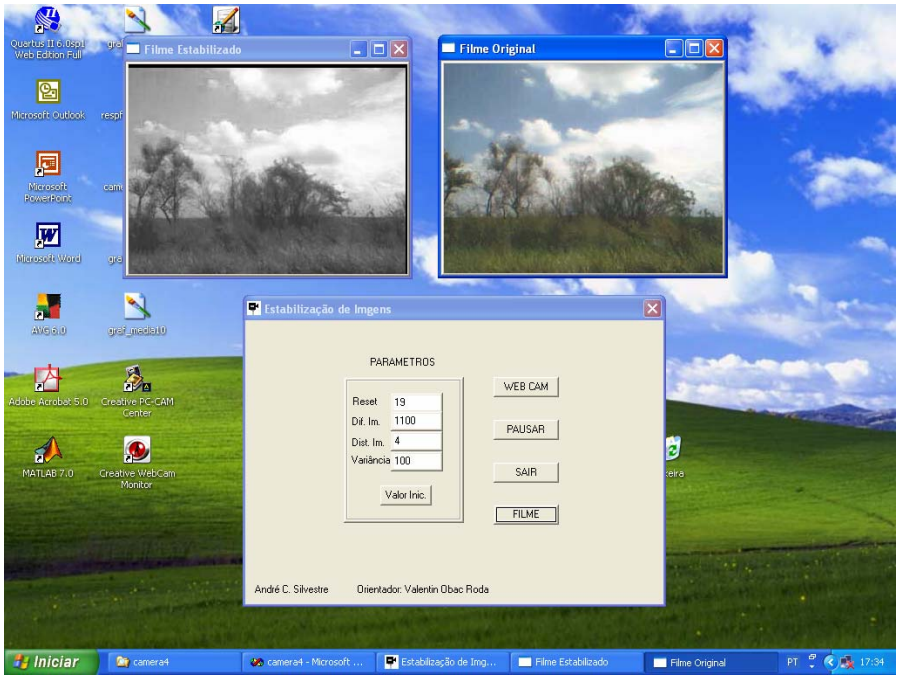


Figura 5.2 – Sistema de Estabilização de imagens

OpenCV é uma coleção de funções em C ou C++, desenvolvida pela empresa Intel, que implementa alguns algoritmos popularmente conhecidos de processamento de imagem ou visão computacional.

O aplicativo desenvolvido foi executado em um microcomputador Pentium IV. Durante a simulação em tempo real, as imagens foram capturadas por uma câmera, tipo "WebCam", conectada a porta USB do Microcomputador.

As imagens de 240 x 320 pixeis RGB, capturadas pela câmera WebCam Creative NX Pro a 12 fps (frames por segundo), foram convertidas do sistema de cor RGB para YCrCb. O canal de Luminância Y foi utilizado como entrada para o sistema de estabilização.

Foram capturadas imagens em ambientes fechados, como uma sala de laboratório ou escritório, exemplificado na figura 5.3, e imagens em ambientes abertos, como por exemplo, imagens de paisagens ou de transito urbano, capturadas em um carro em movimento, como mostrado na figura 5.4:

Figura 5.3 – Ambiente fechado

Figura 5.4 – Ambiente aberto

5.2 Resultados

Primeiramente, foram realizados os testes para as imagens de ambiente fechado.

Nas figuras 5.5 e 5.6, são apresentados os vetores de movimento global estimados nas direções horizontal e vertical, respectivamente, e também os vetores de

movimento suavizados (movimentos desejados) e indesejados (movimentos de alta freqüência, vibrações), calculados para as imagens capturadas.

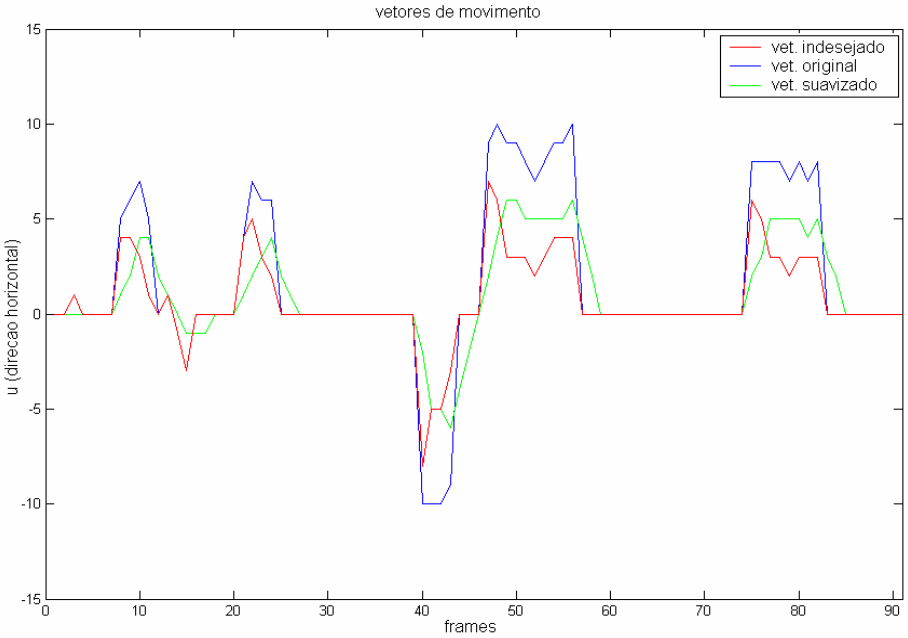


Figura 5.5 – Vetores de movimento horizontal

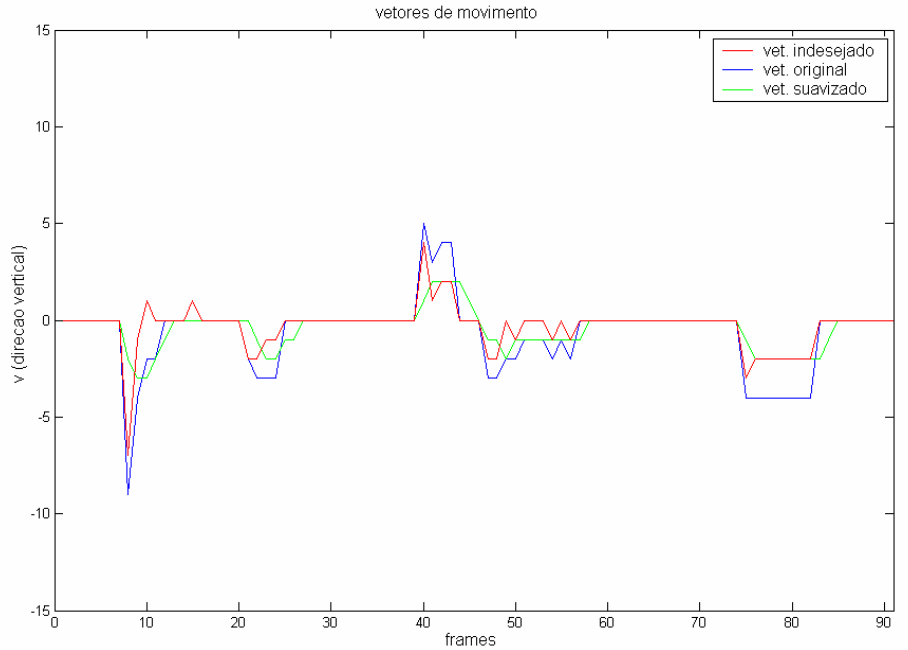


Figura 5.6 – Vetores de movimento vertical

Os ângulos de rotação global estimados, ângulos suavizados (desejados) e indesejados, para uma seqüência de imagem artificialmente rotacionada, são mostrados na figura 5.7.

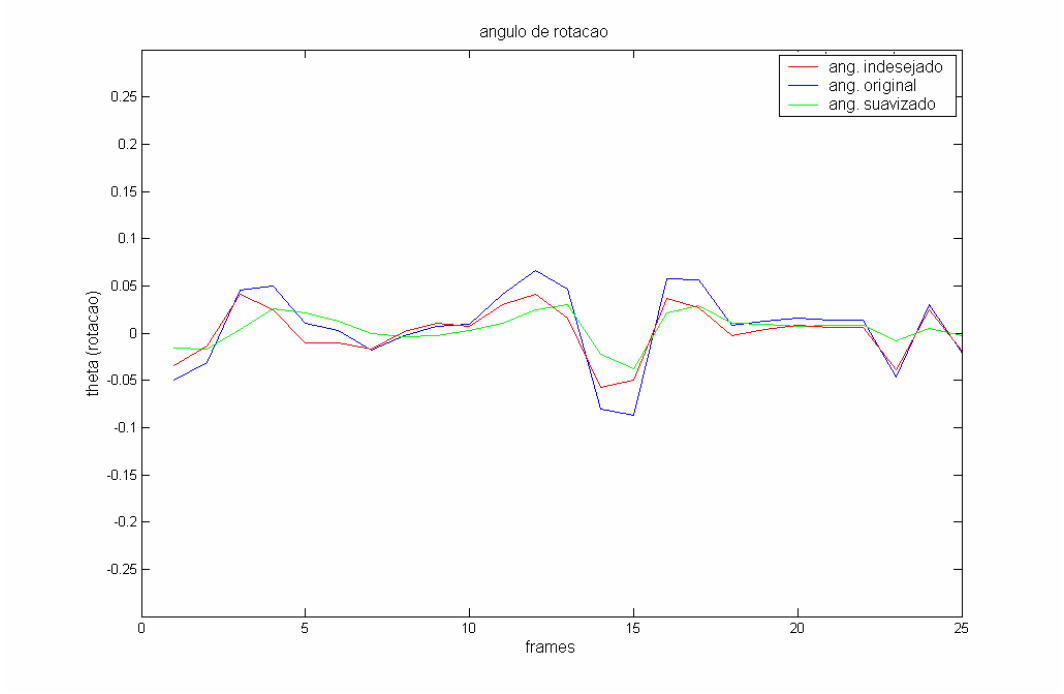


Figura 5.7 – Ângulo de rotação

Em seguida, foram utilizados como imagens de entrada, os vídeos capturados em ambiente aberto por um carro em movimento, o que pode ser considerado como uma situação real de uso do sistema de estabilização de imagens.

Nas figuras 5.8 e 5.9 são apresentados os vetores de movimento global estimados nas direções horizontal e vertical, respectivamente, e também os vetores de movimento suavizados (movimentos desejados) e indesejados, calculados para as imagens capturadas.

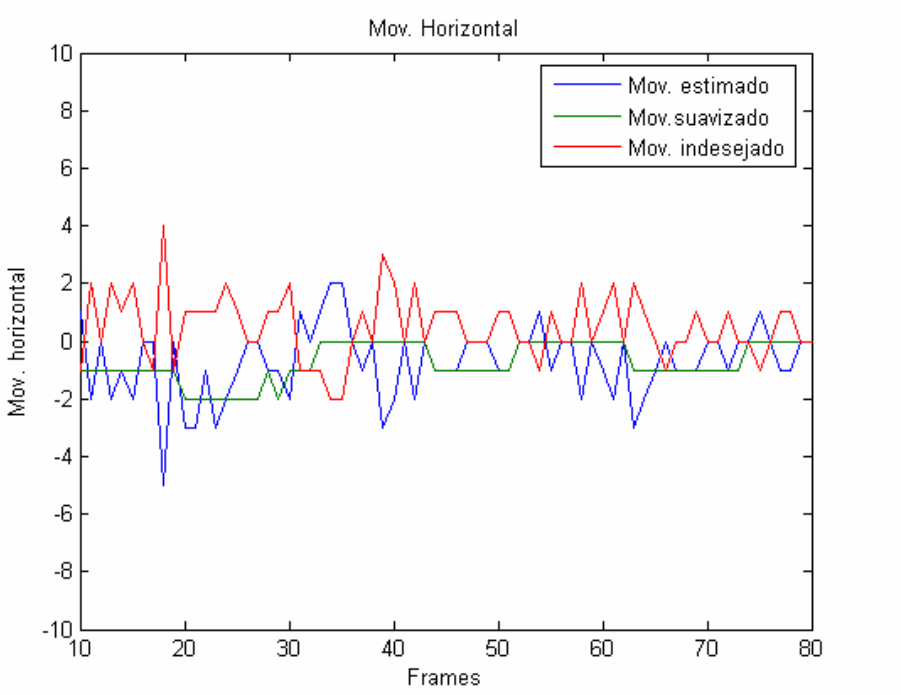


Figura 5.8– Vetores de movimento horizontal

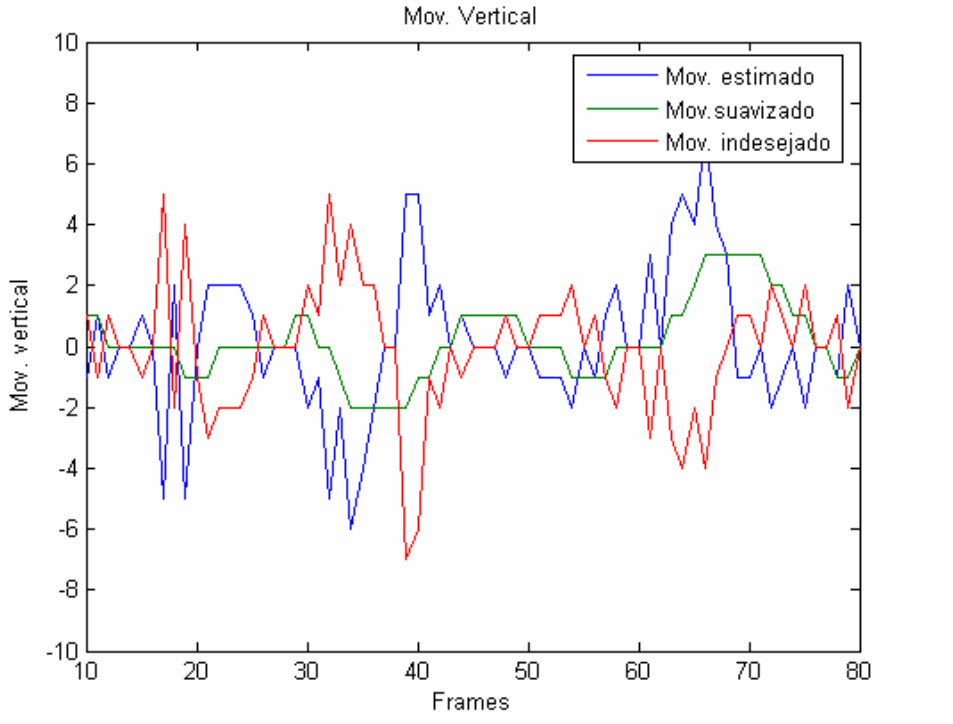


Figura 5.9 – Vetores de movimento vertical

Os dados apresentados nas figuras acima nos mostram a ocorrência de uma suavização para a perturbação presente na seqüência de imagens originais, já que,

movimentos suavizados apresentam pouca variação de direção e os vetores de movimentos indesejados (que futuramente serão utilizados na correção de movimento) são compostos por grande parte das mudanças de direção de alta freqüência presentes na estimação de movimento da imagem, permitindo desta forma, que movimentos suaves sejam mantidos.

Após a estabilização da seqüência de imagens, foram estimados os vetores de movimento para a seqüência estabilizada. Nas imagens em ambiente fechado, os resultados obtidos para a direção horizontal e para o ângulo de rotação são comparados com os resultados de estimação de movimento da seqüência de imagens original e são apresentados nas figuras 5.10, 5.11, respectivamente.

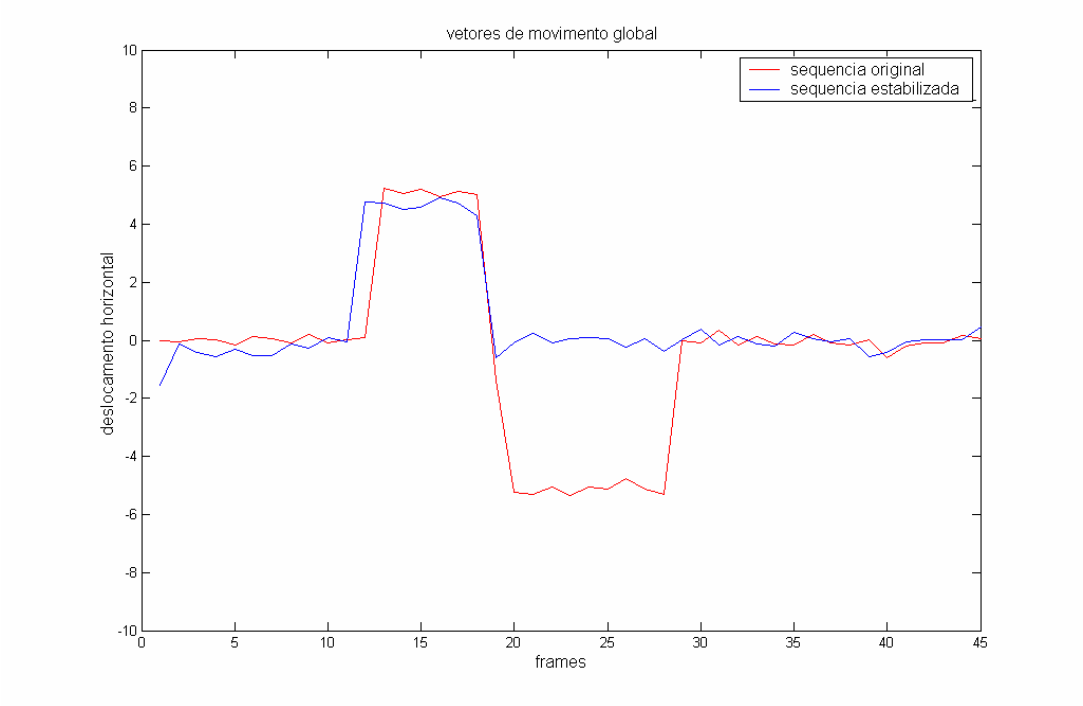


Figura 5.10 - Vetores de movimento horizontal (seqüência original e estabilizada)

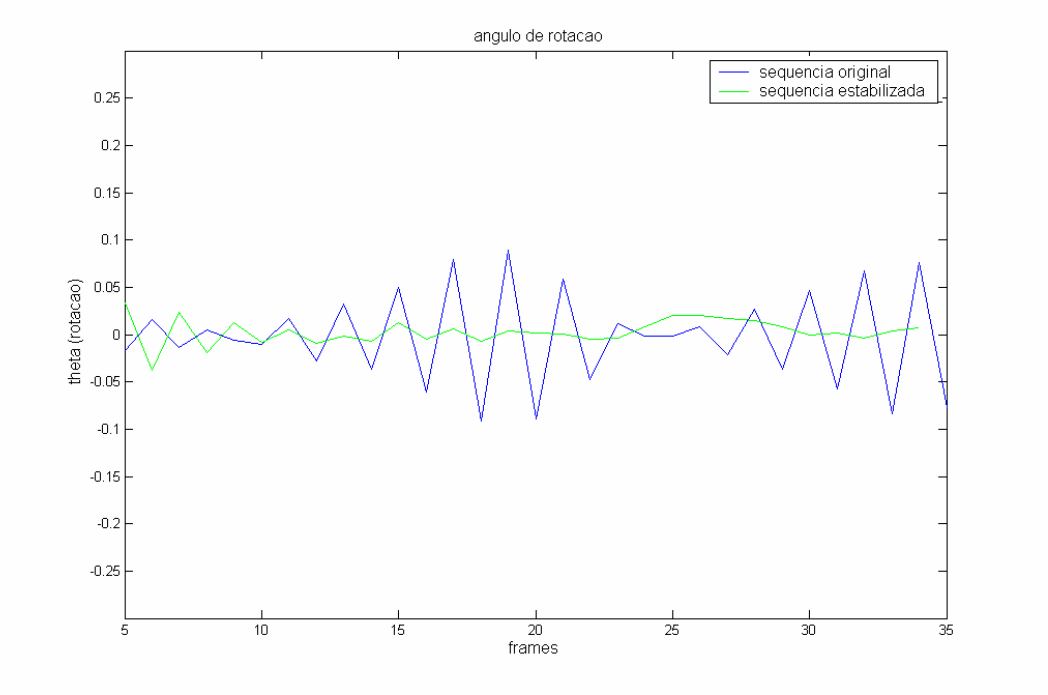


Figura 5.11 - Ângulo de rotação (seqüência original e estabilizada)

Foram obtidos também, os vetores de movimento para as imagens em ambiente aberto. A figura 5.12 apresenta os vetores calculados para a direção horizontal de movimento, para as imagens originais e estabilizadas.

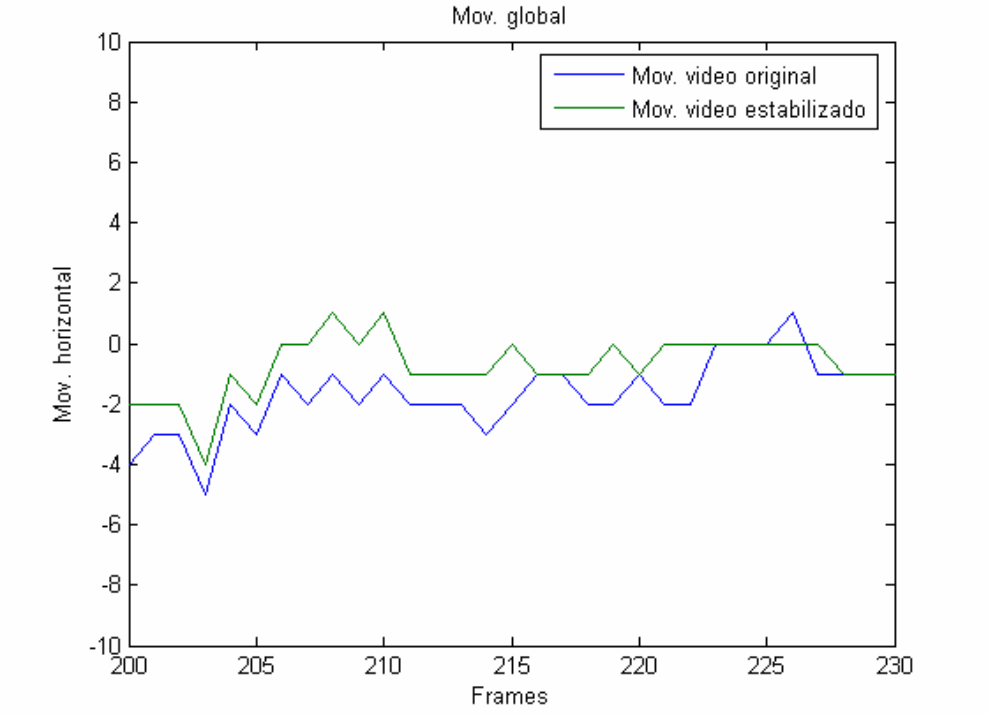


Figura 5.12 – Vetores de movimento (ambiente fechado)

Em todos os testes realizados, os parâmetros do sistema citados anteriormente foram fixado nos valores apresentados no capítulo 4 e um filtro Média de Movimento de ordem 10 foi utilizado.

Com o objetivo de realizar uma melhor análise dos resultados encontrados, foram calculadas as variâncias para os vetores de movimento original e estabilizado, das direções verticais e horizontais, para as seqüências de imagens capturadas. Os resultados são apresentados na tabela 5.1.

Direção	Movimento original	Movimento estabilizado
Horizontal – Ambiente Fechado	9,7469	2,3810
Vertical – Ambiente Fechado	1,6681	0,9181
Horizontal – Ambiente Aberto	1,3897	0,813
Vertical – Ambiente Aberto	13,969	13,3499

Tabela 5.1 – Variância dos vetores de movimento global

As variâncias calculadas para os movimentos originais possuem um valor maior do que em relação aos valores obtidos para os movimentos estabilizados tanto para imagens de ambiente aberto quanto fechado, indicando desta forma, que o sistema de estabilização de imagem foi capaz de suavizar os movimentos para as duas situações de ambiente retirando os movimentos indesejados de alta freqüência, presentes nas imagens originais capturadas. Entretanto, é importante notar um melhor desempenho do sistema para ambiente fechado, mostrando que os parâmetros utilizados são mais adequados a este cenário.

5.3 Conclusão

Através dos resultados encontrados é possível observar que mesmo após o processo de estimação de movimento, ainda existem componentes de alta freqüência na seqüência de imagens estabilizadas, porém, obteve-se uma grande redução das vibrações nas imagens originais, estabilizando-as, principalmente para ambientes fechados. Assim, podemos verificar um bom comportamento do sistema em tempo real, estimando, classificando, suavizando e compensando os movimentos.

Capítulo 6

Análise em Hardware do Sistema

Esta secção propõe uma análise da implementação em hardware, do sistema de estabilização de imagens proposto.

A análise desenvolvida foi baseada na estrutura do sistema apresentada no capitulo 5, portanto, pode ser dividida em sete blocos básicos de processamento:

- Estimação de movimento;
- Classificação de movimento;
- Vetor de movimento global;
- Imagem de referência;
- Suavização de movimento;
- Detecção de grandes movimentos;
- Correção de movimento.

Para cada bloco de processamento foi desenvolvido um programa no software ALTERA Quartus 6.0 utilizando funções pré-definidas em hardware pelo software, que podem ser implementadas facilmente em uma FPGA.

O objetivo desta análise é apresentar a estrutura básica de uma possível solução da implementação em hardware do sistema, observando quais componentes, funções e

operações devem ser utilizadas e visando uma futura implementação real em hardware. Processos de endereçamento de memória e sinais de sincronismo do sistema não foram estudados.

6.1 Estimação de movimento

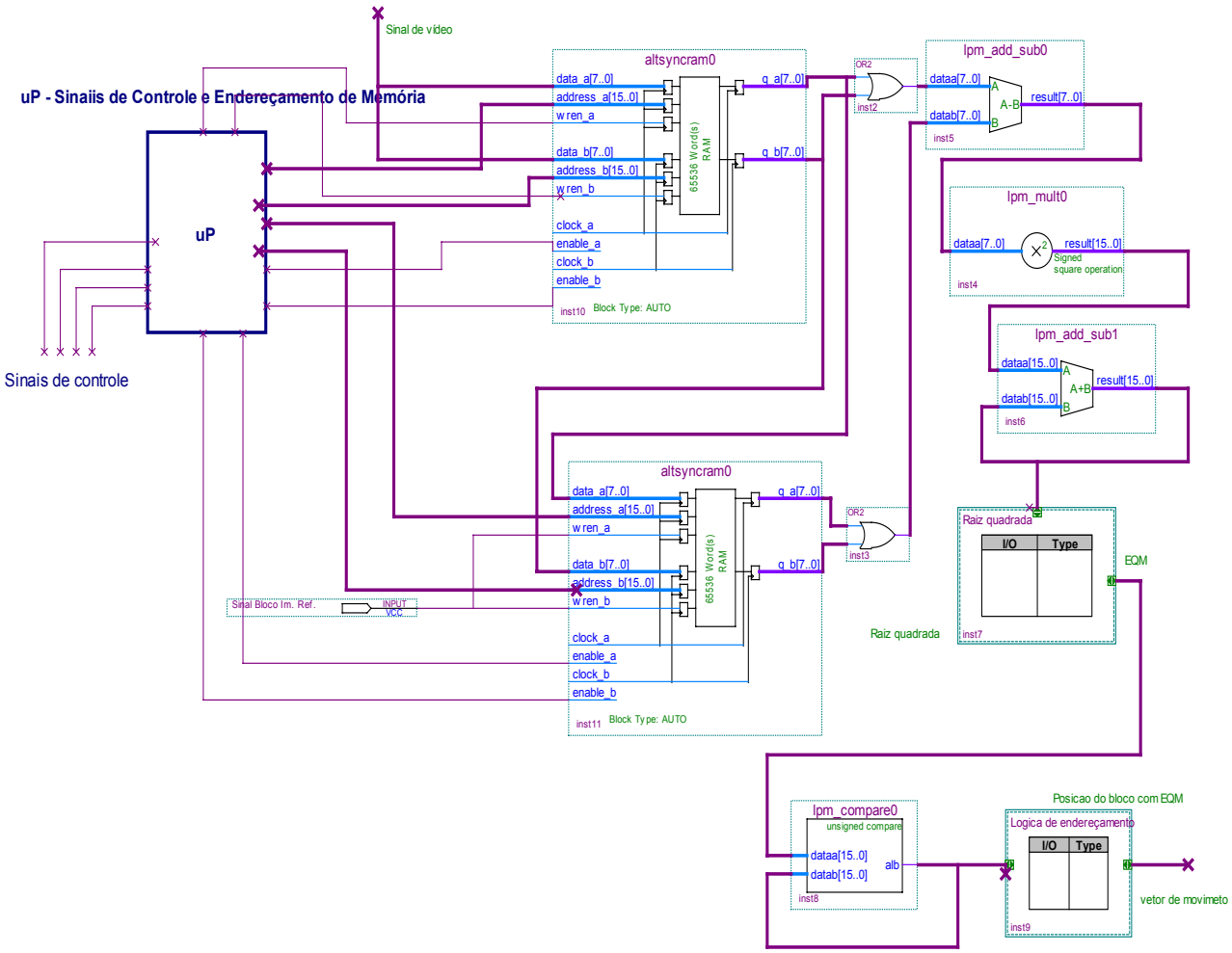


Figura 6.1 – Estimação de movimento

Para o processo de estabilização de movimento apresentado na figura 6.1, foram utilizadas duas memórias para o armazenamento dos quadros do vídeo atual e de referência, operadores de soma e multiplicação, comparadores e duas portas lógicas OU.

O bloco chamado de uP representa os sinais de sincronismo, de habilitação dos demais processos e de endereçamento das memórias.

Para o calculo da raiz quadrada utilizada no EMQ, pode-se utilizar a série de Taylor desta função. O diagrama da figura 6.2 representa o calculo da raiz quadrada por este método.

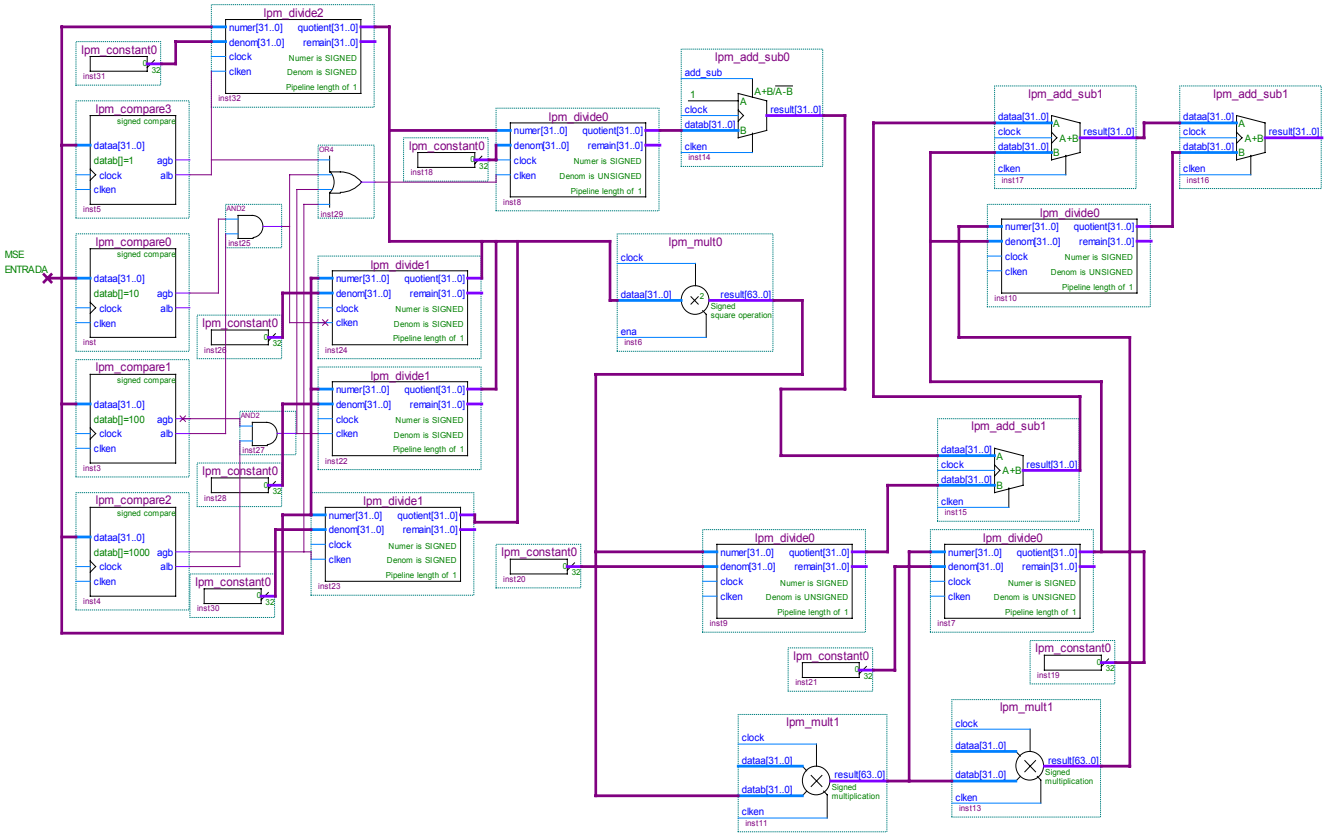


Figura 6.2 – Raiz quadrada

Na estimação de movimento rotacional, o vetor de movimento estimado pelo mesmo processo e estrutura anterior é ponderado por uma curva gaussiana através de uma matriz e é utilizado como entrada para o cálculo dos parâmetros do modelo Affine.

Os parâmetros do modelo Affine são calculados através do sistema de equações apresentados anteriormente no capitulo 4, desta forma, este processo apresenta cálculos mais complexos que, por sua vez, podem ser facilmente realizados por um software implementado num microprocessador.

6.2 Classificação de movimento

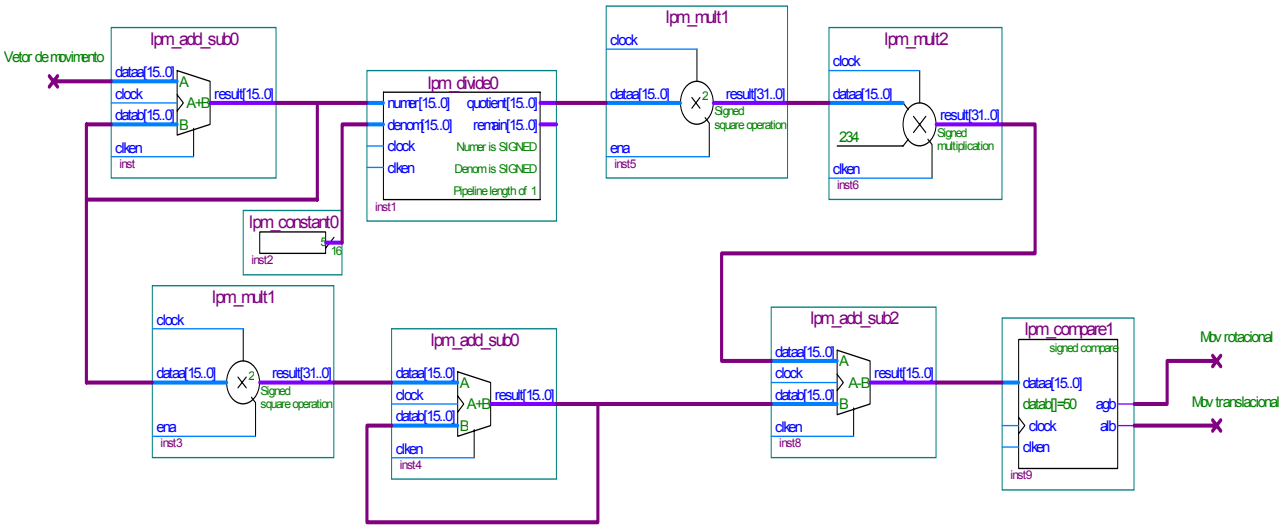


Figura 6.3 – Classificação de Movimento

O processo de classificação de movimento, representado na figura 6.3, foi desenvolvido utilizando um comparador e operadores de multiplicação, soma e divisão. Este processo recebe como entrada, os vetores de movimento estimados anteriormente, calcula a variância entre os vetores e os classifica em movimentos rotacionais e translacionais. Desta forma, tem como saída um sinal (bit) de habilitação da ponderação matricial na estimação de movimento rotacional ou de habilitação para o cálculo do vetor de movimento global translacional.

O diagrama apresentado acima representa o processo para uma coordenada do vetor de movimento estimado pelo bloco estimação de movimento, portanto, é necessário aplicar o mesmo processo para a outra coordenada do vetor.

6.3 Vetor de movimento global

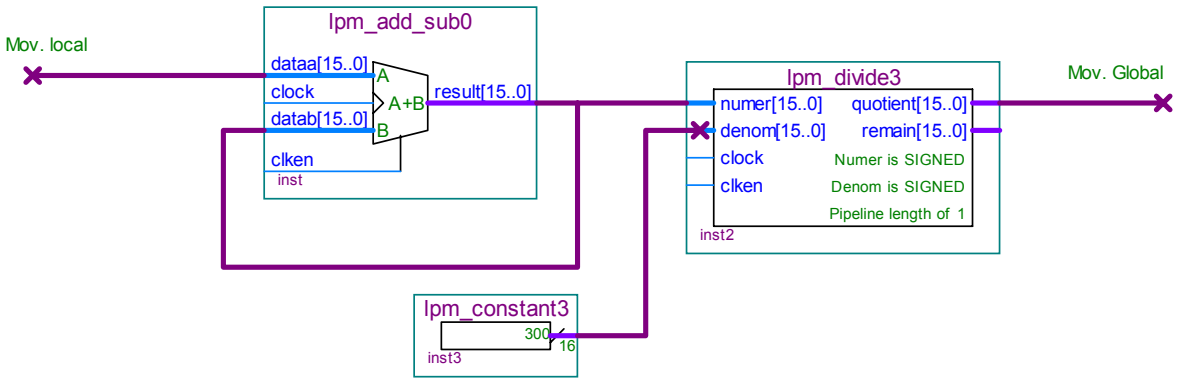


Figura 6.4 – Movimento Global

Para determinar o vetor de movimento global, foram empregados operadores de soma e divisão, como mostra a figura 6.4. O sinal de saída do bloco de classificação de movimento deve habilitar ou não habilitar esta operação, isto é, no caso da classificação de movimento translacional, temos como entrada, para este bloco, os vetores de movimento estimados anteriormente, por outro lado, se um movimento rotacional é classificado, esta operação não é realizada.

Novamente, o bloco de processamento desenvolvido representa apenas uma coordenada dos vetores de movimento, sendo necessário, aplicar o mesmo processo à outra coordenada.

6.4 Imagem de referência

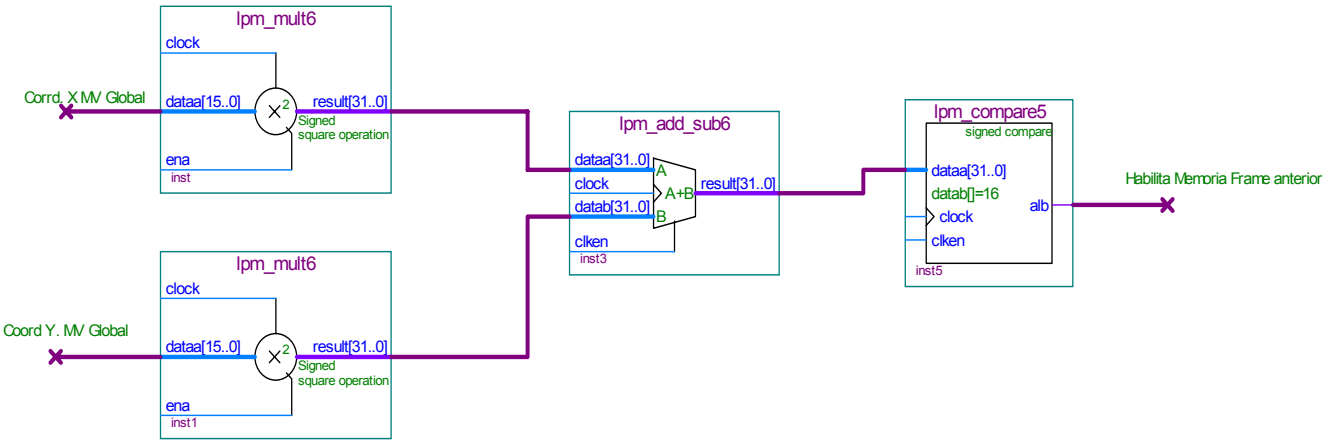


Figura 6.5 – Imagem de referência

Nesta etapa, foram utilizados um comparador e operadores de multiplicação, soma apresentados na figura 6.5. Este processo resulta num sinal de habilitação para a memória que irá armazenar o quadro de referência, e tem como sinal de entrada, os vetores de movimento globais estimados, para os movimentos translacionais ou os vetores que representam estes movimentos translacionais no modelo "Affine", sendo o último utilizado quando movimentos rotacionais são classificados pelo sistema.

6.5 Suavização de movimento

O diagrama de suavização de movimento foi desenvolvido para um filtro Média de Movimento de ordem 10. Foram utilizados 9 atrasadores ("buffers") e operadores de soma e divisão. A figura 6.6 representa o cálculo para uma coordenada do vetor de movimento global estimado ou para um parâmetro do modelo Affine, sendo novamente necessário aplicar o mesmo processo à outra coordenada ou parâmetro. Novamente, os sinais de entrada são os vetores de movimento globais calculados anteriormente ou os parâmetros do modelo "Affine".

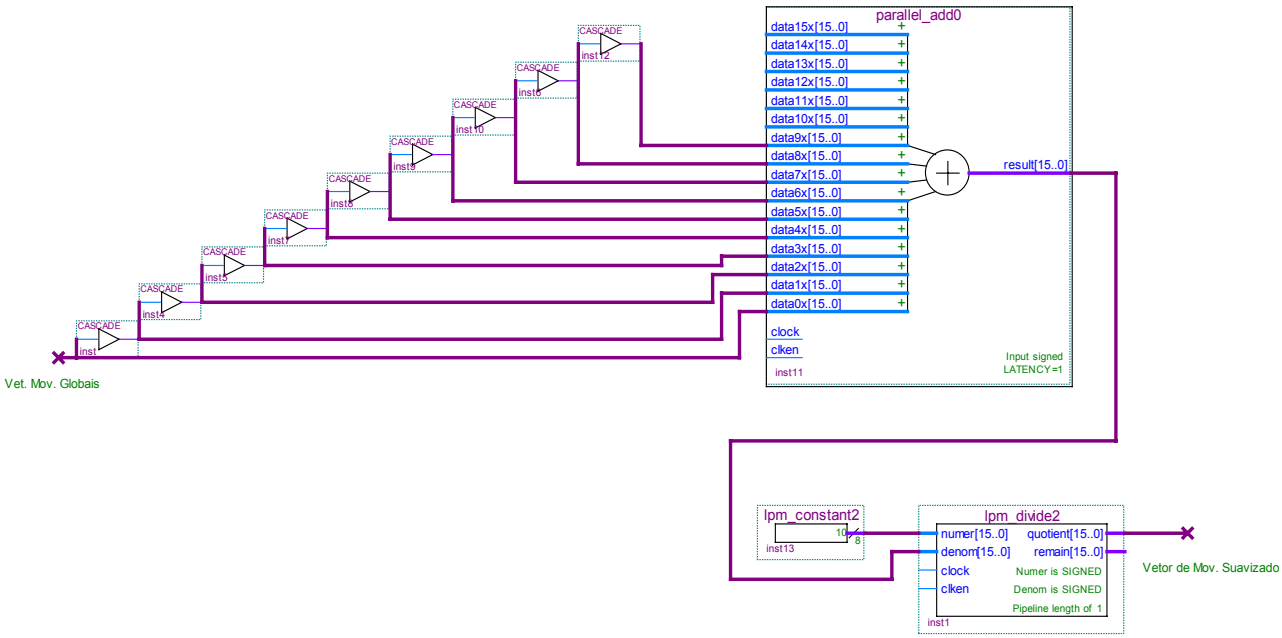


Figura 6.6 – Suavização de movimento

6.6 Detecção de grandes movimentos

Na detecção de grandes variações na imagem foram utilizadas as memórias com as informações dos quadros atuais e de referência do vídeo, operadores de soma, divisão, multiplicação e dois comparadores, como foi apresentado na figura 6.7. Este processo necessita de uma lógica complexa de endereçamento das memórias, porém, esta lógica não foi desenvolvida neste trabalho.

O diagrama de blocos apresentado resulta num sinal de habilitação para todo o sistema de estabilização de movimento.

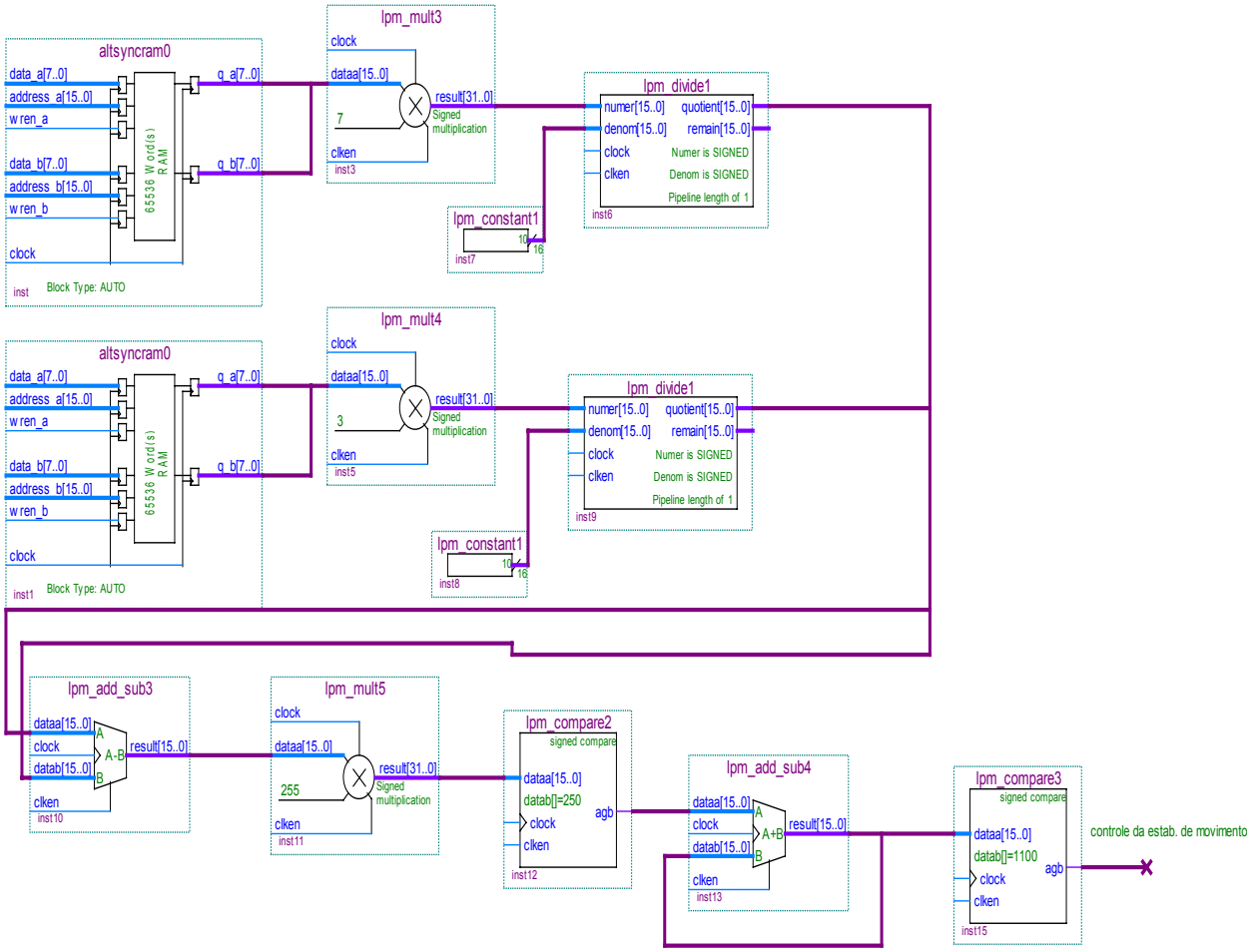


Figura 6.7 – Detecção de grandes movimentos

6.7 Correção de movimento

Este processo é realizado através do endereçamento da memória que armazena o quadro atual da seqüência de vídeo, baseado no vetor de movimento indesejado estimado, calculado pelo processo apresentado na figura 6.8. O diagrama apresentado recebe os vetores de movimento globais e os vetores de movimento suavizados, calcula os vetores de movimento indesejados e estabelece como a imagem deve ser deslocada através do endereçamento das memórias. Foram utilizados operadores de subtração, de multiplicação e comparadores para realizar este cálculo. No diagrama é apresentado o

processo para a coordenada horizontal, sendo necessário substituir o número de colunas pelo número de linhas, para realizar o mesmo processo para a coordenada vertical.

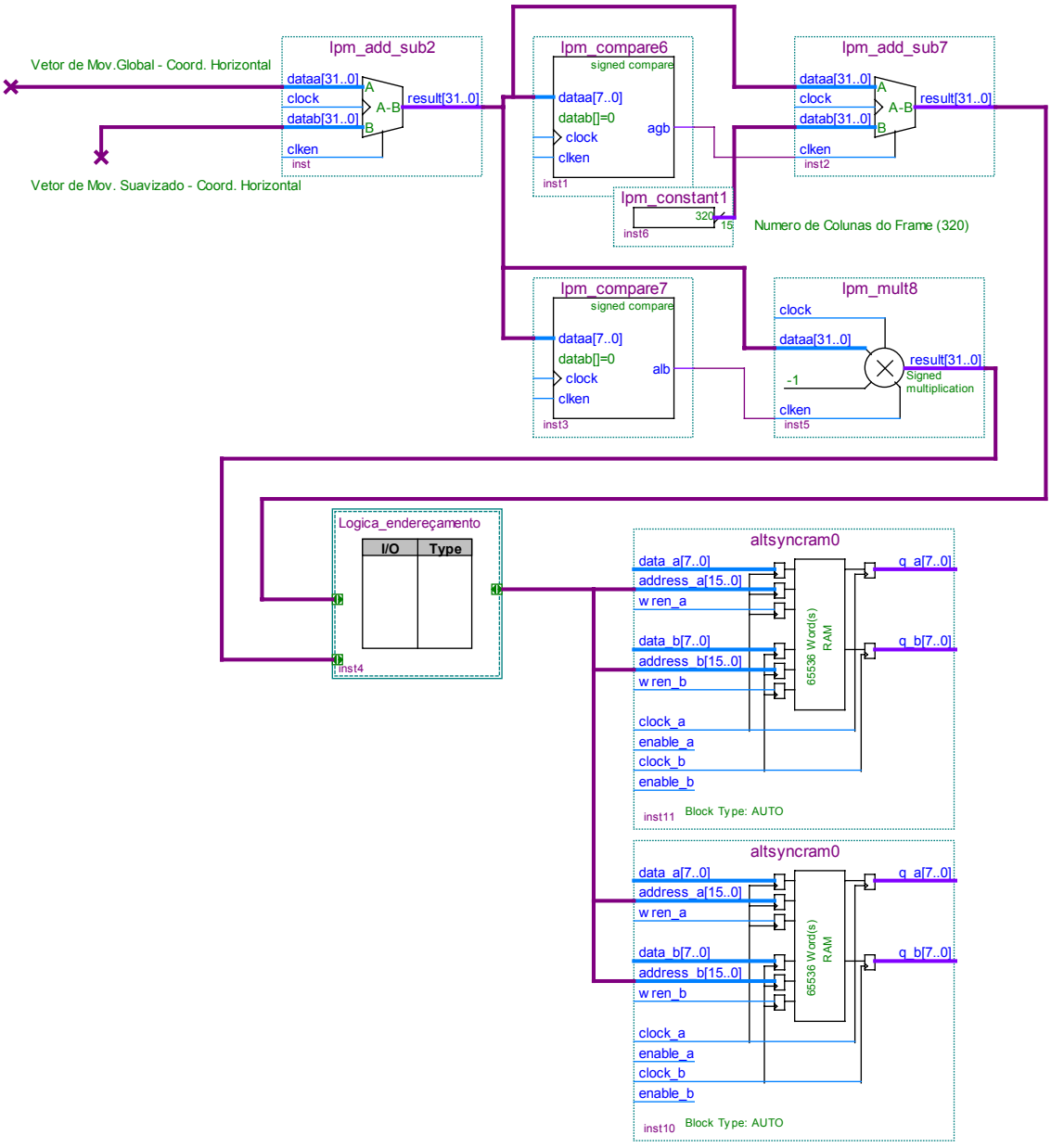


Figura 6.8 – Correção de movimento

6.8 Conclusão

Neste capítulo, foi proposta uma estrutura para a implementação em hardware do sistema de estabilização de imagens. Apesar dos processos de estimação e correção de movimento apresentarem uma maior dificuldade em sua implementação, necessitando de funções mais complexas no cálculo dos vetores de movimento e no endereçamento das memórias, foi possível observar que, vários blocos de processamento do sistema desenvolvido podem ser implementados através de operadores e funções pré-definidas em hardware, facilmente aplicadas a uma FPGA. As funções mais complexas podem ser implementadas em software, utilizando linguagem C em um microprocessador de alto desempenho (Como, por exemplo, tipo ARM).

Capítulo 7

Conclusões e sugestões para trabalhos futuros

O trabalho apresentou um sistema de estabilização de imagens capaz de trabalhar em tempo real, estimar, classificar, suavizar e corrigir movimentos translacionais e rotacionais entre as imagens de um vídeo. Os movimentos são estimados pela técnica Casamentos de Blocos (CB), com a estratégia de busca n passos. Para os movimentos rotacionais foi criado um modelo Affine, o qual tem seus parâmetros determinados pelos vetores de movimento obtidos pela técnica CB ponderada. A classificação em movimentos translacionais e rotacionais foi realizada através do cálculo da variância dos vetores de movimento estimados pelo CB. Para a suavização de movimento, foi utilizado um filtro média de movimento de ordem 10. A correção dos movimentos foi realizada deslocando as imagens em direção contrária aos vetores de movimento indesejados estimados, baseados nos vetores de movimento suavizados e originais. Foi criado um software em C++ utilizando a biblioteca OpenCV, especializada em processamento de imagem. Diversos testes do sistema, em diferentes situações de captura de imagem, foram realizados. Os resultados

experimentais obtidos e algumas conclusões baseadas nos mesmos, também foram apresentados neste trabalho.

7.1 Principais contribuições

As principais contribuições do trabalho foram:

- Desenvolvimento de um sistema de estabilização totalmente digital;
- Desenvolvimento de um sistema de estabilização capaz de atuar em tempo real e corrigir movimentos translacionais e rotacionais entre as imagens;
- Desenvolvimento e utilização de software em C++ e da biblioteca OpenCV em sistemas de estabilização de imagens;
- Desenvolvimento de uma estrutura básica em hardware do sistema de estabilização de imagem desenvolvido;
- Publicação e apresentação do trabalho "Estabilização digital de imagens em tempo real em seqüências de vídeo" no II Workshop de Visão Computacional – WVC 2006.

7.2 Sugestões para trabalhos futuros

As sugestões para trabalhos futuros são:

- Evolução, adequação e estudo da estrutura em hardware apresentada do sistema desenvolvido;
- Implementação de um hardware dedicado para o sistema de estabilização de imagens;
- Utilização de diferentes técnicas de estimação de movimento ou critérios de combinação do CB;
- Utilização de filtros adaptativos para a suavização de imagem;
- Utilização de técnicas com mosaico de imagens ou dilatação e cortes destas, evitando bordas sem informações nas imagens estabilizadas;
- Utilização do processo de estabilização de imagens em sistemas aplicados de visão computacional.

Referências

BAPTISTA, E. (2000). *Estabilizando Trepidações* : Revista Zoom Magazine. Disponível em: <http://www.fazendovideo.com.br>.Acessado em 22. Maio. 2006

BROOKS, A.C. (2003), *Real-Time Digital Image Stabilization*: EE 420 Image Processing Computer Project Final Paper.

CHANG, J.Y. et al. (2002). *Digital Image Translational and Rotational Motion Stabilization using Optical Flow Technique*: IEEE Transactions on Consumer Electronics, Vol. 48, Nº 1.

CHEN,T.(2000). *Video Stabilization Algorithm Using a Block-Based Parametric Motion Model:* EE392J Project Report, Stanford University, USA.

CHOI, K.S. et. al.(2000), *An Efficient digital Image Stabilizing Technique for mobile Video Communications* : Department of Electronics Engineering, Korea University, Korea.

DURIC, Z.; ROSENFELD, A.(1996), *Image Sequence Stabilization in Real Time*: Academic Press Limited, USA

ENGELSBERG, A.; SCHMIDT, G. (1999). *A comparative review of digital image stabilizing algorithms for mobile video communications:* IEEE Transactions on Consumer Electronics, Vol. 45, Nº 3.

FERREIRA, R.;CARDOSO, B.; CARVALHO, F.(2001). *Integração Homen-Máquina por detecção de movimento* : PT Inovação, Portugal.

GONZALES, R. C.; WOODS, R. E.(1993), *Digital Image Processing*: Prentice Hall

GUESTRIN, C.; COZMAN,F.; KROTKOV, E.(1998). *Fast Software Image Stabilization whit Color Registration*: Proceeding of the 1998 IEEE/RSJ , Intl. Conference on Intelligent Robots and Systems Victoria, B.C.Canada.

HANDBOOK: SteadiCam Manual (2001).
Disponível em <http:// www.seaticam.com/setadicammanual.pdf> Acessado em 10.Maio.2006.

HANSEN, M. et al. (1994). *Real Time Scene Stabilization and Mosaic Construction:* David Scarnoff Research Center, Priceton, NJ.

KO, S.J. et al. (1999). *Fast Digital Image Stabilizer Based on Gray – Coded Bit – Plane Matching:* IEEE Transaction on Consumer Electronics Vol. 45, Nº 3.

KURAZUME R.; HIROSE, S. (2000), *Development of Image Stabilization System for Remote Operation of walking robots*: Proceedings of the 2000 IEEE International Conference on Robotics e Automation ,San Francisco, CA.

LITVIN, A.;KONRAD, J.; KARL, W. C.(2003). *Probabilistic video stabilization using Kalman filtering and mosaicking* : Symposium on Electronic Imaging, Image and Video Communications and Proc, Santa Clara, CA, USA.

MORIMOTO, C.; CHELLAPA, R. (1996). *Fast Electronic Digital Image Stabilization:* IEEE Proceedings of ICPR `96, p.284-288.

MORIMOTO, C.; CHELLAPA, R. (1995). *Automatic Digital Image Stabilization:* Computer Vision Laboratory, Center for Automation Research, University of Maryland.

REFERENCE MANUAL: Open Source Computer Vision Library (2001). Intel Corporation, USA.

SMITH, W. (1998). *The Scientist and Engineer's Guide to Digital Signal Processing*. Disponível em:<http://www.dps.com>. Acesso em: 10. Maio. 2006.

TEARE, S.W. et al. (2004). *Embedded Controller Based Image Stabilizer:* 2004, AAS 204 th Meeting, Seção 10 Instrumentation, Ground-Based.

TEKALP,A.M. (1995). *Digital Video Processing*: Prentice Hall Signal Processing Series. Upper Saddle River ,NJ, Cap. 6, p. 95 – 116.

UOMORI, K. et. al.(1990), *Automatic Image Stabilizing System by Full-Digital Signal Processing:* IEEE Transactions on Consumer Electronics, Vol. 36, No. 3.

WELCH, G.; BISHOP, G.(2004), *An Introduction to the Kalman Filter*: Department of Computer Science University of North at Chapel Hill

ZHU, Z. et al. (1998). *Camera Stabilization Based on 2.5D Motion Estimation and Inertial Motion Filtering:* 1998 International Conference on Intelligent Vehicles.

APÊNDICE A – Software do sistema de estabilização de imagem desenvolvido

- Captura da imagem gerada pela WebCam

```
cap = cvCaptureFromCAM(CV_CAP_MIL);
cvNamedWindow("WebCam",CV_WINDOW_AUTOSIZE);
ncol= cvGetCaptureProperty(cap,CV_CAP_PROP_FRAME_HEIGHT);
nlin= cvGetCaptureProperty(cap,CV_CAP_PROP_FRAME_WIDTH);
im = cvCreateImage(cvSize(nlin,ncol),IPL_DEPTH_8U, 3);
im1 = cvCreateImage(cvSize(nlin,ncol),IPL_DEPTH_8U, 3);
im2 = cvCreateImage(cvSize(nlin,ncol),IPL_DEPTH_8U, 3);
im3 = cvCreateImage(cvSize(nlin,ncol),IPL_DEPTH_8U, 1);
im4 = cvCreateImage(cvSize(nlin,ncol),IPL_DEPTH_8U, 1);

while(cap)
{

cvGrabFrame(cap);
camera=cvRetrieveFrame( cap );
cvCvtColor(camera,im1, CV_BGR2HSV);
cvConvertImage( im1, im2, 1);
cvSplit( im2, 0, 0 , im3,0);

if(i>1)
{

Icurr = cvGetMat( im3, Icurr, 0, 0 );
while (j<76800) // WIDTH * HEIGHT
{
mat2[j]=Icurr->data.ptr[j];
j=j+1;
}

cvShowImage("WebCam",Icurr);
cvWaitKey(1);

j=0;
o=0;
y=0;
w=0;
```

```
for (y=0;y<240;y++)
{
 o=320*y;
 for (g=0;g<320;g++)
 {
        Fcurr[y] [g]=mat2[g+o];
        Fcurr[y] [g] = Fcurr[y] [g] / 255;
 }

}
```

- Cálculo da diferença entre imagens para a detecção de grandes variações.

```
o=0;
for (y=0;y<240;y++)
{
 o=320*y;
 for (g=0;g<320;g++)
 {
        out[g+o] = (0.7*Fcurr[y] [g] -((1-0.7)*Fant[y] [g]))*255;
        if(out[g+o]>=250)
        {
        outant++;
        }
 }
}


if (outant>sinthreshold2)
{
diferenca = 1;
o=0;
outant = 0;
for (y=0;y<240;y++)
{
  o=320*y;
 for (g=0;g<320;g++)
 {
        out[g+o] = Fcurr[y] [g]*255;
 }
}
}
else
{
diferenca = 0;
outant=0;
}// Fim da estimação de grandes variações
```

- Estimação de movimento (n step) entre o Frame atual e de referencia (frame estabilizado)

```
if (diferenca ==0)
{
o=0;
y=0;
 for (y=1;y<14;y++)
 {
          for (o=1;o<19;o++)
          {
      for (t=0;t<16;t++)
                {
                        for (p=0;p<16;p++)
                        {
                         bloco2[t] [p]=Fcurr[16*y+t] [16*o+p];
                        }
                }
      for (t=0;t<16;t++)
                {
                        for (p=0;p<16;p++)
                        {
                         vetor2[w]=bloco2[p] [t];
                         w=w+1;
                        }
                }
          w=0;
          h=0;
          q=0;
          passo = 16;
          passo2 = passo/2;
          endx = 0;
          endy=0;
          endxf=0;
          endyf=0;
          endxw = 0;
          endyw=0;
          endxfw=0;
          endyfw=0;
          auxend=0;
```

```
    for (step=1;step<5;step++)
    {
     h=0;
     q=0;
     for (k=-passo2;k<=passo2;k+=passo2)
     {
      for (d=-passo2;d<=passo2;d+=passo2)
      {
       for (t=0;t<16;t++)
              {
               for (p=0;p<16;p++)
               {
                bloco1[t] [p]=Fant[16*y+t+k+endx] [16*o+p+d+endy];
               }
              }
for (t=0;t<16;t++)
              {
               for (p=0;p<16;p++)
               {
                vetor1[w]=bloco1[p] [t];
                w=w+1;
               }
              }

             w=0;
      smse = 0;
```

- Cálculo do critério MSE

```
             for (w=0;w<256;w++)
             {
               mse[w]=vetor2[w]-vetor1[w];
               mse[w]=mse[w]*mse[w];
               smse = smse + mse[w];
             }
             atual= sqrt(smse);
             w=0;
```

- Casamento de blocos ponderado

```
             for (w=0;w<256;w++)
             {
               msew[w]=vetor2[w]-vetor1[w];
               msew[w] = msew[w] * mat4[w];
               msew[w]=msew[w]*msew[w];
               smsew = smsew + msew[w];
             }
```

```
                atualw= sqrt(smsew);
                w=0;

                if (atual<MSE)
                {
                 MSE = atual;
                 index = h;
                }
                h++;
                if (atualw<MSEw)
                {
                 MSEw = atualw;
                 indexw = q;
                }
                q++;
            }
           }


     h=0;
     MSE = 1000;
     auxend = index/3;
     endx = -passo2 + ((index-auxend*3)*passo2);
     endy = -passo2 + auxend*passo2;
     endxf = endxf+endx;
    endyf = endyf+endy;
     q=0;
     MSEw = 1000;
     auxend = indexw/3;
     endxw = -passo2 + ((indexw-auxend*3)*passo2);
     endyw = -passo2 + auxend*passo2;
     passo2=passo2/2;
     endxfw = endxfw+endxw;
     endyfw = endyfw+endyw;
          }
          u[y] [o] = endxf;
          v[y] [o] = endyf;
          uw[y] [o] = endxfw;
          vw[y] [o] = endyfw;
          }
 }
// Fim da estimação de movimento
```

- Construção dos vetores com os movimentos estimados

```
q=0;
l=0;
for (o=1;o<19;o++)
{
```

```
            for (y=1;y<14;y++)
            {vetor3[l] = u[y] [o];
             vetor4[q] = v[y] [o];
             l++;
             q++;
    }
}
l=0;
q=0;
Us=0;
Vs=0;
```

- Classificação de movimento

```
for (p=0;p<234;p++)
{
 Us=Us+vetor3[p];
 Vs=Vs+vetor4[p];
}

Us=Us/234;
Vs=Vs/234;
cvWaitKey(1);
```

- Cálculo da variância dos vetores de movimento estimados

```
aux2 = 0;
p=0;
l=0;
Us = 0;
Vs = 0;
for (p=0;p<234;p++)
{
 Us=Us+vetor3[p];
 Vs=Vs+vetor4[p];
}

Us=Us/234;
Vs=Vs/234;
p=0;
Uvar = 0;
Vvar = 0;
for (p=0;p<234;p++)
{
 Uvar=Uvar+(vetor3[p]*vetor3[p]);
 Vvar=Vvar+(vetor4[p]*vetor4[p]);
}
```

```
Uvar = Uvar - 234*(Us*Us);
Vvar = Vvar - 234*(Vs*Vs);
p=0;
Uvar = Uvar/233;
Vvar = Vvar/233;
```

- Decisão entre movimentos rotacionais e translacionais

```
if (Uvar < varthreshold && Vvar < varthreshold)
{
// Movimento translacional
metodo=1;
}
else
{
// Movimento rotacional
 metodo=0;
 for (o=1;o<19;o++)
 {
  for (y=1;y<14;y++)
  {
   vetor3w[l] = uw[y] [o];
   vetor4w[q] = vw[y] [o];
   l++;
   q++;
  }
 }
 Us=0;
 Vs=0;
 for (p=0;p<234;p++)
 {
  Us=Us+vetor3w[p];
  Vs=Vs+vetor4w[p];
 }
 Us=Us/234;
 Vs=Vs/234;
}
```

- Calculo vetor de movimento global

```
Uvar=0;
Vvar=0;

us = cvRound(Us);
vs = cvRound(Vs);
```

- Vetor de valores anteriores de movimento para utilização no processo de suavização de movimento

```
for (p=0;p<9;p++)
{
        valorvs[p]=valorvs[p+1];
        valorus[p]=valorus[p+1];
}
valorvs[9] = vs;
valorus[9] = us;
```

- Cálculo dos parâmetros para o modelo AFFINE

```
h=0;
for (t=1;t<14;t++)
{
 y=16*t+8;
 for (q=1;q<19;q++)
 {
  o=16*q+8;
  Y=y+vw[t] [q];
  X =o+uw[t] [q];
  vetorXY[h]=X;
  Hxy[h] [0]=o;
  Hxy[h] [1]=-y;
  Hxy[h] [2]=1;
  Hxy[h] [3]=0;
  h++;
  vetorXY[h]=Y;
  Hxy[h] [0]=y;
  Hxy[h] [1]=o;
  Hxy[h] [2]=0;
  Hxy[h] [3]=1;
  h++;
 }
}



cvSetData(temp,Hxy,32);
cvSetData(temp2,vetorXY,8);
cvSolve( temp, temp2, temp3, CV_SVD );

alpha =temp3->data.db[0];
beta =temp3->data.db[1];
xoffset =temp3->data.db[2];
yoffset =temp3->data.db[3];
theta = atan(beta/alpha);
```

- Vetor de valores anteriores de parâmetros AFFINE para utilização no processo de suavização de movimento

```
for (p=0;p<9;p++)
{
 xoffsetvet[p]=xoffsetvet[p+1];
 yoffsetvet[p]=yoffsetvet[p+1];
 thetavet[p]=thetavet[p+1];
}

xoffsetvet[9]=xoffset;
yoffsetvet[9]=yoffset;
thetavet[9]=theta;
```

- Suavização de movimento para movimentos translacionais (Filtro Média de Movimento)

```
if (metodo==1)
{
 convus[0] = valorus[0] * filter;
 convus[1] = valorus[1] * filter+ convus[0];
 convus[2] = valorus[2] * filter+ convus[1];
 convus[3] = valorus[3] * filter+ convus[2];
 convus[4] = valorus[4] * filter+ convus[3];
 convus[5] = valorus[5] * filter+ convus[4];
 convus[6] = valorus[6] * filter+ convus[5];
 convus[7] = valorus[7] * filter+ convus[6];
 convus[8] = valorus[8] * filter+ convus[7];
 convus[9] = valorus[9] * filter+ convus[8];
 convus[10] = convus[9] - convus[0];
 convus[11] = convus[9] - convus[1];
 convus[12] = convus[9] - convus[2];
 convus[13] = convus[9] - convus[3];
 convus[14] = convus[9] - convus[4];
 convus[15] = convus[9] - convus[5];
 convus[16] = convus[9] - convus[6];
 convus[17] = convus[9] - convus[7];
 convus[18] = convus[9] - convus[8];

 convvs[0] = valorvs[0] * filter;
 convvs[1] = valorvs[1] * filter+ convvs[0];
 convvs[2] = valorvs[2] * filter+ convvs[1];
 convvs[3] = valorvs[3] * filter+ convvs[2];
 convvs[4] = valorvs[4] * filter+ convvs[3];
 convvs[5] = valorvs[5] * filter+ convvs[4];
 convvs[6] = valorvs[6] * filter+ convvs[5];
 convvs[7] = valorvs[7] * filter+ convvs[6];
```

```
convvs[8] = valorvs[8] * filter+ convvs[7];
convvs[9] = valorvs[9] * filter+ convvs[8];
convvs[10] = convvs[9] - convvs[0];
convvs[11] = convvs[9] - convvs[1];
convvs[12] = convvs[9] - convvs[2];
convvs[13] = convvs[9] - convvs[3];
convvs[14] = convvs[9] - convvs[4];
convvs[15] = convvs[9] - convvs[5];
convvs[16] = convvs[9] - convvs[6];
convvs[17] = convvs[9] - convvs[7];
convvs[18] = convvs[9] - convvs[8];

Usuav = cvRound(convus[9]);
Vsuav = cvRound(convvs[9]);
```

- Cálculo da distancia (movimento) entre as imagens

```
difmv = 0;
difmv = sqrt((us*us)+(vs*vs));
```

- Cálculo do movimento global indesejado

```
if (us>=-2 && us<=2 && vs>=-2 && vs<=2)
{
 uf=-us;
 vf=-vs;
}
else
{
 uf = Usuav - us;
 vf = Vsuav - vs;
}
```

- Compensação do movimento e determinação do Frame estabilizado

```
t=0;
q=0;
if (uf>=0 && vf>=0)
{
 for (y=0;y<240-vf;y++)
 {
  t=0;
  for (g=0;g<320-uf;g++)
  {
   t++;
   out[q] = Fcurr[y] [g] *255;
```

```
  q++;
 }
 if (t<320)
 {
  while (t<320)
  {
   out[q] = 0;
   q++;
   t++;
  }
 }
}
if (q<76800)
{
 while (q<76800)
 {
  out[q]=0;
  q++;
 }
}
}

else if(uf<0 && vf>=0)
{
 for (y=0;y<240-vf;y++)
 {
  t=0;
  for (g=0;g<-uf;g++)
  {
   out[q] = 0;
   q++;
  }
  for (g=-uf;g<320;g++)
  {
   t++;
   out[q] = Fcurr[y] [g] *255;
   q++;
  }
 }
 if (q<76800)
 {
   while (q<76800)
   {
          out[q]=0;
    q++;
   }
 }
}

else if(uf>=0 && vf<0)
```

```
{
 for (y=0;y<-vf*320;y++)
 {
  out[q] =0;
  q++;
 }
 for (y=-vf;y<240;y++)
 {
  t=0;
  for (g=0;g<320-uf;g++)
  {
   t++;
   out[q] = Fcurr[y] [g] *255;
   q++;
  }
  if (t<320)
  {
   while (t<320)
   {
          out[q]=0;
    q++;
    t++;
   }
  }
 }
}

else if(uf<0 && vf<0)
{
 for(y=0;y<-vf*320;y++)
 {
  out[q] = 0;
  q++;
 }
 for (y=-vf;y<240;y++)
 {
  t=0;
  for (g=0;g<-uf;g++)
  {
   out[q] = 0;
   q++;
  }
  for (g=-uf;g<320;g++)
  {
   t++;
   out[q] = Fcurr[y] [g] *255 ;
   q++;
  }
 }
 if (q<76800)
```

```
  {
   while (q<76800)
   {
    out[q]=0;
    q++;
   }
  }
 }
```

```
} //fim metodo-1 (movimento translacional)
```

- Suavização de movimento para movimentos rotacional (Filtro Média de Movimento)

```
else
{
 l=0;
 q=0;
 sumtheta = sumtheta + theta;
 sumxoffset = sumxoffset + xoffset;
 sumyoffset = sumyoffset + yoffset;

convxoffset[0] = xoffsetvet[0] * filter;
 convxoffset[1] = xoffsetvet[1] * filter+ convxoffset[0];
 convxoffset[2] = xoffsetvet[2] * filter+ convxoffset[1];
 convxoffset[3] = xoffsetvet[3] * filter+ convxoffset[2];
 convxoffset[4] = xoffsetvet[4] * filter+ convxoffset[3];

 convxoffset[5] = xoffsetvet[5] * filter+ convxoffset[4];
 convxoffset[6] = xoffsetvet[6] * filter+ convxoffset[5];
 convxoffset[7] = xoffsetvet[7] * filter+ convxoffset[6];
 convxoffset[8] = xoffsetvet[8] * filter+ convxoffset[7];
 convxoffset[9] = xoffsetvet[9] * filter+ convxoffset[8];

 convxoffset[10] = convxoffset[9] - convxoffset[0];
 convxoffset[11] = convxoffset[9] - convxoffset[1];
 convxoffset[12] = convxoffset[9] - convxoffset[2];
 convxoffset[13] = convxoffset[9] - convxoffset[3];

 convxoffset[14] = convxoffset[9] - convxoffset[4];
 convxoffset[15] = convxoffset[9] - convxoffset[5];
 convxoffset[16] = convxoffset[9] - convxoffset[6];
 convxoffset[17] = convxoffset[9] - convxoffset[7];
 convxoffset[18] = convxoffset[9] - convxoffset[8];


 convyoffset[0] = yoffsetvet[0] * filter;
 convyoffset[1] = yoffsetvet[1] * filter+ convyoffset[0];
 convyoffset[2] = yoffsetvet[2] * filter+ convyoffset[1];
```

```
convyoffset[3] = yoffsetvet[3] * filter+ convyoffset[2];
convyoffset[4] = yoffsetvet[4] * filter+ convyoffset[3];

convyoffset[5] = yoffsetvet[5] * filter+ convyoffset[4];
convyoffset[6] = yoffsetvet[6] * filter+ convyoffset[5];
convyoffset[7] = yoffsetvet[7] * filter+ convyoffset[6];
convyoffset[8] = yoffsetvet[8] * filter+ convyoffset[7];
convyoffset[9] = yoffsetvet[9] * filter+ convyoffset[8];

convyoffset[10] = convyoffset[9] - convyoffset[0];
convyoffset[11] = convyoffset[9] - convyoffset[1];
convyoffset[12] = convyoffset[9] - convyoffset[2];
convyoffset[13] = convyoffset[9] - convyoffset[3];

convyoffset[14] = convyoffset[9] - convyoffset[4];
convyoffset[15] = convyoffset[9] - convyoffset[5];
convyoffset[16] = convyoffset[9] - convyoffset[6];
convyoffset[17] = convyoffset[9] - convyoffset[7];
convyoffset[18] = convyoffset[9] - convyoffset[8];


convtheta[0] = thetavet[0] * filter;
convtheta[1] = thetavet[1] * filter+ convtheta[0];
convtheta[2] = thetavet[2] * filter+ convtheta[1];
convtheta[3] = thetavet[3] * filter+ convtheta[2];
convtheta[4] = thetavet[4] * filter+ convtheta[3];

convtheta[5] = thetavet[5] * filter+ convtheta[4];
convtheta[6] = thetavet[6] * filter+ convtheta[5];
convtheta[7] = thetavet[7] * filter+ convtheta[6];
convtheta[8] = thetavet[8] * filter+ convtheta[7];
convtheta[9] = thetavet[9] * filter+ convtheta[8];

convtheta[10] = convtheta[9] - convtheta[0];
convtheta[11] = convtheta[9] - convtheta[1];
convtheta[12] = convtheta[9] - convtheta[2];
convtheta[13] = convtheta[9] - convtheta[3];

convtheta[14] = convtheta[9] - convtheta[4];
convtheta[15] = convtheta[9] - convtheta[5];
convtheta[16] = convtheta[9] - convtheta[6];
convtheta[17] = convtheta[9] - convtheta[7];
convtheta[18] = convtheta[9] - convtheta[8];
```

- Cálculo dos parâmetros AFFINE para os movimentos indesejados

```
thetaind = convtheta[9]-theta;
```

```
 if (theta<=-2 && theta>=2)
 {
  alphaind= cos(thetaind);
  betaind = sin(thetaind);
 }
 else
 {
 alphaind= cos(-theta);
 betaind = sin(-theta);
 }

xoffsetind = xoffset - convxoffset[9];
yoffsetind = yoffset - convyoffset[9];
```

- Correção do movimento rotacional

```
 for (y =0;y<240;y++)
 {
  for (g=0;g<320;g++)
  {
   coorx = (alphaind * y) +(-betaind *g)  -xoffsetind;
   coory = (betaind * y) + (alphaind *g)  -yoffsetind;
   Uvar=coorx;
   Vvar=coory;
   o=cvRound(Uvar);
   w=cvRound(Vvar);
   Uvar=0;
   Vvar=0;
   if (o<0||w<0||o>=240 || w>=320)
   {
    out[q] = 0;
   }
   else
   {
    out[q] = Fcurr[o] [w]*255;
   }
  q++;
  }
 }

}// fim (movimento rotacional)
}
```

- Apresentação da imagem estabilizada

```
cvSetImageData( teste, out, 320);
```

```
cvShowImage("WebCam Estabilizada",teste);
Uvar = 0;
Vvar = 0;
h=0;
y=0;
o=0;

}// end i - numero de frames
```

- Escolha da imagem de referência

```
t=0;
if (i == 0 || somareset>reset)
{
 somareset = 0;
 im4 = cvCloneImage( im3 );
 Iant= cvGetMat( im4, Iant, 0, 0 );
 while (t<76800)
 {
  mat3[t]= Iant->data.ptr[t];
  t+=1;
 }
 t=0;
 o=0;
 y=1;
 for (y=0;y<240;y++)
 {
  o=320*y;
  for (g=0;g<320;g++)
  {
   Fant[y] [g]=mat3[g+o];
   Fant[y] [g] = Fant[y] [g] / 255;
  }
 }
}

if(difmv < sinthreshold && i>=1 && somareset<=reset )
{
// Frame de referencia = Frame atual suavizado
 t=0;
 o=0;
 y=1;
 for (y=0;y<240;y++)
 {
  o=320*y;
  for (g=0;g<320;g++)
  {
   Fant[y] [g] = out[g+o];
```

```
   Fant[y] [g] = Fant[y] [g] / 255;
  }
 }
} // Fim da escolha da imagem de referencia
somareset++;
h=0;
i=i+1;
}  // Fim da captura de imagem
fclose(pf);
cvReleaseCapture(&cap);
}// Fim do processo de estabilização
```

APÊNDICE B – Biblioteca OpenCV

OpenCV significa Intel® Open Source Computer Vision Library. É uma coleção de funções em C e C++ que, implementam alguns algoritmos mais comuns em processamento de imagens e visão computacional. Pode ser livremente utilizada em aplicações comerciais e não comerciais.

As funções são divididas em várias bibliotecas como: CVCAM, CXCORE, CV, HighGUI, CV, CVAUX, etc.

Neste trabalho foram utilizadas as funções presentes em quatro bibliotecas. As funções e bibliotecas utilizadas são detalhadas abaixo:

CVCAM

CvCam é uma plataforma universal para o processamento de vídeo gerados por uma câmera digital. Possui uma simples interface de programação para leitura e controle de seqüências de vídeo, processamento e armazenamento de frames.

Funções utilizadas

cap = cvCaptureFromCAM(CV_CAP_MIL); - Captura da imagem gerada pela câmera digital.

cvReleaseCapture(&cap); - Finaliza a estrutura criada para a captura da imagem gerada pela câmera.

CV

CV é a biblioteca composta pelas funções de processamento de imagem mais comumente utilizadas, como: operações morfológicas, filtros e conversão de cores, Histogramas, Fluxo Óptico, etc.

Funções utilizadas:

cvCvtColor(camera,im1, CV_BGR2HSV); - Conversão de Cores

im4 = cvCloneImage(im3); - Cópia de imagens

CXCORE

CXCORE é composta por funções que realizam operações básicas, como cálculos estatísticos, cálculos aritméticos e acesso a elementos de vetores, etc.

Funções utilizadas:

cvSetImageData(teste, out, 320); - atribui dados aos pixeis da imagem

cvSetData(temp,Hxy,32); - Atribui o dados ao vetor

cvSolve(temp, temp2, temp3, CV_SVD); - Resolução de sistemas lineares ou problemas de mínimo quadrados.

cvSplit(im2, 0, 0 , im3,0); - Divide os canais da imagem e separa um canal específico

im4 = cvCreateImage(cvSize(nlin,ncol),IPL_DEPTH_8U, 1); - Cria a estrutura de uma imagem e atribui dados a ela.

HiguiGUI

A biblioteca HighGUI é formada por funções de abertura e fechamento de janelas, carregamento e gravação de imagens, etc.

Funções utilizadas:

cvNamedWindow("WebCam",CV_WINDOW_AUTOSIZE); - Cria uma janela com o nome especificado.

ncol= cvGetCaptureProperty(cap,CV_CAP_PROP_FRAME_HEIGHT); - Captura as propriedade do Frame.

cvShowImage("WebCam Estabilizada",teste); - Mostra a imagem numa janela especificada.

cvConvertImage(im1, im2, 1); - Converte um imagem a outra com translação.