METODOLOGIA PARA DETECÇÃO RÁPIDA DE MOVIMENTO EM SEQUÊNCIAS DE IMAGENS

Isaura Nelsivânia Sombra Oliveira

Tese apresentada ao Instituto de Física de São Carlos, da Universidade de São Paulo, para obtenção do título de Doutora em Ciências: Física Aplicada.

Orientador: Prof. Dr. Valentin Obac Roda

ok

São Carlos

2003

USP/IFSC/SBI

8-2-001531

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

Oliveira, Isaura Nelsivânia Sombra

Metodologia para detecção rápida de movimento em seqüências de imagens/ Isaura Nelsivânia Sombra Oliveira. — São Carlos, 2003.

Número de páginas 180p.

Tese (Doutorado) - Instituto de Física de São Carlos, 2003.

Orientador: Prof. Dr. Valentin Obac Roda.

1. Detecção de Movimento. 2. Fluxo ótico. 3. Visão em tempo real

UNIVERSIDADE DE SÃO PAULO
Instituto de Física de São Carlos

Av. Trabalhador São-carlense, 400
Caixa Postal 369
13560-970 - São Carlos - SP - Brasil

MEMBROS DA COMISSÃO JULGADORA DA TESE DE DOUTORADO DE ISAURA NELSIVÂNIA SOMBRA OLIVEIRA, APRESENTADA AO INSTITUTO DE FÍSICA DE SÃO CARLOS, UNIVERSIDADE DE SÃO PAULO, EM 30-05-2003.

COMISSÃO JULGADORA:

Prof. Dr. Valentin Obac Roda (Presidente) – EESC/USP

Prof. Dr. Paulo Mazzoncini de Azevedo Marques – FMRP/USP

Profa. Dra. Annie France Frere Slaets – UMC

Profa. Dra. Agma Juci Machado Traina – ICMC/USP

Dr. Attílio Cucchieri – IFSC/ USP

Dedico este trabalho à Sophia pelas horas em que não brincamos.....

.e que você ainda me emprestou seus brinquedos.

Ao Pedro que em breve estará no mundo extra-uterino,

para dividirmos as tais horas e os tais brinquedos...

AGRADECIMENTOS

Primeiro, agradeço a Deus, o doutor dos doutores, sempre tão simples e bem-humorado, por todos os momentos de amizade e companheirismo nestes meus anos de vida, que incluem este doutoramento.

Aos meus pais, Nelson e Lourdinha, que sem diplomas ou títulos, souberam ser educadores e amigos de todos nós, seus filhos.

Ao Marcelo, meu predileto colega de laboratório, tão predileto que se tornou meu marido, obrigada pela tolerância em ouvir a palavra tese tantas vezes ao dia.

Ao Professor Valentin, pela oportunidade deste trabalho!

Aos meus irmãos, pelo incentivo de sempre!

Aos meus sobrinhos, de quem tenho tantas saudades!!!

Aos colegas do laboratório José Pott, Leonardo Scarabelli, Ivan Bizari, Fabiriciu Benini, e técnicos Ailton, Silvio (*in memorian*), Petrússio, João Carlos, e José Luís pelas ajudas e pelo ambiente descontraído.

Ao Maxmiliam Luppe, Ricardo Villela e Mário Gazziro pelos papos *iluminados* a respeito do meu tema.

Ao Rodrigo Gonzalez Modugno pelo suporte na organização das imagens.

Às amiga Dona Nair Appel, Maria Helena Gomes, Luciane Sarkis, Elvira Majanela, Lena Di Thomazzo pelas ajudas em sistema de rodízio com a Sophia.

Às berçaristas e funcionárias da Unidade de Apoio à Criança da Universidade Federal de São Carlos que acolheram minha nenem com carinho e atenção durante tantas longas manhãs.

Ao pessoal da xerox do CAASO, pela presença alegre nos dias dos pós-graduados.

À nossa caríssima Wladerez Gounella – Wla – pela prontidão, disposição e grande tolerância no atendimento aos pós-graduandos.

Ao CNPq pelo suporte financeiro. Ao Instituto de Física de São Carlos pela infra-estrutura e organização disponibilizada aos pós-graduandos.

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

Índice

1 INTRODUÇÃO 1

1.1 Objetivos e Contribuições do Trabalho 3

1.2 Organização da Tese 4

2 REVISÃO BIBLIOGRÁFICA SOBRE FLUXO ÓTICO EM DETECÇÃO DE MOVIMENTO 5

2.1 Introdução 5

2.2 Campo de Movimento e Fluxo ótico 9

2.3 Problemas de Abertura e Oclusão 14

2.4 Métodos baseados na Equação do Fluxo ótico 17

2.4.1 Métodos baseados em gradiente 17

2.4.2 Métodos baseados em correlação 18

2.4.3 Métodos baseados em filtros sintonizados de velocidade 20

2.5 Revisão das Técnicas 21

2.5.1 Método de Horn e Schunck 21

2.5.2 Métodos Diferenciais de Segunda Ordem 23

2.5.3 Métodos de Movimento de Blocos 24

2.6 Conclusão 24

3 REVISÃO SOBRE MÉTODOS DE DETECÇÃO DE BORDAS 26

3.1 Introdução 26

3.2 Operadores Gradiente 27

3.3 Operador Laplaciano. 30

3.4 Detectores de bordas para determinação de movimento 34

3.5 Conclusões 35

4 REDUÇÃO DO TEMPO DE DETECÇÃO DE MOVIMENTO 36

4.1 Introdução 36

4.2 Implementação do Algoritmo 36

4.3 Conclusões 42

5 RESULTADOS 43

5.1 Introdução 43

5.2 Validação do Algoritmo 43

5.3 Apresentação dos Resultados 48

5.3.1 Imagens sintéticas 48

5.3.2 Imagens reais com variáveis de entrada controladas 49

5.3.3 Imagens reais em um barco em movimento 51

5.3.4 Outras imagens de teste 53

5.4 Análise comparativa dos resultados 55

5.5 Conclusões 57

6 CONCLUSÃO 58

6.1 Contribuições deste trabalho 58

6.2 Conclusões Gerais 59

6.3 Trabalhos Futuros 60

7 REFERÊNCIAS BIBLIOGRÁFICAS 61

APÊNDICE A: Erro Quadrático 68

APÊNDICE B: Resultados de Aplicação do Método Desenvolvido 69

APÊNDICE C: Listagem do Programa 165

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

RESUMO

Algoritmos de detecção de movimento em seqüências de imagens devem satisfazer os requisitos de precisão, robustez e velocidade de processamento. A forma de combinar estes três itens depende do desenvolvimento do algoritmo e da aplicação a que se destina, sem que os itens de robustez e precisão sejam comprometidos. Neste trabalho investigamos técnicas para detecção do movimento que satisfazem tais requisitos. A técnica escolhida para detecção de movimento foi a do fluxo ótico (FO) devido às suas características de precisão nos resultados. Como esta técnica exige elevado esforço computacional, propõe-se nesta tese uma metodologia que aplica as equações de fluxo ótico em reduzidas áreas da imagem processada. Estas áreas são selecionadas utilizando algumas técnicas de pré-processamento que identificam regiões da imagem com maior probabilidade de movimento presente. Posteriormente a esta identificação são aplicadas as equações de FO nas regiões de interesse. Para avaliação e validação do método proposto, comparam-se os diagramas de agulhas resultantes das áreas reduzidas aos diagramas resultantes de toda a imagem mediante critérios estatísticos e de tempo de processamento envolvido. Os algoritmos são testados utilizando imagens sintéticas e imagens reais.

ABSTRACT

Algorithms for motion detection in image sequences must satisfy the following requirements: accuracy, robustness and speed. The way that accuracy, robustness and speed are combined depends on the algorithm development and on the application. In this work, it has investigated motion detection techniques that satisfy the mentioned requirements. The Optical Flow technique was chosen for the motion detection due to its good performance in the results. As the Optical Flow requires intensive computational load, we propose in this thesis a methodology where Optical Flow Equations are applied in specific areas of the processed image. These areas were selected using pre-processing techniques that identify regions of image with larger motion probability. After the motion areas identification, Optical Flow Equations are applied to the regions of interest. To assess and validate the proposed method, the needle diagrams obtained in the reduced areas are compared with the ones obtained from the whole image according to statistical criteria and processing time. The proposed algorithms have been tested in synthetic and real images.

1 Introdução

Existem várias motivações para estudos de técnicas de detecção de movimento em visão computacional. Dentre elas estão o desenvolvimento de mecanismos de auxílio à navegação para robôs móveis; execução de tarefas perigosas e muitas vezes inacessíveis para o ser humano (exemplo: inspeções visuais microscópicas sobre a motilidade de espermatozóides para análises de reprodução bovina – [PAZINATTO00]); diagnóstico cardíaco baseado em monitoramento dos respectivos movimentos, tarefas monótonas ao ser humano como monitoramento de pedestres em estações de metrô, controle de tráfego rodoviário e prevenção de eventuais tumultos em multidões [BOGHOSSIAN99]. Em muitas destas aplicações, faz-se necessário à implementação de um sistema visual detector de informação acerca do movimento existente no mundo real.

A estimação do movimento que consiste em descrever o movimento existente no plano-objeto ou no plano-imagem, tornou-se um dos principais problemas em processamento digital de imagens, e tem sido objeto de estudo de continuadas pesquisas [HORN81], [NELSON87], [BÜLTHOFF89], [ANANDAN89], [BARRON94], [DUCHON95], [SABE96], [CHHABRA97].

As etapas de estimação do movimento de uma cena basicamente são: detecção da existência de movimento, sua localização na cena e por fim, sua

quantificação, que está relacionada com a distância do objeto à câmera. Para que um algoritmo de reconhecimento de movimento de objetos em uma cena tenha êxito e seja prático do ponto de vista do ambiente do mundo real, é necessário que o mesmo seja robusto na presença de ruído, rápido, principalmente quando se deseja trabalhar em tempo real, e preciso [CAMUS94]. Um bom exemplo é quando se deseja movimentar um robô, cujo processo de visão deva necessariamente desviá-lo de obstáculos e, portanto fornecer resultado basicamente *qualitativo* [NELSON87]. Neste caso não importa o quão refinado é o desvio (quantidade de desvio), e sim unicamente que haja um desvio satisfatório do obstáculo a tempo de evitar a colisão, o que implica num sistema veloz e robusto a ruído, mas não necessariamente preciso.

Apesar de colocarmos a precisão de um algoritmo num grau de exigência inferior aos da robustez e velocidade, escolhemos como ferramenta para análise de movimento um método cuja maior vantagem é a precisão de seus resultados, que é o Fluxo Ótico. Por definição este consiste do conjunto de vetores de velocidade de cada *pixel* da imagem. Em se tratando de um sistema cuja precisão não é prioritária, por que então utilizá-la? Como podemos detectar movimento de forma mais rápida, robusta e precisa? Adotamos a solução de aplicar um pré-processamento, o qual reduz a área útil para aplicarmos o FO, reduzindo assim o tempo de processamento. Desta forma busca-se a garantia dos itens velocidade e precisão. Mas quanto à robustez? Este item dependerá dos algoritmos escolhidos nas associações de métodos. Deste modo, a associação destes pares de técnicas nos possibilitaria uma investigação comparativa e detalhada sobre os resultados obtidos, levando-nos a propor uma metodologia da velocidade de processamento em projetos de detecção de movimento em tempo real baseado em fluxo ótico.

IFSC-USP SERVIÇO DE BIBLIOTECA E INFORMAÇÃO

A presente pesquisa dedica-se ao estudo de aumento de velocidade de processamento em metodologias de detecção de movimento, baseando-se principalmente nas pesquisas prévias de [CABESTAING91] e [HORN81].

O resultado do cálculo de fluxo ótico é apresentado, neste trabalho, utilizando uma imagem denominada Diagrama de Agulhas (DA), que indica a direção e a magnitude do movimento de cada *pixel* nos quadros de uma seqüência. Após o processo e seleção de área segundo técnicas apresentadas nos próximos capítulos, analisaremos o desempenho das respectivas associações de métodos, utilizando o cálculo do erro mínimo quadrático dos vetores de cada DA gerado, em relação ao diagrama de agulhas obtido sem a devida redução de área.

1.1 Objetivos e Contribuições do Trabalho

Desenvolver uma metodologia de detecção rápida do movimento em tempo real, utilizando uma técnica bastante precisa, cujos inconvenientes da alta complexidade computacional e do grande tempo de execução (este é diretamente proporcional ao tamanho das imagens e das máscaras, número de iterações e de operadores para cálculo das derivadas da equação do fluxo ótico) serão atenuados através de sua aplicação apenas em algumas regiões pré-selecionadas na imagem. Estas regiões pré-selecionadas serão aquelas definidas através da detecção das bordas dos objetos em movimento na cena, ou seja, elimina-se a aplicação do cálculo do FO nas regiões que não pertençam às vizinhanças das bordas. A eficiência destas escolhas influencia de forma diretamente a otimização do tempo de execução do programa.

1.2 Organização da Tese

Esta tese está organizada em 6 capítulos, incluindo as referências bibliográficas. A disposição dos próximos capítulos segue a seguinte ordem:

Capítulo 2: Apresenta uma revisão bibliográfica dos principais trabalhos sobre detecção de movimento em imagens, sobretudo os que utilizam o método do fluxo ótico.

Capítulo 3: Revisão dos métodos de detecção de bordas e correlação aplicados a imagens com movimento.

Capítulo 4: Apresenta a formulação proposta neste trabalho, descrevendo o desenvolvimento do algoritmo e sua validação.

Capítulo 5: Apresenta os resultados de aplicação do algoritmo em várias seqüências de imagens e suas respectivas análises e discussões.

Capítulo 6: Apresenta as conclusões gerais e sugestões para trabalhos futuros.

Capítulo 7: Referências Bibliográficas

Apêndice A: Apresenta a definição do Erro Médio Quadrático.

Apêndice B: Apresenta conjuntos de seqüências de imagens e seus respectivos resultados obtidos aplicando o algoritmo posposto.

Apêndice C: Apresenta a listagem do programa desenvolvido.

SC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

2 Revisão bibliográfica sobre fluxo ótico em detecção de movimento

2.1 Introdução

Imagens resultantes de um movimento do objeto diante da câmera, ou da câmera diante do objeto a ser filmado, ou ainda do movimento do objeto e movimento da câmera, apresentam mudanças na imagem que correspondem a estes movimentos reais.

Para definição de Fluxo Ótico (FO), em nível de equações, considere $S_c(x_1,x_2, t)$ a distribuição de intensidade contínua no domínio do espaço e do tempo. Se a intensidade se mantém constante ao longo da trajetória do movimento, tem-se que:

$$\frac{dS_c(x_1, x_2, t)}{dt} = 0 \qquad \textbf{(2.1)}$$

Onde x_1 e x_2 variam de acordo com a trajetória do movimento [TEKALP95]. A **equação 2.1** é uma expressão de derivada total e denota a taxa de variação de intensidade ao longo da trajetória. Usando a regra de derivação da cadeia e

representando as coordenadas x_1 e x_2 na forma de um vetor posição **x** pode-se reescrever esta equação da seguinte maneira:

$$\frac{\partial S_c(x;t)}{\partial x_1} v_1(x;t) + \frac{\partial S_c(x;t)}{\partial x_2} v_2(x;t) + \frac{\partial S_c(x;t)}{\partial t} = 0 \qquad \textbf{(2.2)}$$

onde $v_1 = \frac{d x_1}{dt}$ e $v_2 = \frac{d x_2}{dt}$ são as componentes do vetor de velocidade **v** em termos das coordenadas espaciais contínuas. A **equação 2.2** é conhecida como sendo a equação do fluxo ótico (EFO). Ela pode ser rescrita da forma:

$$\left\langle \nabla S_c(\mathbf{x};t), \mathbf{v}\ (\mathbf{x};t) \right\rangle + \frac{\partial S_c(x;t)}{\partial t} = 0 \qquad \textbf{(2.3)}$$

Onde $< , >$ é simbologia para produto interno e.

$$\nabla S_c(\mathbf{x};t) = \left[\frac{\partial S_c(x;t)}{\partial x_1} \frac{\partial S_c(x;t)}{\partial x_2} \right]^T \qquad \textbf{(2.4)}$$

é o vetor gradiente, ou seja, o vetor cujo módulo e direção, quantificam respectivamente a diferença e a direção de variação da distribuição de intensidade da imagem ao longo do plano x. O símbolo **T** representa o vetor transposto.

Uma inspeção na **equação 2.3** revela que só é possível estimar o vetor de fluxo ótico cuja componente está na direção do gradiente da imagem espacial, a qual denomina-se de Fluxo Normal. A outra componente, que é ortogonal ao gradiente desaparece devido à operação de produto interno. Esta situação é melhor ilustrada na

Figura 2.1, onde todos os vetores cujas extremidades tocam a linha pontilhada satisfazem a equação (2.3).

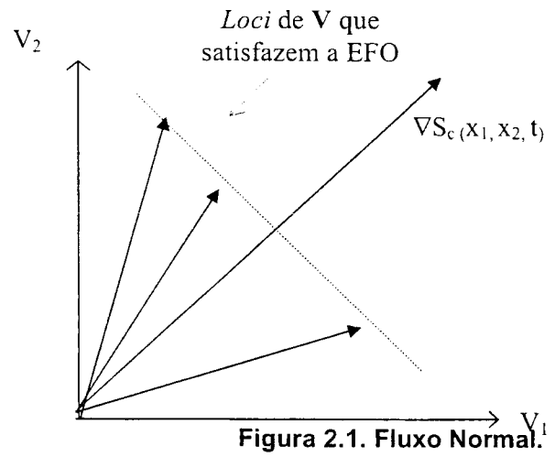


Figura 2.1. Fluxo Normal.

Observa-se que a EFO requer que a intensidade espaço-temporal da imagem seja diferenciável.

A precisão da técnica de Fluxo ótico, no entanto, dependerá diretamente da forma de implementação das derivadas envolvidas em seu cálculo, que por sua vez também são altamente sensíveis à ruídos, podendo vir a comprometer o requisito da robustez do algoritmo.

No que se refere à velocidade do algoritmo, a complexidade que é geralmente citada na bibliografia em relação ao fluxo ótico é função da quantidade de operações necessárias ao seu cálculo, visto que a imagem é composta por muitos *pixels*, o que com freqüência motiva implementações maciçamente paralelas. Uma opção utilizada para diminuição da quantidade de operações é calcular o fluxo ótico (FO) apenas nas áreas de maior contraste onde as medidas de fluxo são mais confiáveis. Nossa proposta de otimização da velocidade, o FO é calculado apenas em algumas áreas do

plano-imagem, sendo que estas áreas são determinadas na vizinhança das bordas dos objetos em movimento, previamente detectadas.

Assim, para as propostas de visão em tempo real, deseja-se encontrar um método de cálculo de FO que seja menos sensível a ruídos no processamento da imagem, que forneça uma saída independente da estrutura na imagem, e que seja computacionalmente veloz [CAMUS97].

Costa Roda e Köberle em [COSTA94] relatam um sistema de reconhecimento de padrões visuais inspirado em sistemas biológicos de visão, que é formado por vários módulos especializados e interconectados, onde cada um é dedicado ao processamento de um atributo específico, como textura, cor, bordas e movimento. Estes módulos são interconectados em níveis de hierarquia para que aumentem a velocidade do processamento. Para a detecção de movimento, eles utilizam o exemplo de uma câmera movida sob controle de um módulo específico, cuja ação é determinada pelo FO da imagem, sendo assim análogo ao movimento dos olhos dos primatas. Se a imagem adquirida pela câmera for monocromática, a equação do FO deverá ser usada em conjunto com uma suavização espaço-temporal apropriada, que requer que o vetor de deslocamento varie suavemente pela vizinhança. Se a imagem for colorida também se pode utilizar o mesmo método, ou então pode-se aplicar a equação de FO a cada uma das bandas (vermelho, verde e azul) separadamente [OHTA90].

Segundo Adrian Low, em [LOW91] quando se restringe a área da imagem, na qual procura-se o movimento, aumenta-se a velocidade da análise a ser feita. A contribuição inovadora deste projeto está relacionada a este raciocínio, pois a proposta consiste num método novo de análise de movimento que utilize FO, auxiliado por técnicas de pré-processamento. Ou seja, detectam-se as bordas dos

objetos em movimento na cena, limitando assim o cálculo do FO unicamente nas regiões da cena onde ocorre movimento, o que diminui o tempo de processamento.

2.2 Campo de Movimento e Fluxo ótico

Quando se tem uma câmera e uma cena a ser filmada, e nesta filmagem ocorre algum tipo de movimento relativo real, tem-se uma correspondente mudança na imagem capturada. Estas mudanças são registradas em nível de *pixel* através de um vetor de velocidade também chamado de campo de velocidade ou Campo de Movimento. O Campo de Movimento (CM) é a projeção em perspectiva da velocidade de um objeto do espaço 3D sobre o plano imagem [HORN86].

Fluxo ótico é o movimento aparente dos padrões de brilho da imagem. Em outras palavras, tendo-se uma seqüência de dois quadros consecutivos, um dado *pix*el posicionado em (x_1 , x_2) e com intensidade de brilho *Sc* no primeiro quadro, e no segundo quadro uma nova intensidade de brilho *Sc*'. Supondo variações de intensidade por todo o plano, podemos calculá-las e assim montar uma matriz de resultado, onde para cada posição da imagem, teremos um vetor que quantifica o respectivo movimento em sua intensidade e direção. A esta matriz de resultados dá-se o nome de Diagrama de Agulhas (DA).

Idealmente o FO deveria corresponder ao Campo de Movimento, o que nem sempre ocorre. Isto pode ser explicado pelo exemplo a seguir. Considere uma esfera perfeitamente uniforme girando sobre seu próprio eixo, diante de uma câmera de vídeo genérica, como mostrado na **Figura 2.2(a)**. É possível verificar que existe variação de brilho na imagem da esfera, desde que a mesma seja curva. No entanto,

esta variação de brilho, que também pode ser interpretada como sombreamento, não varia em função do movimento real da superfície, ou seja, o objeto gira, mas a imagem não muda. Isto implica que o fluxo ótico é sempre nulo, mesmo com o Campo de Movimento sendo variável. Agora considere esta mesma esfera em uma posição fixa e com uma fonte de iluminação móvel como indicado na **Figura 2.2 (b)**. A região sombreada da imagem irá mudar em função da variação de posição da fonte de luz, o que implica no Fluxo Ótico ser nitidamente diferente de zero. Apesar disto, sabemos que o Campo de Movimento da **Figura 2.2(b)** é sempre nulo.

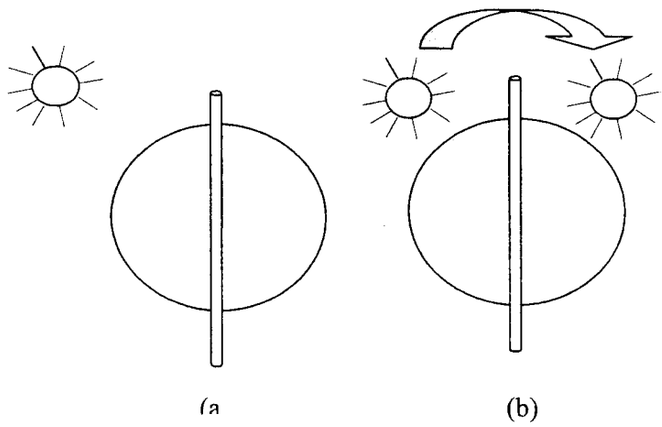


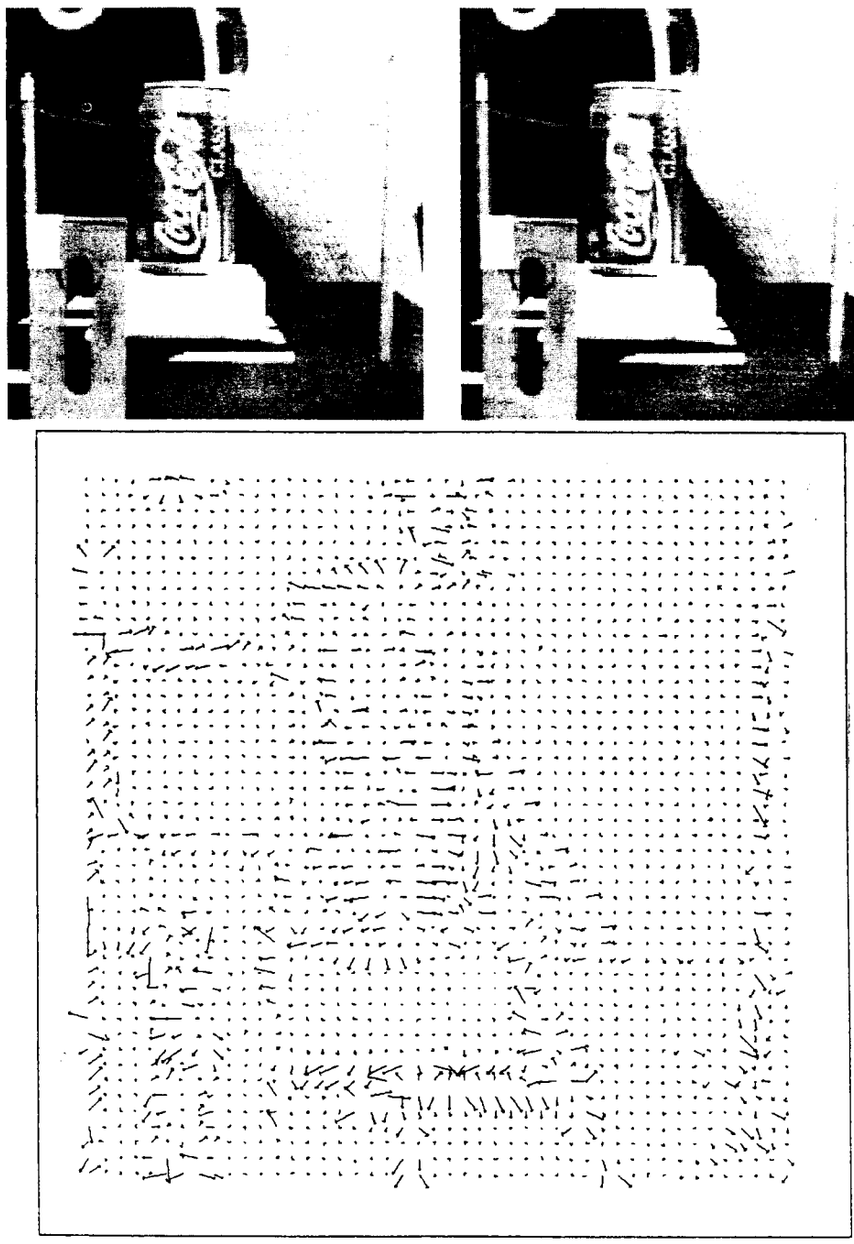
Figura 2.2. (a) esfera girando sob iluminação constante
(b) esfera fixa sob iluminação variável [HORN86]

Os exemplos citados anteriormente servem para esclarecer que nem sempre o FO e CM são coincidentes, e o que fica evidente é que as informações relevantes para a estimação de movimento são as mudanças ocorrentes na imagem.

Um exemplo de fluxo ótico e seu diagrama de agulhas é mostrado nas **Figuras 2.3 e 2.4.** [PATIÑO97].

USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

Já a figura 2.5 é uma aplicação de metodologia de detecção de movimento em monitoramento de multidões. A mesma mostra a comemoração do Ano Novo em Londres, onde um Sistema de Televisão em Circuito Fechado é usado para acompanhamento on-line de eventuais tumultos [BOGHOSSIAN99].



Figuras 2.3 – Na parte superior estão os dois quadros de uma cena contendo uma lata de bebida transladada. Na parte inferior é mostrado o DA correspondente ao movimento existente. [PATIÑO97]

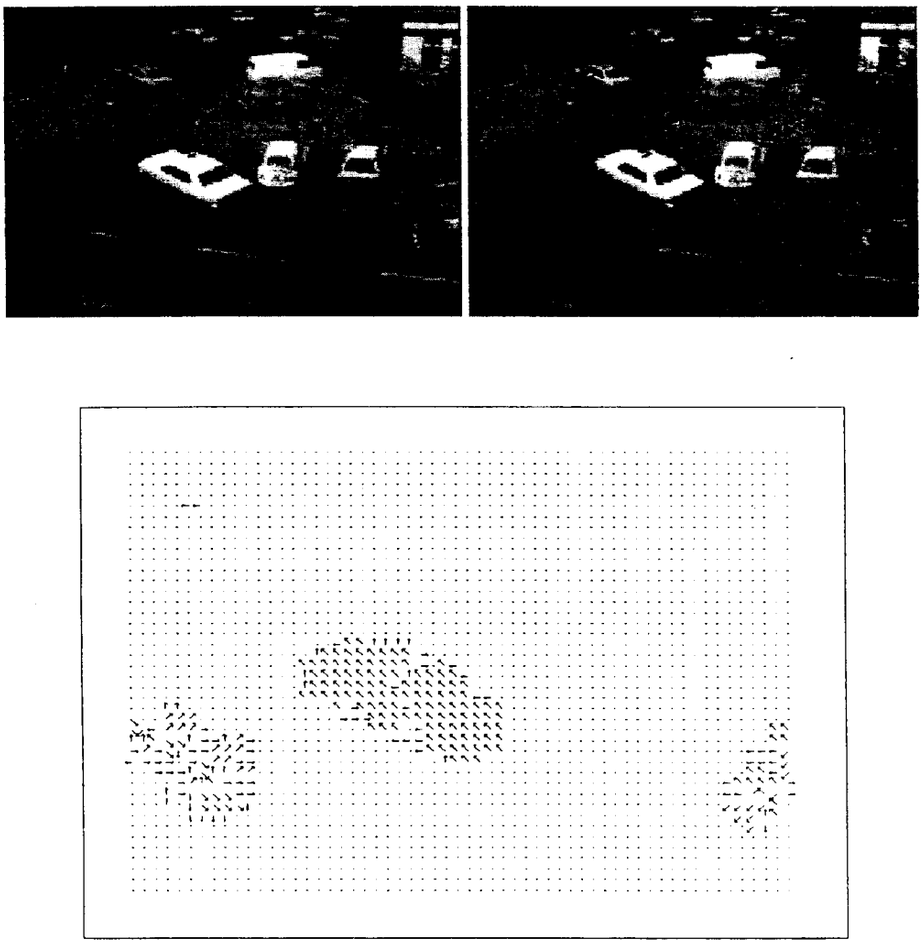


Figura 2.4 – Na parte superior estão os dois quadros de imagens de uma cena de tráfego. Na parte inferior é mostrado o DA correspondente ao movimento existente. [PATIÑO97]

Figura 2.5. Comemoração do Ano Novo em Londres, onde um Sistema de Televisão em Circuito Fechado para acompanhamento "on-line" de eventuais tumultos. Algoritmos de detecção de movimento auxiliam a otimização destas tarefas.

2.3 Problemas de Abertura e Oclusão

A estimação de movimento a partir de fluxo ótico apresenta problemas decorrentes do tipo de solução para a EFO. Estes problemas estão relacionados com a existência, unicidade e continuidade das soluções, e, portanto dependem da natureza do movimento a ser detectado. Quando faltam informações extras sobre o movimento em questão, então estamos diante do problema "ill-posed". Bertero em [BERTERO88] diz que a ocorrência de um problema de solução para estimações

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

através de fluxo ótico é denominada "ill-posed" quando não existe uma única solução e/ou a(s) solução(es) não dependem dos dados.

A oclusão é um tipo de problema "ill-posed" e ocorre quando não há correspondência entre *pixels* de dois quadros subseqüentes, que ocorrem, por exemplo, em cenas com rotação e/ou translação do objeto. A oclusão pode ser mais bem compreendida através da **Figura 2.6**:

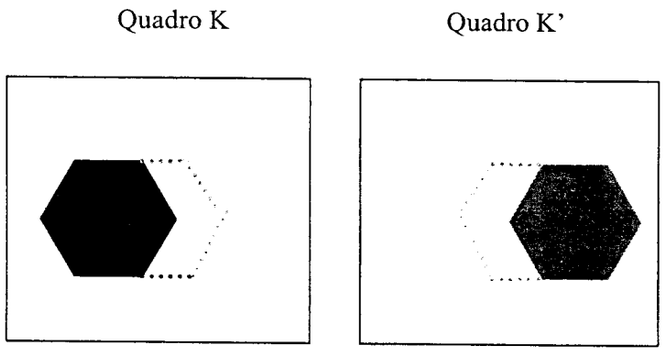


Figura 2.6. Problemas de oclusão.

O objeto com tracejado cheio translada-se na direção horizontal entre os intervalos de tempo *t* e *t'*, que correspondem respectivamente aos quadros *K* e *K'*. Na **Figura 2.6**, a região pontilhada no quadro K mostra o fundo a ser coberto no quadro *K+1*, enquanto a região pontilhada do quadro *K+1* corresponde à região do fundo descoberta pelo movimento do objeto. Em ambas as situações, os respectivos *pixel*s das linhas pontilhadas não têm correspondentes na outra imagem. Em outras palavras, quando ocorre oclusão não existe uma solução para a estimação do movimento.

Há vários estudos de estimação de movimento com o objetivo de minimizar os problemas de oclusão. [ZHANG02], dentre outros, desenvolve um algoritmo que

reduz as ambigüidades das formas detectadas durante a captura da imagem. Para tal, desenvolveu-se um algoritmo de segmentação de movimento baseado em análise do erro residual, que detecta com êxito as regiões com oclusão, e as considera como ruído. Este algoritmo foi testado em situações com presença de oclusões em seqüências sintéticas e reais, fornecendo resultados satisfatórios em capturas de imagens de cabeça.

Já o problema da abertura corresponde a situações em que a solução para a estimação 2-D de um movimento não é única. Se os vetores de movimento para cada *pixel* são variáveis independentes, então o número de incógnitas é duas vezes o número de equações. Ciente de que podemos calcular somente os componentes de velocidade paralelos ao gradiente espacial de imagem, analisemos a **Figura 2.7** em [TEKALP95].

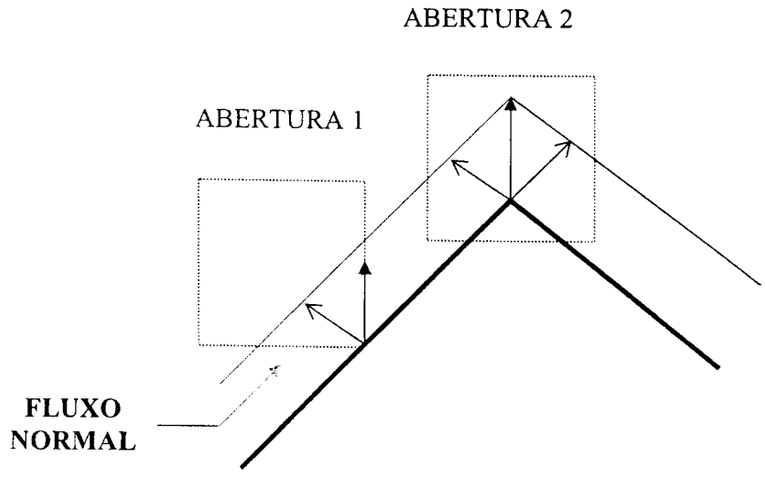


Figura 2.7. Problema da abertura num bloco com movimento ascendente, tirado do livro de Murat Tekalp [TEKALP95].

Considere um objeto com movimento ascendente e duas janelas distintas de estimação de movimento, respectivamente denominadas por ABERTURA 1 e ABERTURA 2. Observe que na ABERTURA 1, não é possível determinar a direção do movimento, ou seja há multiplicidade de soluções. Na ABERTURA 2 é possível estimar o movimento correto. Ou seja, pode-se determinar somente os componentes de velocidade paralelos ao gradiente de imagem espacial pela computação local.

2.4 Métodos baseados na Equação do Fluxo ótico

Nesta seção são apresentados alguns métodos de obtenção de Fluxo ótico. Devido à natureza "ill-posed" intrínseca ao Fluxo ótico, os algoritmos que implementam esta metodologia de detecção de movimento precisam ser modelados de acordo com a estrutura do Campo de Movimento. As técnicas de FO podem ser divididas em [CAMUS], como as baseadas em:

- Gradiente;
- Correspondência;
- Filtros sintonizados de velocidade.

2.4.1 Métodos baseados em gradiente

São métodos que implementam a solução da EFO, através de derivadas e por isso, são técnicas sensíveis a ruídos, que geralmente requerem uma suavização prévia das imagens. Neste processo, corre-se o risco de suavizar descontinuidades do

IFSC-USP SERVIÇO DE BIBLIOTECA E INFORMAÇÃO

Campo de Movimento, introduzindo medidas imprecisas, que aumentam o custo computacional.

2.4.2 Métodos baseados em correlação

Devido aos problemas de abertura, descritos anteriormente, a determinação do FO torna-se dificultada. Afora situações ideais, nas quais o movimento é unicamente fronto-paralelo, sem presença de ruído, com iluminação ambiente constante e translação de um número inteiro de *pixels*, a única forma de tornar o FO calculável é fazer considerações de tal modo que o problema "ill-posed" torne-se "well-posed", ou seja, com solução existente e única.

Buelthoff em [BUELTHOFF89] descreve um método de detecção de movimento baseado na técnica de Fluxo Ótico por correlação. Chamgming Sun em [SUN02] apresenta um algoritmo rápido e confiável que estima o movimento de uma seqüência de imagens, calculando o FO também através de correlação cruzada. Este processa imagens de 256 x 256 em um computador de 85MHz com um tempo de processamento da ordem de poucos segundos.

Para entender a metodologia citada, observe a **Figura 2.8**. Assume-se que o máximo deslocamento disponível para qualquer *pixel* na imagem, com direção arbitrária, seja η. Lembrando que em geral avaliamos movimento de objetos rígidos, portanto, a velocidade de um dado *pixel* é a mesma velocidade daqueles *pixels* que formam sua vizinhança. A vizinhança neste trabalho considera uma janela de tamanho v centrada num *pixel* qualquer. No exemplo da **Figura 2.8** estes parâmetros são $\eta = 2$ e $v = 5$.

O movimento de um dado *pixel* posicionado em (x, y) é determinado como sendo o caminho percorrido pela janela de v *X* v *pixels* centrado em (x, y) através dos $(2\eta+1)*(2\eta+1)$ possíveis deslocamentos. O movimento correto é detectado pela simulação da varredura da janela através dos possíveis deslocamentos, computando a medida do respectivo casamento numa matriz.

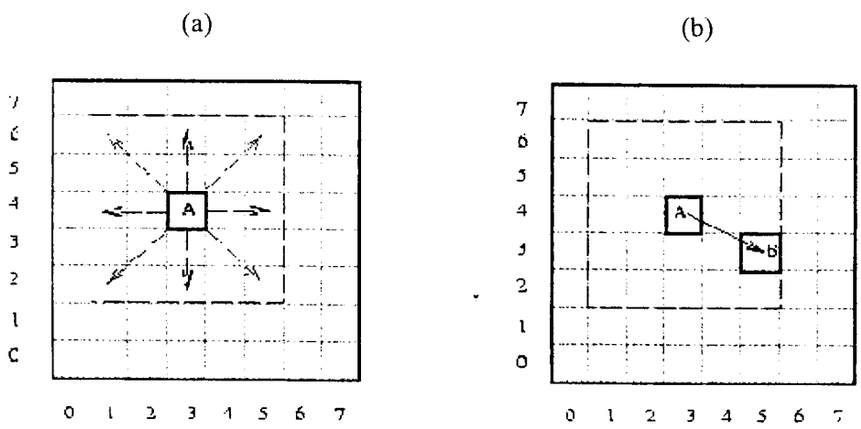


Figura 2.8. (a) Busca do pixel A com η igual a 2

(b) Movimento mostrado pelo vetor cuja origem é A e extremo é B

Esta matriz mostra as posições de um *pixel* nos quadros de uma seqüência para as quais o perfil pertencente à janela escolhida na primeira imagem deve ter se movido.

A **Figura 2.9** mostra o FO para *pixels* adjacentes a um dado *pixel*, como sendo constante, devido à consideração de que este seja um objeto rígido.

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

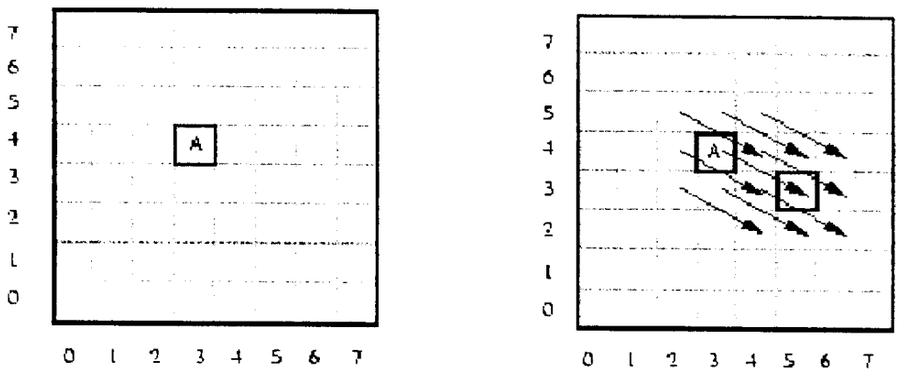


Figura 2.9 O número de *pixels* adjacentes de um dado *pixel* A, assumido como sendo constante devido ao fato de ser um corpo rígido (v igual a 3).

As técnicas baseadas em correlação tendem a ser bastante robustas, principalmente em imagens onde ocorre translação. Além desta característica há o fato da implementação maciçamente paralela fazer com que a correlação não satisfaça o requisito de velocidade dos sistemas de visão robótica. Os algoritmos tornam-se "close-to-real-time". Algoritmos em tempo real devem ser aqueles que são rápidos o suficiente para serem usados por um sistema de realimentação ou reação.

2.4.3 Métodos baseados em filtros sintonizados de velocidade.

Um dos métodos de estimação de Fluxo ótico avalia o movimento em termos das freqüências temporal e espacial. Dessa forma, um padrão bidimensional em movimento no plano-imagem corresponde a um movimento noutro plano definido no domínio das freqüências espaço-temporais w_x, w_y, w_t. Este plano é descrito por:

$$w_t = u\, w_x + v\, w_y \qquad \textbf{(2.5)}$$

Esta abordagem supõe que os vários pontos do movimento sejam encontrados no domínio das freqüências espaço-temporais formando um plano. Em situações com ocorrência de problemas de abertura, os pontos do domínio das freqüências formam uma reta, e o movimento não pode ser completamente detectado.

2.5 Revisão das Técnicas

2.5.1 Método de Horn e Schunck

A essência deste método é a suposição de que o Campo de Movimento satisfaz a EFO com variação constante de intensidade *pixel* a *pixel* ao longo de toda a imagem. Esta variação é descrita pela expressão:

$$\frac{dS_c(x_1, x_2, t)}{dt} = 0 \qquad \textbf{(2.6)}$$

Sendo que a **Equação 2.7** denota o erro na equação do FO.

$$\varepsilon_{FO}(\mathbf{v(x;t)}) = \left\langle \nabla S_c(\mathbf{x;t}), \mathbf{v(x;t)} \right\rangle + \frac{\partial S_c(x;t)}{\partial t} \qquad \textbf{(2.7)}$$

Quando a **Equação 2.6** é satisfeita, o erro é igual a zero. Na ausência de situações de oclusão e de ruídos, podemos minimizar o quadrado do erro e assim formar um

sistema de duas equações, visto que a EFO não é suficiente para a determinação do vetor velocidade.

A variação *pixel* a *pixel* dos vetores de velocidade pode ser quantificada pela soma dos quadrados das magnitudes dos gradientes espaciais das componentes da velocidade, como mostra a **Equação 2.8**.

$$\varepsilon^2{}_{FO}(\mathbf{v}(\mathbf{x};\mathbf{t})) = \left\|\nabla \mathbf{v}_1(x,t)\right\|^2 + \left\|\nabla \mathbf{v}_2(x,t)\right\|^2 \qquad \textbf{(2.8)}$$

Esta equação descreve a soma dos Laplacianos, e pode ser rescrita por:

$$\varepsilon^2{}_{FO}(\mathbf{v}(\mathrm{x};\mathrm{t})) = \left(\frac{\partial \mathrm{v}_1(\mathrm{x};\mathrm{t})}{\partial \mathrm{x}_1}\right)^2 + \left(\frac{\partial \mathrm{v}_1(\mathrm{x};\mathrm{t})}{\partial \mathrm{x}_2}\right)^2 + \left(\frac{\partial \mathrm{v}_2(\mathrm{x};\mathrm{t})}{\partial \mathrm{x}_1}\right)^2 + \left(\frac{\partial \mathrm{v}_2(\mathrm{x};\mathrm{t})}{\partial \mathrm{x}_2}\right)^2 \qquad \textbf{(2.9)}$$

Nestas expressões as coordenadas espaciais e temporais são consideradas variáveis contínuas. A **Equação 2.9,** quando minimizada, suaviza o vetor velocidade e, portanto, é chamada de equação de suavização.

As **Equações 2.10 e 2.11**, descrevem a proposta de Horn e Schunck que é uma solução iterativa para determinar os valores das componentes de **v**, partindo de um ponto inicial suposto.

$$\hat{\mathrm{v}}_1{}^{(n+1)}(\mathrm{x};\mathrm{t}) = \bar{v}_1{}^{(n)}(\mathrm{x};\mathrm{t}) - \frac{\partial \mathrm{S_c}(\mathrm{x};\mathrm{t})}{\partial \mathrm{x}_1} \frac{\frac{\partial \mathrm{S_c}(\mathrm{x};\mathrm{t})}{\partial \mathrm{x}_1}\bar{v}_1{}^{(n)}(x,t) + \frac{\partial \mathrm{S_c}(\mathrm{x};\mathrm{t})}{\partial \mathrm{x}_2}\bar{v}_2{}^{(n)}(x,t) + \frac{\partial \mathrm{S_c}(\mathrm{x};\mathrm{t})}{\partial \mathrm{t}}}{\alpha^2 + \left(\frac{\partial \mathrm{S_c}(\mathrm{x};\mathrm{t})}{\partial \mathrm{x}_1}\right)^2 + \left(\frac{\partial \mathrm{S_c}(\mathrm{x};\mathrm{t})}{\partial \mathrm{x}_2}\right)^2} \qquad \textbf{(2.10)}$$

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

$$\hat{v}_2^{(n+1)}(x;t) = \bar{v}_2^{(n)}(x;t) - \frac{\partial S_c(x;t)}{\partial x_2} \frac{\frac{\partial S_c(x;t)}{\partial x_1} \bar{v}_1^{(n)}(x,t) + \frac{\partial S_c(x;t)}{\partial x_2} \bar{v}_2^{(n)}(x,t) + \frac{\partial S_c(x;t)}{\partial t}}{\alpha^2 + \left(\frac{\partial S_c(x;t)}{\partial x_1}\right)^2 + \left(\frac{\partial S_c(x;t)}{\partial x_2}\right)^2} \qquad \textbf{(2.11)}$$

O parâmetro α é selecionado heuristicamente e controla a restrição de suavização da imagem, a partir da escolha de α a qual determina se as componentes de *v* convergem mais ou menos. O parâmetros *n+1* e *n* são respectivamente as iterações calculadas, atual e anterior, respectivamente. Os valores iniciais de v em geral são nulos e os gradientes são todos estimados usando diferenças finitas.

2.5.2 Métodos Diferenciais de Segunda Ordem

Na busca de determinar o Fluxo ótico, outros pesquisadores como [NAGEL87] e [URAS88] sugeriram a substituição do uso da equação do fluxo ótico descrita em **(2.1)**, que é a conservação do gradiente espacial da distribuição de intensidade do *pixel* ao longo da trajetória, pela **Equação 2.12** mostrada a seguir:

$$\begin{bmatrix} \hat{v}_1(x;t) \\ \hat{v}_2(x;t) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 S_c(\mathbf{x};t)}{\partial x_1^2} & \frac{\partial^2 S_c(\mathbf{x};t)}{\partial x_1 x_2} \\ \frac{\partial^2 S_c(\mathbf{x};t)}{\partial x_1 x_2} & \frac{\partial^2 S_c(\mathbf{x};t)}{\partial x_2^2} \end{bmatrix}^{-1} \begin{bmatrix} -\frac{\partial^2 S_c(\mathbf{x};t)}{\partial t \partial x_1} \\ -\frac{\partial^2 S_c(\mathbf{x};t)}{\partial t \partial x_2} \end{bmatrix} \qquad \textbf{(2.12)}$$

Onde *–1* determina a inversa da matriz. A solução desta equação apresenta restrições em relação a alguns tipos de movimento como rotação e escalonamento por aproximação ("zooming").

2.5.3 Métodos de Movimento de Blocos

Uma forma de solucionar o problema da abertura citado anteriormente é considerar o vetor de movimento constante ao longo de um determinado bloco de *pixels*. Lucas e Kanade em [LUCAS81] utilizaram tal consideração, e concluíram que embora este também seja um método pouco confiável em situações de movimento rotacional, ele fornece a possibilidade de cálculo do vetor de velocidade unicamente devido ao movimento translacional de uma determinada seqüência. Não chegamos a utilizar esta técnica em nossa implementação.

2.6 Conclusão

Observa-se que existem vários métodos implementados de fluxo ótico para detecção de movimento em seqüências de imagens disponíveis na bibliografia, cada um apresentando habilidades próprias, sejam eles mais rápidos, ou melhor aplicáveis a situações de movimento translacional, outros a rotacional, ou ainda sendo mais ou menos sensíveis a ruídos na iluminação, etc. Existe obviamente um compromisso entre as habilidades do método escolhido e a aplicação a que o mesmo se destina. Um exemplo é o método de Sarcinelli Filho em [FILHO01], onde o autor busca um método apropriado para navegação robótica, implementando alguns dos descritos na literatura. As experiências de navegação são feitas em um robô móvel que contém uma câmera CCD. O fluxo ótico esperado é nulo em suas componentes horizontais, devido à natureza do experimento. A análise dos resultados é feita em função dos

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

parâmetros de confiabilidade do cálculo do Fluxo ótico (valor médio da magnitude dos vetores de FO e desvio padrão associado, ângulo dos vetores de FO e erro médio associado), do custo computacional e do tempo de colisão.

3 Revisão sobre Métodos de Detecção de Bordas

3.1 Introdução

As técnicas de detecção de bordas consistem na segmentação de imagens, baseadas na operação de convolução com máscaras (filtros) pré-definidas, com o objetivo de identificar as fronteiras entre regiões com níveis de cinza distintos. A estas fronteiras dá-se o nome de bordas. Analisaremos neste capítulo apenas as bordas de luminosidade. Gonzalez em [GONZALEZ92] ilustra o conceito de detecção com a figura abaixo:

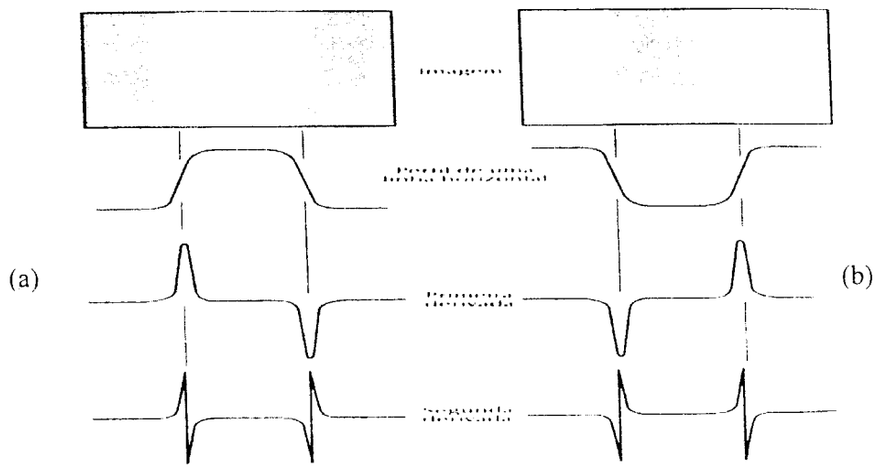


Figura 3.1 Perfis de nível de cinza e suas respectivas derivadas de primeira e Segunda ordem

A **Figura 3.1 (a)** mostra uma imagem com uma faixa clara sobre um fundo escuro, o perfil de nível de cinza ao longo de uma linha de varredura horizontal, e suas respectivas derivadas de primeira e segunda ordem. A mudança do nível de cinza escuro para o nível de cinza claro é modelada por uma transição entre os níveis de cinza, suavizada pela etapa de amostragem das imagens digitalizadas.

Observe que a primeira derivada é nula nas áreas de nível de cinza constante e diferente de zero nas áreas de mudança de nível e, portanto, sua magnitude é útil para detectar a existência de uma borda na imagem. A segunda derivada muda de sinal em função do nível de cinza, e é nula nas áreas de nível constante, tendo um cruzamento com zero nestas regiões. Este sinal é usado para determinar se um pixel da borda localiza-se na área escura ou clara da imagem e seus pontos de cruzamento com zero são de fundamental importância para localização das bordas. Em uma imagem estas derivadas de primeira e segunda ordem são obtidas respectivamente pela magnitude do gradiente e pelo Laplaciano da imagem. A **Figura 3.1(b)** tem comportamento analogamente inverso.

Nas últimas décadas foram desenvolvidos muitos detectores de borda, dentre os quais os de maior interesse neste trabalho são os detectores de Roberts, de Sobel, de Prewitt e Laplaciano.

3.2 Operadores Gradiente

Estes operadores são descritos pela expressão:

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

$$\nabla \mathbf{f} = \begin{bmatrix} Gx \\ Gy \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathrm{f}}{\partial \mathrm{x}} \\ \frac{\partial \mathrm{f}}{\partial \mathrm{y}} \end{bmatrix} \tag{3.1}$$

onde ∇f é o gradiente da imagem f (x,y) na posição (x,y), e Gx e Gy são as máscaras que mudam de acordo com o tipo de detecção.

Em detecção de bordas a magnitude e a direção do vetor gradiente são de grande importância, fornecendo informação sobre a taxa de aumento de f (x, y) na direção de ∇f e a direção do mesmo. As equações da magnitude e da direção do vetor gradiente são dadas pela expressão:

$$mag(\nabla \mathbf{f}) = \left[Gx^2 + Gy^2\right]^{1/2} \tag{3.2}$$

$$\alpha\ (x\ ,\ y\ ) = \tan^{-1}\left[\frac{Gx}{Gy}\right] \tag{3.3}$$

A **Figura 3.2** mostra as máscaras de alguns métodos de detecção de bordas disponíveis na literatura ([GONZALEZ92], etc).

Pelas máscaras pode-se observar que o Sobel é o operador que acentua os *pixels* próximos ao centro da máscara. A implementação das derivadas destes operadores pode ser elaborada das mais diferentes formas, e, portanto apresentar desempenhos diferentes, podendo inclusive acentuar os ruídos. Em relação ao ruído o operador de Sobel torna-se mais atrativo, pois suaviza a imagem.

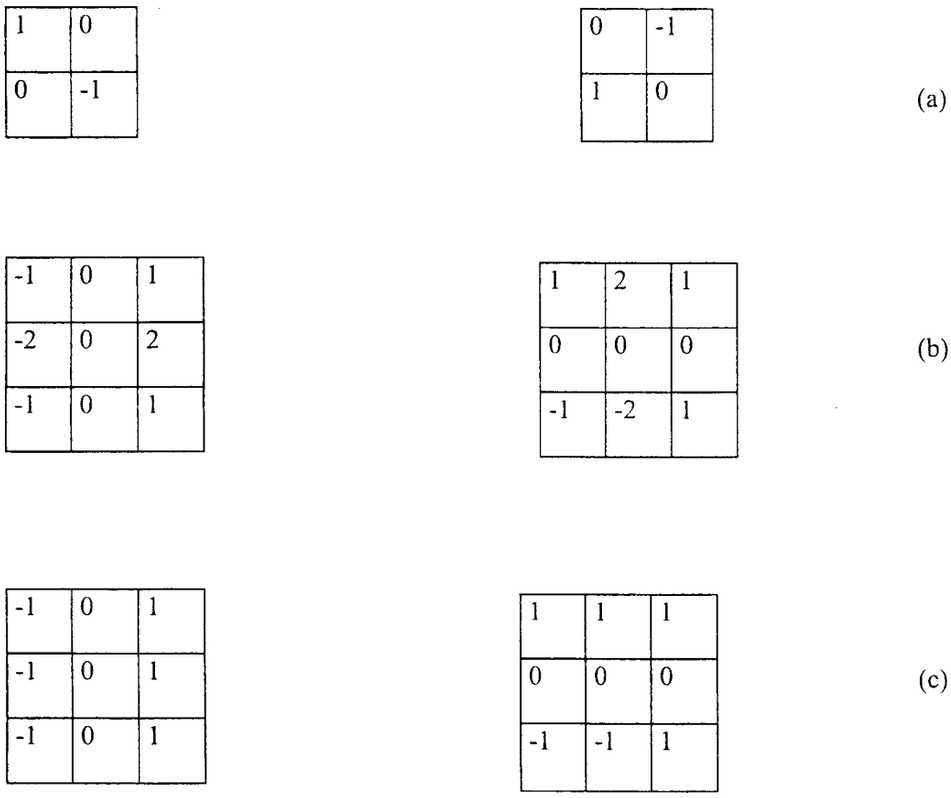


Figura 3.1. Máscaras dos operadores gradiente (a) Roberts; (b) Sobel; (c) Prewitt.

Veja também a **Figura 3.3**, que é um exemplo de aplicação de detecção de bordas de Prewitt e de Sobel, com diferenças pouco perceptíveis.

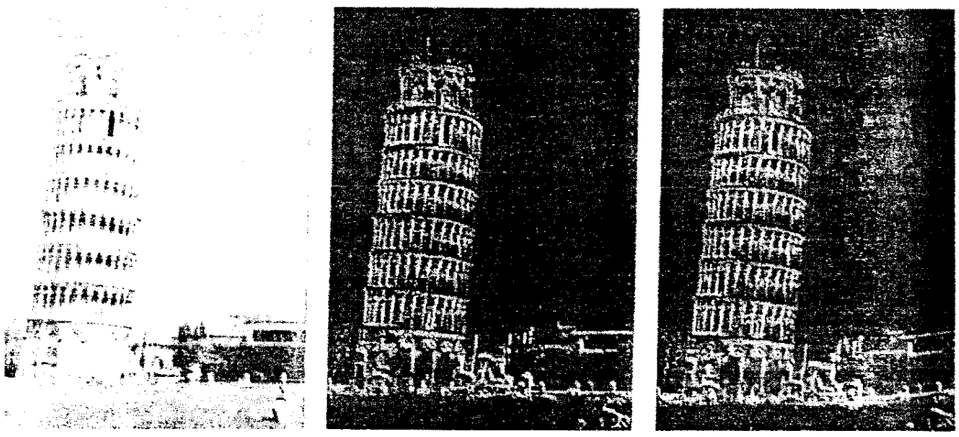


Figura 3.3. (a) imagem original, (b) realce de bordas utilizando os operadores de Prewitt vertical e horizontal, (c) realce de bordas utilizando os operadores de Sobel vertical e horizontal [FILHO99].

3.3 Operador Laplaciano.

O operador Laplaciano baseia-se na derivada de segunda ordem, e, portanto, detecta as bordas, quando a função resultante da aplicação deste operador na imagem cruza pelo zero. É definido por:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad (3.4)$$

Na forma digital pode ser implementado de várias formas, contanto que o coeficiente associado ao *pixel* central seja positivo, e os *pixels* externos sejam negativos, como mostram os exemplos de máscaras presentes na a seguir.

0	-1	0
-1	4	-1
0	-1	0

(a)

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	24	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

(b)

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	8	8	8	-1	-1	-1
-1	-1	-1	8	8	8	-1	-1	-1
-1	-1	-1	8	8	8	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

(c)

Figura 3.2. Máscaras usadas no cálculo do Laplaciano, a primeira de tamanho 3x 3, a segunda 5 x 5 e a terceira 9 x 9.

A **Figura 3.4** a seguir mostra os resultados obtidos com a aplicação de cada máscara aplicada a uma única imagem monocromática.

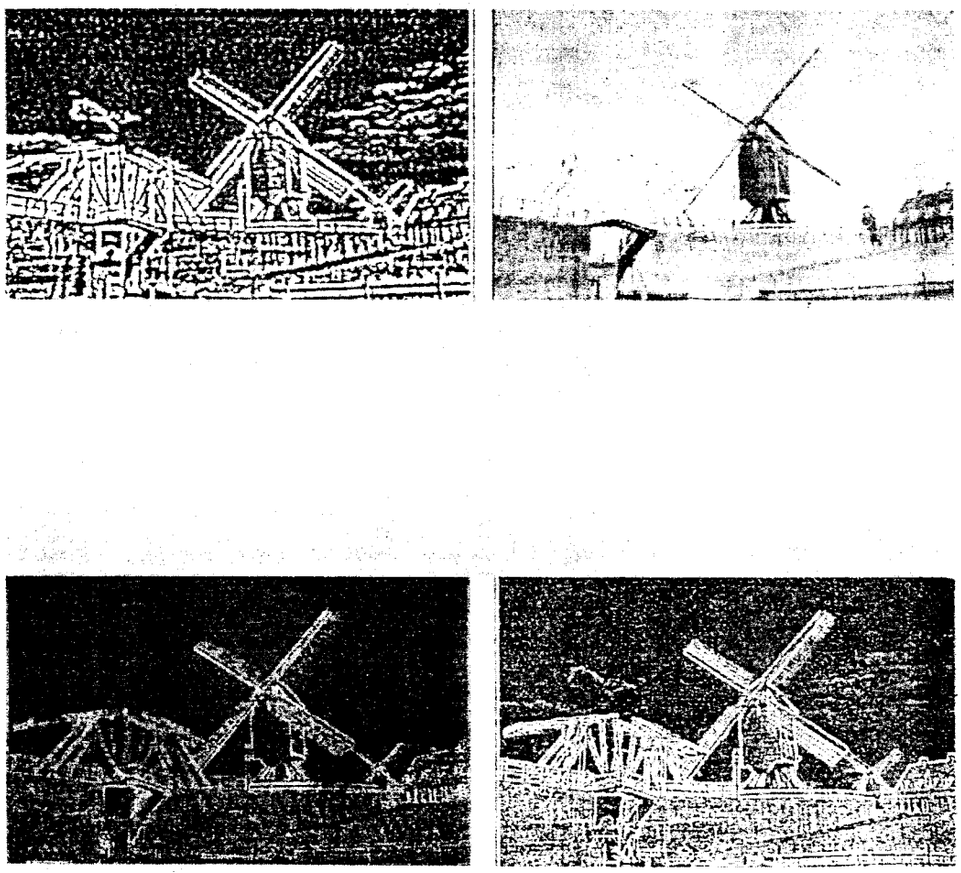
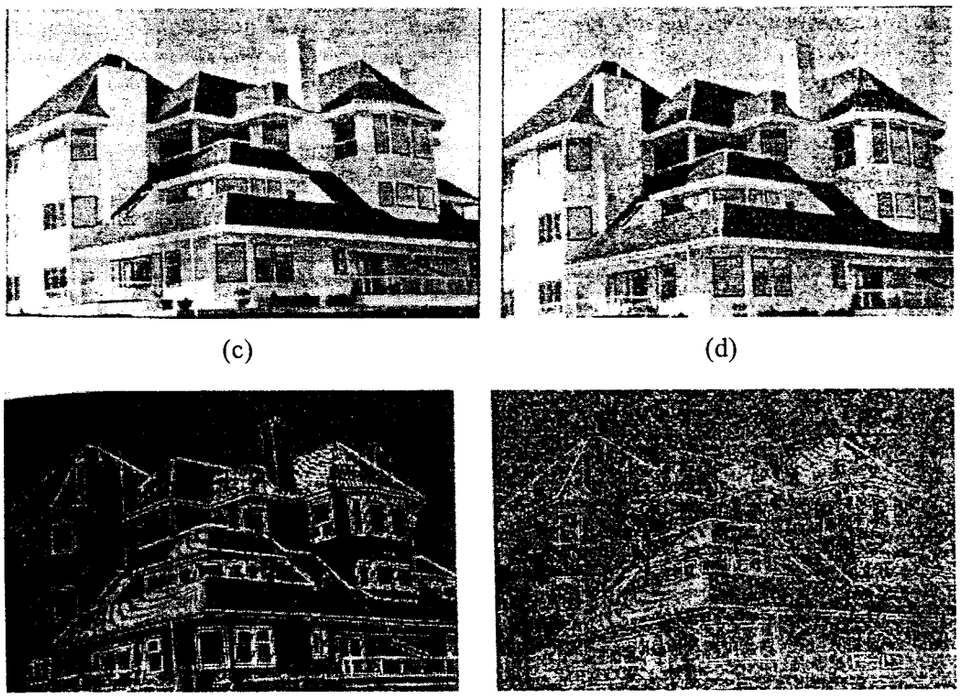


Figura 3.3. (b)Imagem original; (a), (c), (d) Imagens resultantes de laplacianos 9x9, 3x3, 5x5 [FILHO99].

Embora este operador seja sensível à transição de intensidade, ele tem suas limitações práticas, que o tornam um detector secundário que estabelece se um *pixel*

está do lado escuro ou claro da imagem. Estas limitações estão relacionadas aos fatos de sendo uma derivada de Segunda ordem, sua sensibilidade a ruído ser aumentada e por produzir bordas duplas, que impossibilitam de determinar a direção exata da borda. Os exemplos da **Figura 3.5** mostram o Laplaciano aplicado a uma imagem sem ruído, e a esta mesma imagem acrescida de ruído.

(a) {b}



(c) (d)

Figura 3.5. (a) Imagem original (b) Imagem acrescida de ruído

(c) Laplaciano de (a) (d) Laplaciano de (b)

3.4 Detectores de bordas para determinação de movimento

O ponto básico deste projeto é o aumento de velocidade dos algoritmos de detecção de movimento baseados em técnicas precisas e computacionalmente pesadas, através da redução da área útil de trabalho da imagem.

Optou-se por fazer a redução usando a metodologia de detecção das bordas dos objetos em movimento na imagem. Cabestaing em [CABESTAING91] descreve o processador STREAM, cuja idéia é ilustrada na **Figura 3.6**.

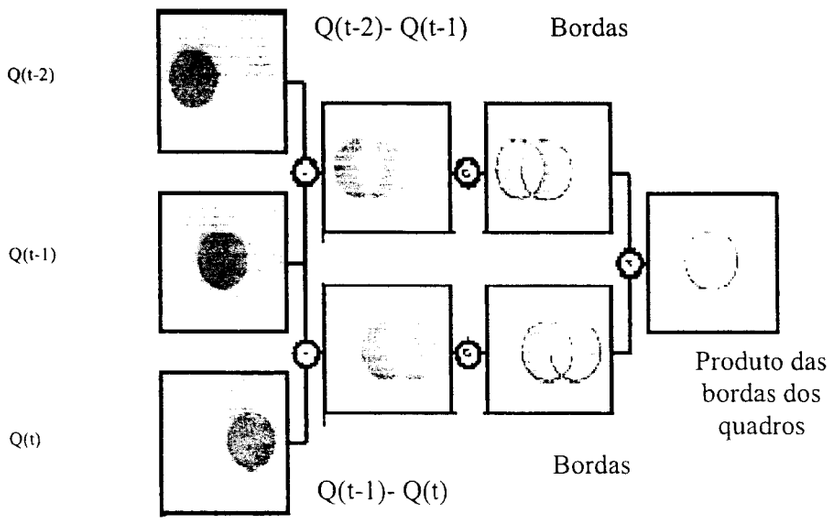


Figura 3.6. Detector de Cabestaing.

Este detector de movimento é aplicado em conjunto de três quadros - Q(t-2), Q(t-1) e Q(t) - de uma dada seqüência, fazendo inicialmente a subtração dos pares de quadros (segundo e primeiro; terceiro e segundo), obtendo dois outros resultados. Efetuando esta subtração, aplica-se um detector de bordas dentre os conhecidos na literatura (Sobel, Prewitt, Roberts) em cada um dos dois resultados, e a seguir multiplicam-se os mesmos. A imagem resultante da aplicação do método de Cabestaing numa seqüência de três quadros corresponde a área da imagem com maior probabilidade de movimento de um ou mais objetos. A partir de conhecimento desta técnica, optamos por aplicá-la em nossas seqüências e seu resultado corresponderá à área útil da imagem na qual calcularemos o Fluxo ótico, ação que implica em redução de tempo de processamento ou ganho de velocidade.

3.5 Conclusões

Entre os métodos de pré-processamento utilizados na busca de uma redução de área útil, eficiente e adequada à nossa proposta, o método de Cabestaing, foi o que melhor apresentou resultados, descartando assim outras associações aos métodos de fluxo ótico existentes. Definindo-se então tal escolha, partimos para a implementação do mesmo em conjunto com métodos de cálculo de Fluxo ótico, e suas respectivas análises em função de parâmetros de tempo de execução e de Erro Quadrático Médio dos vetores dos diagramas de agulha obtidos pela nova implementação e respectiva implementação de fluxo ótico existente na bibliografia.

4 Redução do Tempo de Detecção de Movimento

4.1 Introdução

Como citado anteriormente, o intuito fundamental deste trabalho é aumentar a velocidade de algoritmos de detecção de movimento, através de utilização de métodos de pré-processamento que reduzam a área útil de cálculos de Fluxo ótico, sem alterar a precisão dos resultados.

4.2 Implementação do Algoritmo

Implementou-se inicialmente o algoritmo de determinação das regiões correspondentes ao movimento em seqüência de três quadros consecutivos, baseada no trabalho de [CABESTAING91]. A imagem resultante da aplicação do algoritmo indica a região onde o movimento é obtido. Numa etapa posterior esta máscara é utilizada para selecionar a região dos quadros onde dever-se-á calcular o fluxo ótico, ou seja, fazer a redução de área útil para cálculo do FO, e conseqüentemente reduzir

o tempo de processamento. Além disto, pode-se optar pelo cálculo do fluxo ótico direto, que em nosso caso, utiliza a metodologia de Horn e Schunck [HORN81].

Tendo-se calculado o Fluxo ótico com a máscara de Cabestaing e o Fluxo ótico sem esta máscara, calculamos o Erro Quadrático Médio (Apêndice A) resultante dos dois métodos. Este Erro Quadrático Médio mede o desvio da técnica final sugerida por esta tese, em relação a uma das técnicas usadas na literatura, quantificando assim a possível perda de informação do movimento decorrente de nosso método. Esta técnica é bastante popular devido a se relacionar razoavelmente com testes subjetivos de qualidade visual e por ser matematicamente praticável [PRATT91]. J. Kim em [KIM01] utiliza o Erro Quadrático Médio para comparar a técnica de estimação de FO estudada com outras existentes na literatura, verificando assim seu satisfatório desempenho.

Também se calcula a relação entre os respectivos tempos de processamento das duas técnicas. Dessa forma avalia-se o aumento de velocidade conseguido na detecção do movimento devido à redução de área útil.

Destas duas formas, através do EQM e da razão entre os tempos de processamento, avaliamos a eficiência da técnica aqui desenvolvida.

Então considere o algoritmo como sendo:

1) Seleção de três quadros pertencentes a uma seqüência;
2) Cálculo do FO em toda a imagem dos três quadros;
3) Obtenção da região de movimento na imagem através do método de Cabestaing;
4) Redução da área útil da imagem para cálculo do FO, usando o passo 3;
5) Cálculo do FO na área reduzida da imagem;

6) Cálculo do EQM entre os resultados de FO obtidos no passo 2 e no passo 5;
7) Cálculo da razão entre os tempos de processamento de FO nos passos 2 e 5.

O programa desenvolvido para implementação do algoritmo tem sua listagem apresentada no Apêndice C.

Há algumas opções introduzidas no menu do programa como a de cálculo da *sobreposição*, que é um auxílio na avaliação qualitativa do resultado, permitindo sobrepor o diagrama de agulhas referente ao movimento de uma seqüência de entrada - numa cor de destaque selecionada pelo usuário - sobre o primeiro quadro de uma dada seqüência.

Há também outras opções que precedem os cálculos de Fluxo ótico como configuração das relações das medidas da agulha e dos parâmetros de solução da Equação do Fluxo ótico. Estes últimos parâmetros são:

- Precisão: parâmetro relacionado à convergência na estimação das componentes do vetor velocidade de Horn e Schunck.
- FEV: fator de expansão dos vetores de Fluxo ótico, para melhor visualização nas situações em que os mesmos ficam sobrepostos, quando salvos em formato *BMP*.
- Alfa: este parâmetro é selecionado heuristicamente e controla a suavização das descontinuidades dos quadros da seqüência. Sua

escolha determina se as componentes de v convergem mais ou menos.

- Magnificação: fator de ampliação do tamanho da imagem resultante, também quando salvos em formato *BMP*.

O fluxograma deste trabalho é mostrado na **Figura 4.1**., o qual foi implementado em Obj*ect Pascal*, e o compilador foi o Delphi 5.0.

Inicialmente lêem-se três quadros de uma seqüência de imagens. Depois se opta por calcular o Fluxo ótico com ou sem redução de área através do método de Cabestaing [CABESTAING91]. Tendo calculado ambos os fluxos óticos, com e sem redução de área para uma mesma seqüência, calculam-se o respectivo Erro Quadrático Médio e a razão entre os tempos de processamento, como foi explicado no algoritmo.

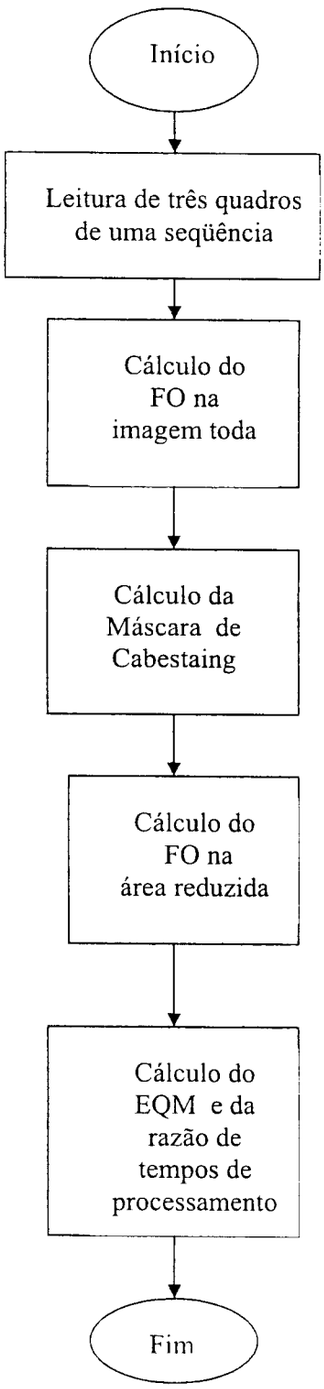


Figura 4.1 Fluxograma do método proposto.

A estrutura visual do programa é mostrada na ***Figura 4.2*** e ***Figura 4.3.***

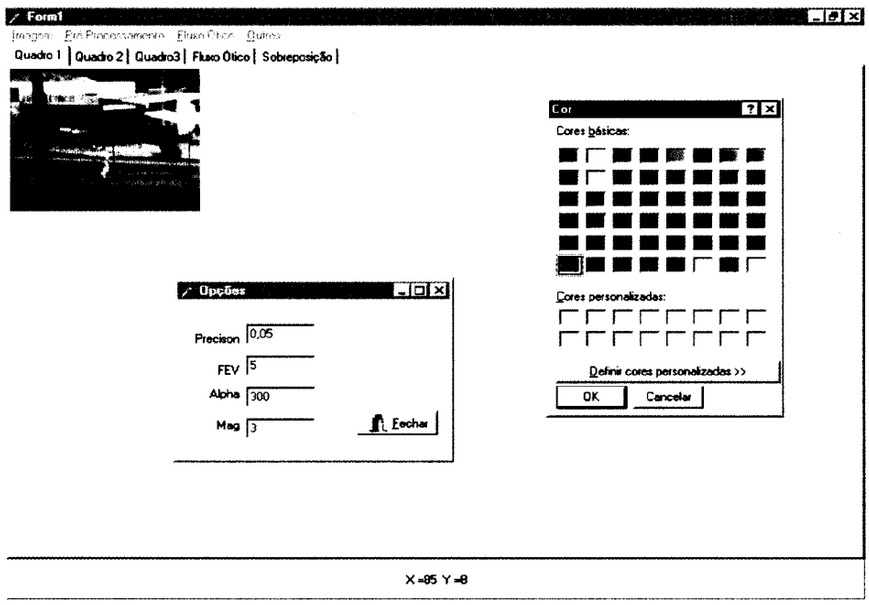


Figura 4.2. Implementação do algoritmo com seleção de parâmetros para cálculo do FO e seleção de cor do Diagrama de Agulhas.

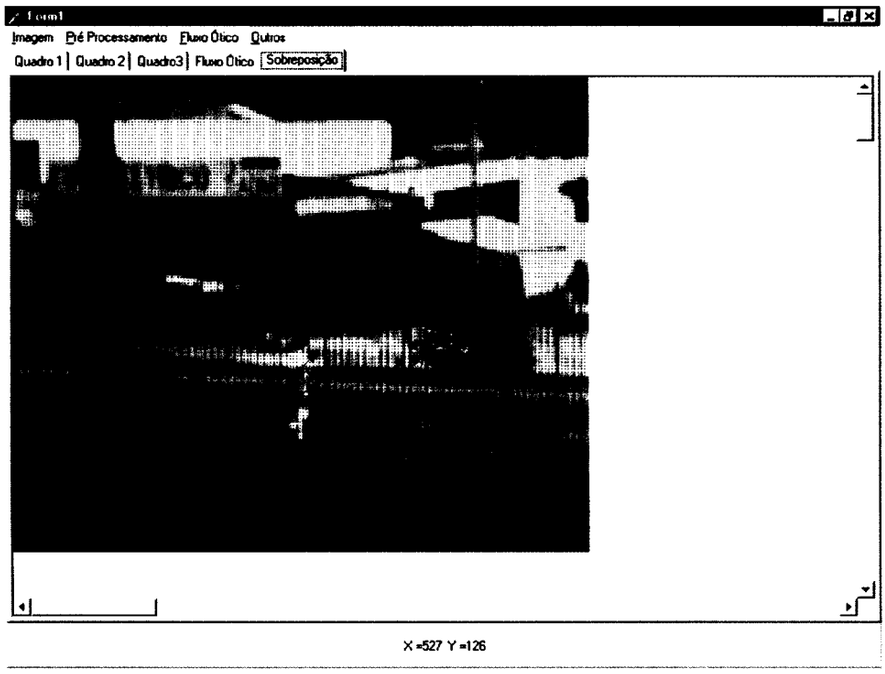


Figura 4.3. Implementação do algoritmo com sobreposição do Diagrama de Agulhas no primeiro quadro da imagem.

Também se implementou um algoritmo de visualização dos Diagramas de Agulha em *POSTSCRIPT*, podendo-se optar pelas saídas em formato *ps*, bastando para isso, usar o arquivo de dados gerado no algoritmo original.

4.3 Conclusões

Neste capítulo foi apresentada a descrição do algoritmo. A análise estatística do erro como fator de comparação quantitativa entre os métodos, torna o algoritmo aplicável, aumentando o desempenho em termos de tempo de processamento, e preservando o fator de precisão da detecção de movimento.

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

5 Resultados

5.1 Introdução

Este capítulo apresenta o método de validação do algoritmo desenvolvido na tese, assim como os resultados experimentais em vários tipos de seqüências e compara-os aos resultados obtidos na utilização do algoritmo disponível na literatura. Desta forma pode-se avaliar as vantagens e limitações do método aqui proposto, sua contribuição e importância no contexto de detecção de movimento.

5.2 Validação do Algoritmo

A metodologia de validação deste algoritmo de redução de tempo de processamento baseou-se na detecção do movimento cujo comportamento era previamente conhecido, ou seja, com suas variáveis de entrada controladas (movimento dos objetos da cena totalmente controlado). A semelhança dos diagramas de agulhas de cada caso, quando analisados visualmente é bastante perceptível. Esta semelhança é quantificada através do cálculo do Erro Quadrático Médio entre os respectivos diagramas de agulha, apresentado em [OLIVEIRA02]. Este erro mede o desvio de nosso método em relação à metodologia de cálculo de Fluxo ótico convencional. Em outras palavras, calculamos o desvio de cada vetor de

velocidade correspondente a cada *pixel* do diagrama obtido com nossa proposta em relação ao vetor do *pixel* de mesma posição do diagrama original. A detecção de movimento na imagem resumiu-se à sua aplicação a princípio em imagens sintéticas e posteriormente em imagens reais.

Nos resultados das Seqüências 1 a 48 do **Apêndice B**, cada uma é formada por 6 imagens que são: três imagens dos quadros de entrada, uma imagem da aplicação da máscara de Cabestaing, uma imagem de Fluxo ótico sem uso daquela máscara e uma imagem de Fluxo ótico com uso da máscara. As seqüências de 1 a 10 são sintéticas com movimentos de rotação e/ou translação de objetos de formas bem definidas. As seqüências 11 a 19 são imagens de brinquedos com os mesmos movimentos, feitas com câmera CCD no Laboratório de Visão Computacional Rápida do Departamento de Engenharia Elétrica da Universidade de São Paulo. As seqüências 20 a 23 foram adquiridas no site http://www.research.microsoft.com/users/anandan/images.htm de P.Anandan. As seqüências de 24 a 26 são obtidas a partir do movimento de um bebê, e as de 27 a 32 são obtidas a partir do monitoramento do tráfego de uma rua ao lado do mesmo laboratório em que trabalhamos. Além destas há outras seqüências como as de 32 a 35 obtidas de dentro de um carro em movimento em direção à uma represa, e as de 35 a 48, de dentro de um barco navegando nesta mesma represa.

Os resultados foram obtidos utilizando um computador tipo IBM-PC com processador *Athlon* 1,2 *GHz* e 256*K* de memória *RAM*. Para imagens sintéticas obtidas a partir de movimento de formas geométricas (Seqüências de 1 a 10), o Erro Quadrático Médio das imagens variou na faixa entre 0,26808 e 0,71686 com desempenho na redução de tempo de processamento da ordem de 25 vezes para o pior caso e 175 vezes para o melhor caso – ver Tabela 5.1. Esta redução de tempo se

deve ao fato da cena ter um único objeto, apresentar condições de iluminação constante e nenhum ruído. Por outro lado, imagens usadas na etapa de validação de nossa proposta, constituídas por desenhos de formas geométricas e aquelas obtidas de objetos em movimento totalmente conhecido (imagens de P. Anandan, e os brinquedos de Sophia), são nas primeiras que o Erro Quadrático Médio está maximizado.

Seqüências	Fluxo ótico	Fluxo ótico com filtro (Cabestaing)		
	tempo (s)	tempo (s)	razão entre os tempos	eqm
Figura B.1	0,220	1,803E-03	122,019	0,31932
Figura B.2	0,160	1,805E-03	88,643	0,33321
Figura B.3	0,220	3,492E-03	63,001	0,71686
Figura B.4	0,150	1,336E-03	112,275	0,69183
Figura B.5	0,170	5,366E-03	31,681	0,59323
Figura B.6	0,170	6,160E-03	27,597	0,56128
Figura B.7	0,170	6,621E-03	25,676	0,33513
Figura B.8	0,220	6,405E-03	34,348	0,37289
Figura B.9	0,220	1,332E-03	165,165	0,28057
Figura B.10	0,220	1,254E-03	175,439	0,26808

Tabela 5.1. Simulação com imagens sintéticas, a partir de movimento de formas geométricas.

Ao aplicarmos o algoritmo desenvolvido, em imagens reais com objetos em movimentos de rotação e/ou de translação, verificamos que a redução do tempo

(Seqüências de 11 a 19 – os brinquedos de Sophia – tabela 5.2) variou de 8 a 22 vezes, com tempos inferiores a 1 segundo, valor este aceitável quando comparável àquele obtido nas imagens sintéticas. Este resultado é satisfatório quando inserido num contexto de um projeto de robótica. O erro quadrático médio para estes resultados foi minimizado, em comparação ao das imagens sintéticas, o que incentiva à continuidade desta pesquisa.

Quando aplicado aos brinquedos de Anandan (Seqüências de 20 a 23 – tabela 5.2) este erro aumentou. Isto ocorre devido ao fato das imagens estarem bastante ruidosas com relação à iluminação do ambiente. Em relação ao desempenho na redução do tempo, a variação foi de 4 a 16 vezes, o que também é satisfatório, mesmo sendo inferior aos resultados obtidos anteriormente.

Seqüências	Fluxo ótico	Fluxo ótico com filtro (Cabestaing)		
	tempo (s)	tempo (s)	razão entre os tempos	eqm
Figura B.11	6,040	0,080	75,500	9,3064E-11
Figura B.12	6,100	0,440	13,864	2,7580E-06
Figura B.13	6,090	0,440	13,841	1,6149E-06
Figura B.14	6,050	0,720	8,403	1,2706E-05
Figura B.15	6,320	0,550	11,491	9,6684E-07
Figura B.16	6,040	0,330	18,303	8,4780E-07
Figura B.17	6,040	0,270	22,370	0,00012
Figura B.18	6,100	0,330	18,485	0,00071
Figura B.19	6,090	0,380	16,026	0,00178
Figura B.20	6,090	1,260	4,833	0,00056
Figura B.21	6,050	0,850	7,118	0,00788
Figura B.22	2,900	1,090	2,661	0,02263
Figura B.23	6,040	1,370	4,409	0,08768

Tabela 5.2. Simulação com imagens reais com variáveis de entrada controladas.

5.3 Apresentação dos Resultados

Neste item apresentam-se os resultados utilizando imagens sintéticas e reais com movimento dos objetos controlado ou não. Todas as imagens de teste encontram-se no Apêndice B.

5.3.1 Imagens sintéticas

Como já citado, as imagens sintéticas foram feitas utilizando o software *PAINTBRUSH,* utilizando formas geométricas bem definidas, com movimentos de translação e/ou rotação. As imagens encontram-se apresentadas no Apêndice B, nas **Figuras B.1.1** a **B10.6**. Observou-se nestes 10 casos que o Erro Quadrático Médio variou de 0,26808 a 0,71686, sendo que o melhor caso foi obtido da aplicação com movimento translacional de um quadrado da seqüência da **Figura B.10** e o pior para o mesmo tipo de movimento de objeto em forma da letra *L* da seqüência da **Figura B.3**. Os valores intermediários foram obtidos para as situações que combinavam movimento translacional com rotacional das seqüências das **Figuras B.5**, **B.7** e **B.8**, respectivamente com EQM 0.59323, 0.33513, 0.37289 , sendo estes casos todos com a forma em *L* . Observe que há uma progressão nos movimentos destes três casos, sendo que o primeiro caso de variação translacional e rotacional separadamente, e os outros dois casos sendo a composição destes dois tipos de movimento.

Conclui-se que a medida de erro obtida com o movimento dos três quadros que é composto de rotação e translação de forma composta a partir do método

desenvolvido apresenta menor EQM, mas nos casos em que há um movimento translacional, e só posteriormente um rotacional, este erro aumenta. Também se observou que ao processar imagens com formas retangulares ou circulares e, portanto, para imagens de formas menos complexas, o erro diminui, mas ao trabalhar com imagens em formato de $\underline{L}$, ou seja, com uma maior complexidade em sua forma observou-se que o erro aumenta. Combinando situações nas quais os objetos apresentam forma de $\underline{L}$, com situações de movimento inicialmente rotacional e posteriormente translacional, ou vice-versa, o EQM aumenta.

Veja Tabela 5.1.

5.3.2 Imagens reais com variáveis de entrada controladas

Estas imagens foram obtidas a partir de um conjunto de objetos que são os brinquedos da Sophia e brinquedos do Anandan, sendo que os primeiros nas seqüências das **Figuras B.11** a **B.19** e os segundos nas **Figuras B.20** a **B.23.** Ambos os casos foram feitos movimentos dos objetos de forma controlada, que permitisse uma previsão qualitativa dos resultados, previsão esta que foi confirmada. A faixa do EQM variou de 0,00178 a 9,3064E-14, enquanto a redução de tempo ficou entre 8 a 75 vezes em relação ao método de Horn & Schunck [HORN81].

Observa-se que tanto o EQM quanto a redução de tempo foram aceitáveis tanto quando usados nos brinquedos da Sophia, quanto nos brinquedos de Anandan.

Os casos das seqüências das **Figuras B.1, B.2, B.3, B.4, B.5, B.6, B.7, B.8** e **B.9** foram intencionalmente realizados com movimentos simples de rotação e/ou translação, sem modificar a projeção de suas formas, auxiliando assim o processo de validação de nosso método. Analisando estes resultados, observa-se que a seqüência

IFSC-USP SERVIÇO DE INFORMAÇÃO

da **Figura B.11,** formada por três patos, onde apenas um tem movimento rotacional, apresenta menor EQM. O pior caso em termos de EQM, foi obtido na Seqüência da Figura B.19, formada por um dinossauro, um despertador e um pato, onde apenas este último sofre movimento rotacional. Já em termos de desempenho no tempo de processamento, o pior caso, foi na Seqüência da **Figura B.14,** formada pelo movimento translacional de um pé e uma mão. Observou-se que mesmo os resultados dos piores casos desta classe de imagens foram satisfatórios.

As seqüências com brinquedos de Anandan tiveram em média valores próximos e baixos, independente do movimento presente. O EQM apresentou menor valor na seqüência da **Figura B.21**, e maior valor na da **Figura B.22**. Em todas as imagens cujos movimentos não são mais complexos que aqueles que envolveram os brinquedos da Sophia, o desempenho da técnica em questão é considerável. Houve a preocupação intencional de aproximar a cena gerada de uma cena real. Observou-se que movimentos nos quais a projeção do brinquedo é modificada, propiciam a introdução de erros devido a possibilidade de não correspondência de pixels entre os quadros. Apesar dos problemas presentes, o algoritmo desenvolvido apresentou resultados satisfatórios.Veja Tabela 5.2.

5.3.3 Imagens reais em um barco em movimento

As imagens desta seção são imagens obtidas com câmera de vídeo CCD, a bordo de um barco em movimento na represa do Lobo da cidade de São Carlos (Projeto PACODAHI - Plataforma Autônoma para Coleta de Dados Hidrológicos. Projeto PADCT/ CNPq - 62.0013/98-2). Neste caso não há controle sobre os movimentos, sejam eles unicamente rotacionais, translacionais, com ou sem problemas *ill-posed.* Estes casos são ilustrados nas Seqüências das **Figuras B.36 a B.48**. (Veja no Apêndice B e na Tabela 5.3)

A Seqüência da **Figura B.46** foi a que apresentou menor EQM. A Seqüência de maior EQM foi a da **Figura B.43,** onde observou-se que um leve movimento ondulatório da água modifica a distribuição de intensidade dos pixels da imagem captada, ocasionando uma seleção de região pelo Método de Cabestaing, certamente maior que a equivalente ao verdadeiro movimento do obstáculo à frente do barco. Em geral os resultados obtidos são satisfatórios e apresentam uma redução no tempo de detecção do movimento como mostrado em [OLIVEIRA01], além de baixo erro em relação ao método de *Horn & Schunk* [SCHUNCK81].

Seqüências		FO com Cabestaing		
	tempo (s)	tempo (s)	razão entre os tempos	eqm
Figura B.36	1,040	0,220	4,727	0,00012
Figura B.37	1,040	0,310	3,355	1,99300
Figura B.38	1,040	0,220	4,727	0,02964
Figura B.39	1,050	0,220	4,773	0,00052
Figura B.40	1,040	0,110	9,455	0,02427
Figura B.41	1,040	0,220	4,727	1,30934
Figura B.42	1,050	0,170	6,176	0,00323
Figura B.43	1,460	0,170	8,588	9,88748
Figura B.44	1,100	0,160	6,875	3,56998
Figura B.45	1,040	0,160	6,500	0,04139
Figura B.46	1,050	0,220	4,773	0.00012
Figura B.47	1,100	0,330	3,333	0,00025
Figura B.48	0,820	0,160	5,125	0,91558

Tabela 5.3. Resultados da aplicação em imagens reais com barco em movimento.

5.3.4 Outras imagens de teste

Outras imagens foram utilizadas nesta tese para testes do algoritmo e são elas as seqüências das **Figuras B.24** a **B.35**. As seqüências das **Figuras B.24** a **B.26** foram obtidas com uma câmera CCD de nosso laboratório. São imagens de leves movimentos de um bebê, em resposta aos estímulos externos, resumindo-se basicamente em movimentos do tronco, cabeça e mãos em busca de objeto exibido. Os resultados corresponderam ao esperado, principalmente se comparando ao método de [HORN81], ou seja, apresentou baixo EQM. As **Figuras B.27** a **B.32** são imagens de tráfego em uma rua, e foram feitas com a mesma câmera, na janela de nosso laboratório, observando o movimento de pedestres, automóveis e motos. Tomamos o cuidado de excluir horários de nascer e pôr do sol, em que há uma variação na intensidade de luz, e por isso, introduzem mais erro.

É interessante notar a resposta de nosso método, detectando de forma correta, mesmo com cenário fixo, um ou mais objetos em diferentes formatos e sentidos de movimento, se presentes na cena.

As **Figuras B.33** a **B.35** foram feitas a bordo do carro em movimento. Além do cenário em movimento, há o movimento do tráfego em sentido oposto.

Ver Tabela 5.4.

Seqüências	FO	FO com Cabestaing		
	tempo (s)	tempo (s)	razão entre os tempos	eqm
Figura B.24	1,050	0,220	4,773	0,00194
Figura B.25	1,040	0,280	3,714	0,03371
Figura B.26	1,040	0,280	3,714	0,00152
Figura B.27	0,490	0,050	9,800	0,00559
Figura B.28	0,490	0,050	9,800	4,7419E-05
Figura B.29	0,490	0,110	4,455	0,00232
Figura B.30	1,120	0,050	22,400	0,00133
Figura B.31	0,500	0,050	10,000	0,00332
Figura B.32	0,490	0,110	4,455	0,00166
Figura B.33	1,040	0,110	9,455	0,00032
Figura B.34	1,100	0,110	10,000	0,03023
Figura B.35	1,040	0,160	6,500	0,00015

Tabela 5.4. Resultados da aplicação do algoritmo em outros tipos de imagens .

5.4 Análise comparativa dos resultados

Métodos de detecção de movimento estão sujeitos a erros, o que nossos exemplos vêm a corroborar. Em imagens sintéticas, destacou-se o erro referente ao movimento ora translacional, ora rotacional, em contrapartida ao formado pela composição dos dois. O primeiro introduzindo mais erro no processo de estimação do movimento, haja vista que, ao contrário do movimento composto, aquele apresenta maiores descontinuidades na imagem, o que interfere no cálculo dos gradientes das distribuições de intensidade dos *pixels*.

Há também entre as imagens sintéticas, a observância daquelas imagens com maior complexidade em sua forma, cuja estimação de movimento é afetada, pelas maiores chances de não correspondência entres seus pixels, dentro de uma dada seqüência, ou seja, menos imune a situações de oclusão.

Nas situações reais, destacam-se aquelas em que ocorre movimento rotacional que modifica a projeção do objeto, assim como cenas com iluminação variável, como se pode observar no apêndice B.

Veja os gráficos mostrados respectivamente nas Figuras 5.1 e 5.2, onde são mostrados os resultados em cor azul para a simulação do método clássico (sem filtro de Cabestaing) e em cor vermelha para o método desenvolvido nesta tese (com filtro de Cabestaing). A ordenada mostra o tempo de processamento e as abscissas mostram as devidas seqüências (as mesmas para ambos os gráficos).

SERVIÇO DE BIBLIOTECA
INFORMAÇÃO

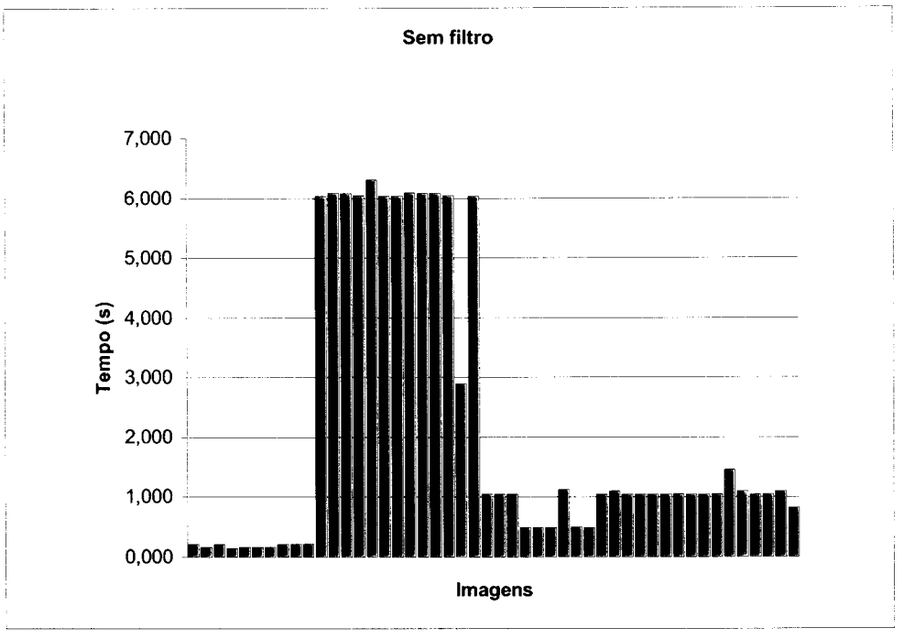


Figura 5.1. Resultados obtidos com método clássico (sem filtro de Cabestaing).

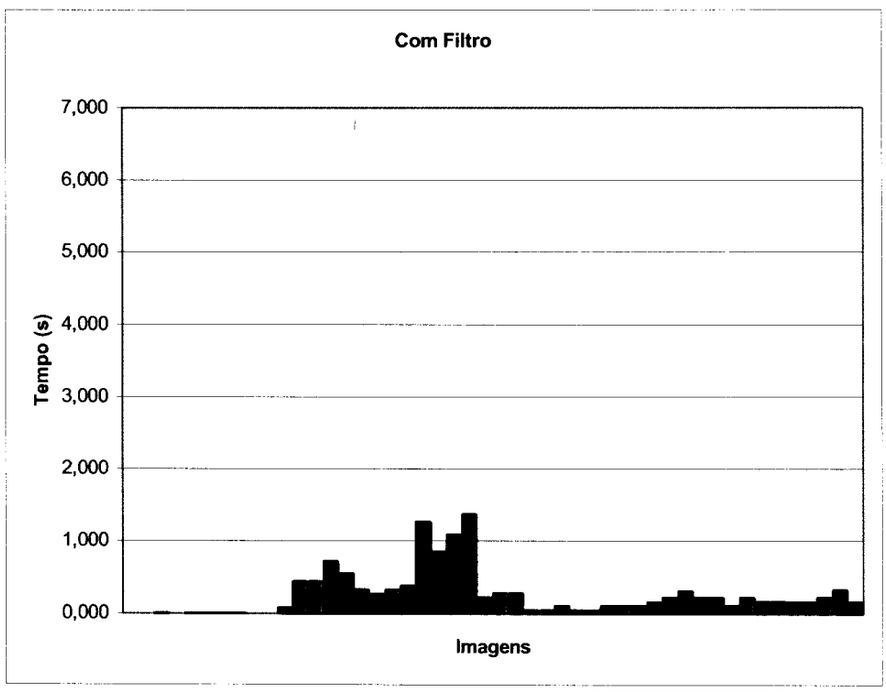


Figura 5.2. Resultados obtidos com método desenvolvido na tese (com filtro de Cabestaing).

5.5 Conclusões

O algoritmo desenvolvido, como foi demonstrado pelas análises dos resultados de cada grupo de seqüências, tem respostas com distintas qualidades, apesar de todas satisfatórias e também dependentes do movimento da cena.

As situações de erro em estimação de movimento existem, e minimizá-las é o desafio das pesquisas já desenvolvidas e ainda em desenvolvimento. Otimizar a detecção de movimento de forma a fazê-la mais rápida, sem afetar sua precisão, requer um estudo relevante sobre a relação custo-benefício destes dois parâmetros. Como é mostrado em [OLIVEIRA03], a técnica desenvolvida é eficiente quando diante desta missão: torna o tempo de processamento na detecção de movimento bem menor, mantendo a precisão no mínimo próxima das técnicas da literatura. Mesmo diante de tal eficiência restam muitos desafios, como por exemplo, o de minimizar o ruído deixado na técnica de redução de área útil para cálculo de FO. Esta minimização de ruído poderá trazer menores erros atrelados a menor tempo de processamento, resultado que seria o ideal.

6 Conclusão

6.1 Contribuições deste trabalho

Neste trabalho, após exaustiva revisão bibliográfica das técnicas disponíveis na literatura para detecção de movimento, desenvolvemos uma técnica que combina a técnica de detecção de bordas dos objetos em movimento na imagem com a técnica de Fluxo ótico. Esta combinação constitui uma linha promissora para futuras investigações. O ato de combinar técnicas consagradas na literatura tem sido uma das tendências de estudos nesta área. Cornelia Fermüller em [FERMÜLLER01], por exemplo, estuda três categorias de técnicas de fluxo ótico (baseadas em gradiente, energia e correlação), analisando diferentes formas de combiná-las em estimações unidimensionais de movimento, e assim compor estimações bidimensionais.

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

6.2 Conclusões Gerais

O tempo de execução do algoritmo de obtenção de fluxo ótico baseado no método de Horn e Schunck [HORN81] é superior ao tempo de execução do algoritmo que combina a mesma técnica com a redução de área útil de cálculo de FO, através de pré-processamento por detecção de bordas. Esta redução de tempo mostra a possibilidade de utilização de tal técnica no autoguiamento de robôs, inclusive de barcos. Há ainda a possibilidade de a partir da redução de tempo, se utilizar o tempo restante para outras aplicações como em tomadas de decisão de novas trajetórias para o robô ou mesmo o barco.

O Erro Quadrático Médio entre os diagramas de agulhas pode auxiliar na quantificação da semelhança do método proposto em relação ao existente na literatura. A utilização de imagens sintéticas e imagens reais com variáveis de entrada controladas foi imprescindível na análise dos resultados obtidos, assim como das situações críticas ao cálculo do FO, como as de oclusão ou as com problema e abertura.

Esta tese poderá auxiliar um outro projeto de nosso grupo de pesquisa - a construção de um robô náutico autoguiado capaz de desviar-se de obstáculos estáticos e dinâmicos, que contém uma sonda desenvolvida em [BRUNO97] e [SOUZA00] para coleta de dados hidrológicos (Projeto PACODAHI, Plataforma Autônoma para Coleta de Dados Hidrológicos. Projeto PADCT/ CNPq - 62.0013/98-2).

6.3 Trabalhos Futuros

Para estudos futuros são listados alguns pontos:

- Investigar a associação de outras técnicas para redução de área útil para cálculo do Fluxo ótico.
- Desenvolver estudos comparativos da associação da técnica de detecção de bordas com diferentes métodos de cálculo de fluxo ótico.
- Uso da técnica proposta em outras plataformas computacionais, como DSP's.
- Aprimorar a implementação desta técnica com intuito de reduzir mais ainda o tempo de processamento.
- Investigar a possibilidade de sua utilização em diagnóstico por imagens médicas.

7 Referências Bibliográficas

[ANANDAN89]

Anandan, P.; "A computational Framework and an Algorithm for the Measurement of Visual Motion", *International Journal of Computer Vision*, 2, 283-310, 1989.

[BARRON94]

Barron, J. L.; Fleet, D. J. e Beauchemin, S. S.; "Systems and Experiment Performance of Optical Flow Techniques", *International Journal of Computer Vision*, 12:1, 43-77, 1994.

[BERTERO88]

Bertero, M.; Poggio, T. A.; Torre V.; "Ill-posed problems in early vision", *Proceedings IEEE*, Vol. 76, p 869-889, August 1988.

[BOGHOSSIANB99]

Boghossian, B. A.; Velastin, S. A.; "Motion-Based Machine Vision Techniques for the Management of Large Crowds", *IEEE 6th International Conference on Electronics, Circuits and Systems (ICICS'99)*, Cyprus, p. 961-964, September 5-8, 1999.

[BRUNO97]

Bruno, R.; "Presque's em instrumentação eletrônica microprocessada para a medida de parâmetros físicos e químicos da água." ; 145p. *Dissertação de mestrado; Instituto de Física de São Carlos -Universidade do Estado de São Paulo*; fevereiro de 1997.

[BÜLTHOFF89]

Bülthoff, H.; Little, J.; Boggio, T.; "A parallel algorithm for real-time computation of optical flow", *Nature*, vol. 337 / 9, p. 549-553, february 1989.

[CABESTAING91]

Cabestaing, F.; Postaire, J.G.; Bonnet, P.; "Real Time Movement Detection: The STREAM Processor", CAMP91, *Computer Architecture for Machine Perception*, p.279-284, Paris, December 16-18, 1991.

[CAMUS94]

Camus. T.; "Real Time Optical Flow", *PhD Thesis*, Department of computer Science, Brown University, Providence, RI 02912, USA, September , 1994.

[CAMUS96]

Camus. T.; Coombs, D.; Herman, M.; Tsai-Hong H.; "Real Time Single-workstation Obstacle Avoidance Using Only Wide-Field Flow Divergence", Proceedings of the 13th International Conference on Pattern Recognition Applications and Robotic Systems, Vol. 3, pp. 323 - 330, Vienna, Austria, August 1996.

[CAMUS97]

Camus. T.; "Real Time Quantized Optical Flow", *Real Time Imaging* 3, 71-86, 1997.

[CHHABRA97]

Chhabra, A. K.; "Real-time Computation of Optical Flow along Contours of significant Intensity Change", *Real-time imaging* 3, 87-99, 1997

[COSTA 94]

Costa, L. da F.; Roda, V. O .; Köberle, R.; "A biologically-inspired system for visual pattern recognition", *1994 IEEE International Symposium on Industrial Electronics*, pp. 25-30, May 25-27, 1994

[DUCHON95]

Duchon, A . P.; Warren, W. H.; Kaelbling, L. P.; "Ecological Robotics: Controlling Behavior with Optical Flow", *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, Pittsburgh, PA, July, 164-169, 1995.

[FERMÜLLER01]

Fermüller, C.; Shulman, D.; Aloimonos, Y.; "The statistics of Optical Flow", *Computer Vision and Image Understanding*, vol. 82, 1-32, 2001

[FILHO99]

Filho, O. M.; Neto, H. V.; Processamento Digital de Imagens, São Paulo Brasport, 406p, 1999.

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

[FILHO01]

Filho, M. S.; Schneebeli, H. A .; Caldeira, E. M. O .; "Cálculo do Fluxo Ótico em Tempo Real e sua Utilização na Navegação de Robôs Móveis", *V Simpósio Brasileiro de Automação Inteligente*, novembro de 2002, Mídia Press, Canela, RS.

[GONZALEZ92]

Gonzalez, R.C. ; Woods, R.E. ; Digital Image Processing, Addison-Wesley, 1992.

[HORN81]

Horn, B.; Schunk, B; "Determining Optical Flow", *Artificial Intelligence* 17, 185-203, 1981.

[KIM01]

Kim, L.; Kim, J.; "Effective nonlinear approach for optical flow estimation"; *Signal Processing;* pp 2249-2252;V 81, issue 10, October 2001.

[LOW91]

Low, A.; Introductory Computer Vision and Image Processing, McGraw-Hill Book Company, 1991.

[LUCAS81]

Lucas, B. D.; Kanade, T.; "An interative image registration technique with an application to stereo vision", *Proc. DARPA Image Understanding Workshop*, pp. 121-130, 1981.

[MARR82]

Marr, D; "Vision", W. H. Freeman, 1982

[NAGEL87]

Nagel, H. H.; "On the estimation of optical flow: relations between different approaches and some new results", *Artificial Intelligence*, vol.33, pp. 299-324, 1987

[NELSON87]

Nelson, R. & Aloimonos, J.; "Obstacle avoidance using flow field divergence", IEEE *Pattern Analysis and Machine Intelligence* -11, 10:1102-1106, 1987.

[OLIVEIRA01]

Oliveira, I. N. S.; Roda, V. O .; "Detecção rápida de movimento com aplicação em sistemas de navegação autônoma". *Primeira Escola Brasileira de Aplicações em dinâmica e Controle*, 25-31, agosto de 2001, São Carlos, SP.

[OLIVEIRA02]

Oliveira, I. N. S.; Roda, V. O .; "Metodologia para detecção rápida do movimento em seqüências de imagens". *V Simpósio Brasileiro de Automação Inteligente*, novembro de 2002, Mídia Press, Canela, RS.

[OLIVEIRA03]

Oliveira , I. N. S. ; Roda ,V. O; Modugno, R. ; "*Optimizing the Processing Time of the Optical Flow in Image Sequences*", submetido a SIBIGRAPI 2003, XVI

Brazilian Symposium on Computer Graphics and Image Processing, October 12-15, 2003.

[PATIÑO97]

Patiño, M. S. ; "Metodologia para detecção de movimento utilizando a técnica de fluxo ótico", 187p , *tese de doutoramento no Instituto de Física de São Carlos - Universidade do Estado de São Paulo*, 1997.

[PAZINATO00]

Pazinatto, F. A .C.; "Algoritmo para seleção automática de reprodutores animais baseado em técnicas de processamento digital de imagens", 97p, *dissertação de mestrado no Departamento de Computação*, Universidade Federal de São Carlos, 2000.

[SABE96]

Sabe, K.; Hashimoto, H.; Harashima, F.; "Analysis on Real Time Motion Estimation with Active Stereo Camera", *AMC'96- MIE. 1996 4th International Workshop on Advanced Motion Control Proceedings*, vol. 1, 365-371, 1996.

[SOUZA00]

SOUZA, M. M. S. de; "Monitor microprocessado para medição de variáveis hidrológicas."; 98p . *Dissertação de mestrado; Instituto de Física de São Carlos - Universidade do Estado de São Paulo*; julho de 2000.

[SUN02]

Sun, Changming, "Fast optical flow using 3D shortest path techniques", *Image and Vision Computing*, vol 20, issues 13-14, 981-991, 2002.

[TEKALP95]

Tekalp, A . M.; "Digital Video Processing", Prentice Hall, PTR, 1995.

[URAS88]

Uras, S.; Girosi, F.; Verri, A.; Torre, V.; "A computacional Approach to motion perception", *Biol. Cybern.*, vol.60, pp.79-97, 1988.

[ZHANG02]

Zhang, Ye; Kambhamettu; "3D head tracking under partial occlusion", *Pattern Recognition*, 35, 1545-1557, 2002

APÊNDICE A

Erro Quadrático

Uma medida que pode ser utilizada para avaliar erro , é o Erro Médio Quadrático, que é dado pela equação:

$$MSE = \frac{1}{MN} \sum_{i,j=1}^{M,N} \left[\left(x(i,j) - y(i,j) \right) \right]^2$$

Onde **x**, no caso em estudo, corresponde ao fluxo ótico original obtido por uma única técnica (por exemplo, obtido por Horn e Schunck) em função de **i** e **j** que são as posições de um dado pixel na imagem de tamanho *M x N*. A função y estimada corresponde ao Fluxo ótico obtido através da combinação da técnica proposta por [HORN81] para obtenção de x com pre-processamento [CABESTAING91] . Assim para cada posição (i, j) entre as imagens, faz-se a diferença entre as respectivas distribuições de intensidade, somando-as dentro de um somatório que inicia na primeira posição destas imagens, finalizando na posição (M,N), e posteriormente dividindo este somatório pelo tamanho total da imagem. Não havendo nenhuma diferença entre o novo método e o clássico, o Erro Quadrático Médio deveria ser nulo, ou bem próximo de zero.

SERVIÇO DE BIBLIOTECA
INFORMAÇÃO

APÊNDICE B

Resultados de Aplicação do Método Desenvolvido

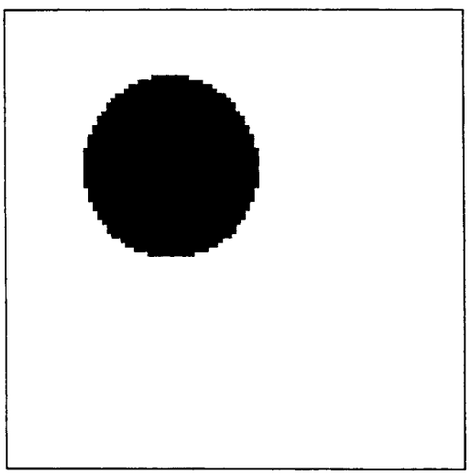


Figura B.1.1. Primeiro quadro de uma seqüência sintética.

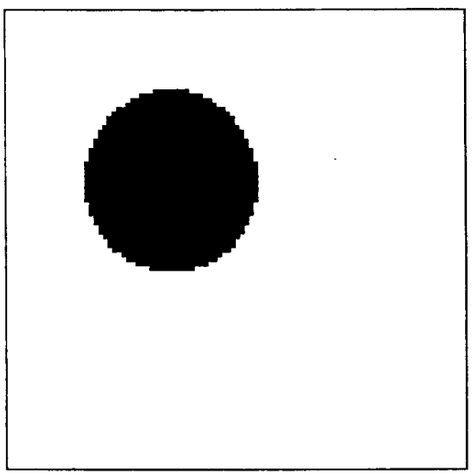


Figura B.1.2. Segundo quadro da seqüência.

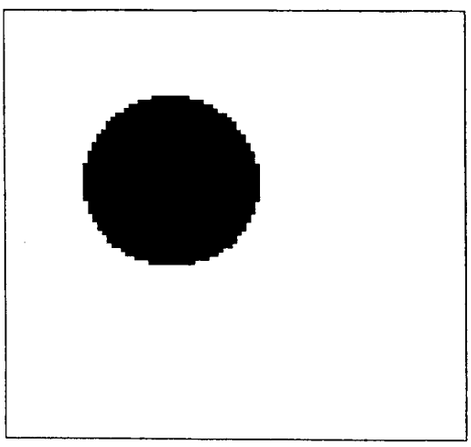


Figura B.1.3. Terceiro quadro da seqüência.

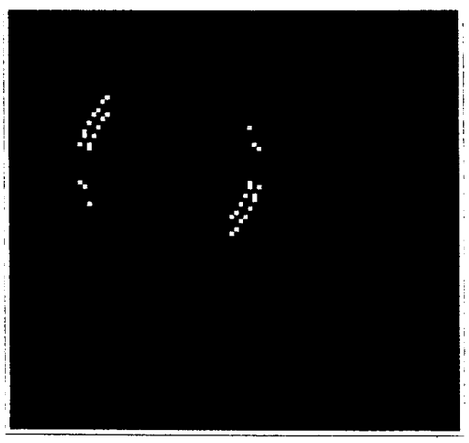


Figura B.1.4. Resultado da aplicação do Cabestaing na seqüência.

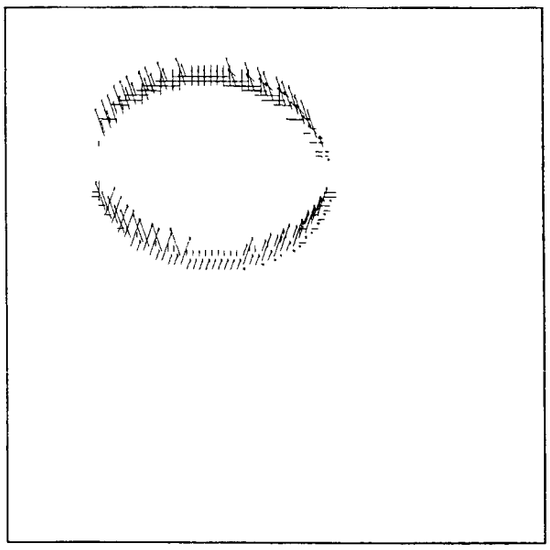


Figura B.1.5. Fluxo ótico da seqüência sintética obtido sem máscara de Cabestaing

Tempo de processamento: 0,220 s

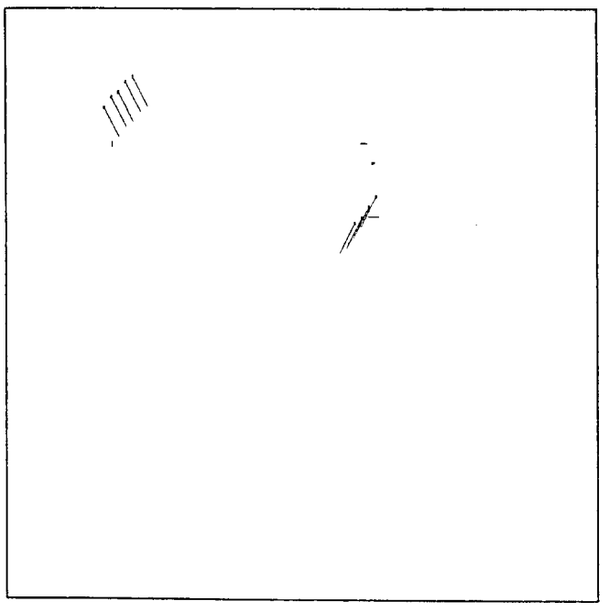


Figura B.1.6. Fluxo ótico obtido com Máscara de Cabestaing

Tempo de processamento: 1,803 ms

Erro Médio Quadrático: 0,31932

Relação entre os tempos de processamento: 122,02 vezes

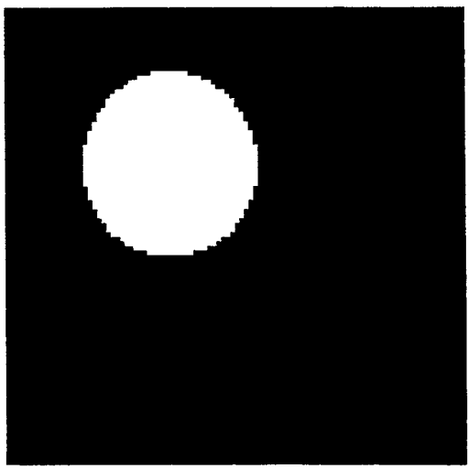


Figura B.2.1. Primeiro quadro de uma seqüência sintética.

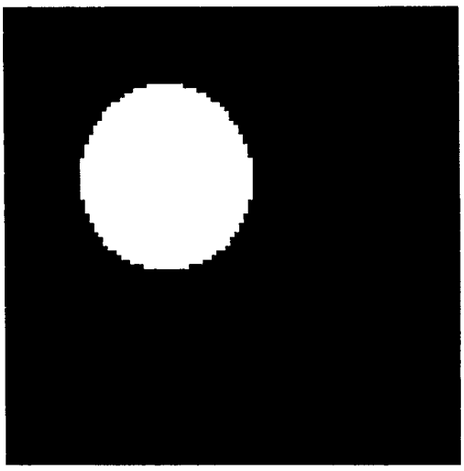


Figura B.2.2. Segundo quadro da seqüência.

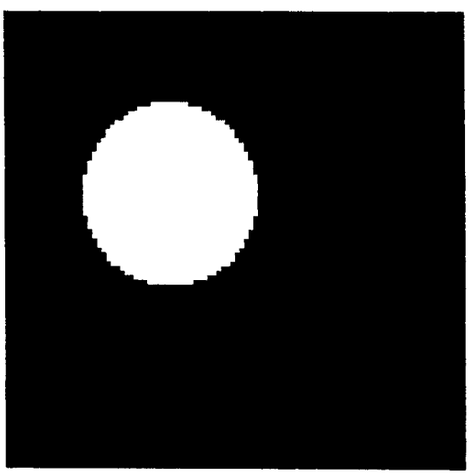


Figura B.2.3. Terceiro quadro da seqüência.

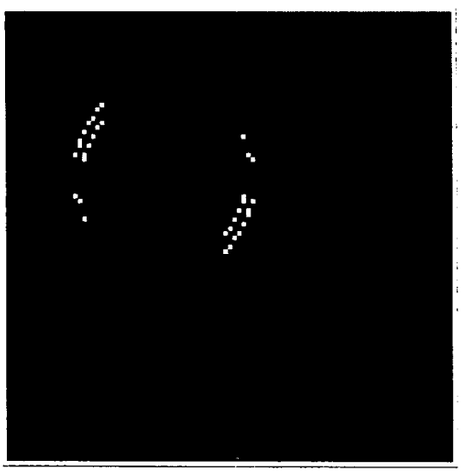


Figura B.2.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

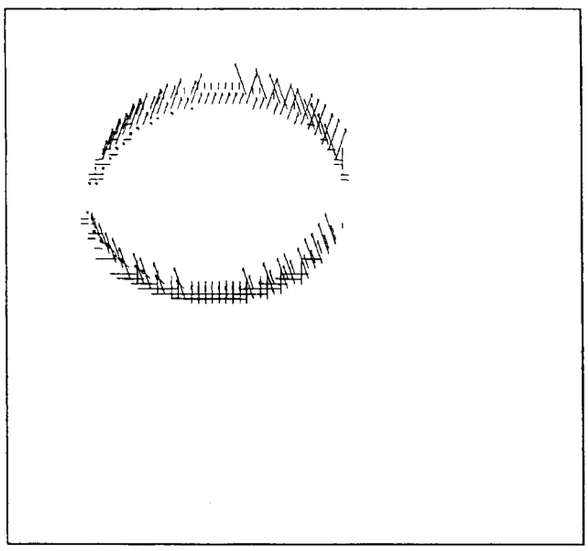


Figura B.2.5. Fluxo ótico da seqüência sintética obtido sem máscara de Cabestaing.

Tempo de processamento: 0,160 s

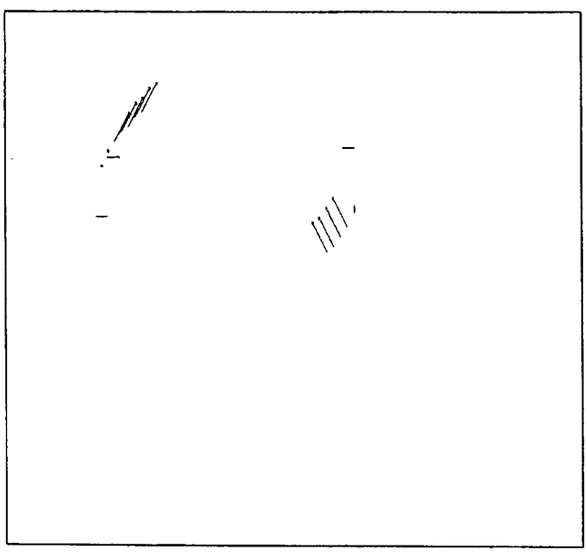


Figura B.2.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 1,805 ms

Erro Médio Quadrático: 0,33321

Relação entre os tempos de processamento: 88,64 vezes

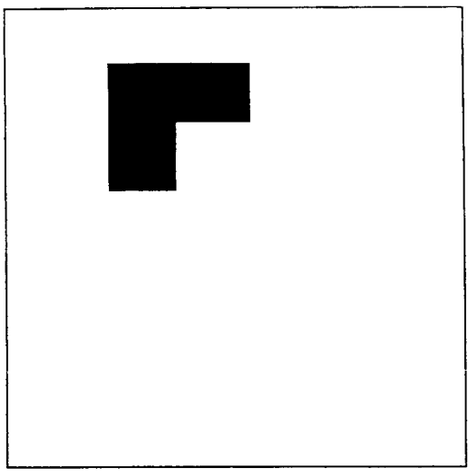


Figura B.3.1. Primeiro quadro de uma seqüência sintética.

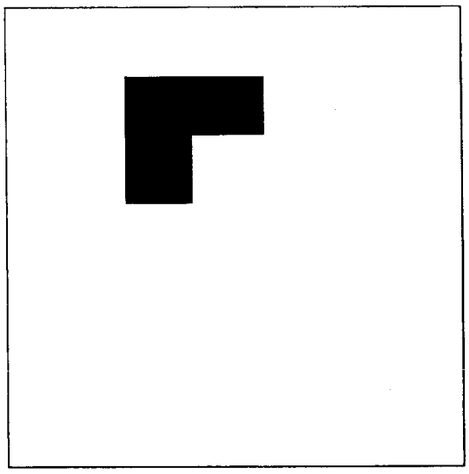


Figura B.3.2. Segundo quadro da seqüência.

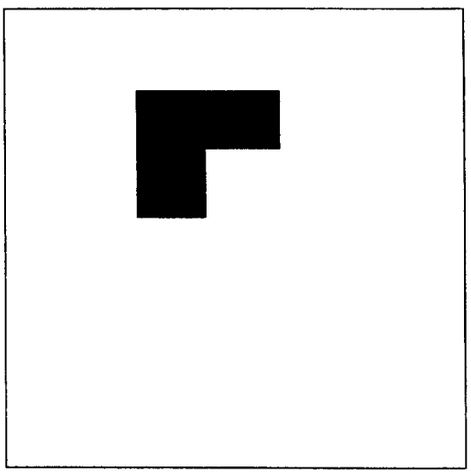


Figura B.3.3. Terceiro quadro da seqüência.

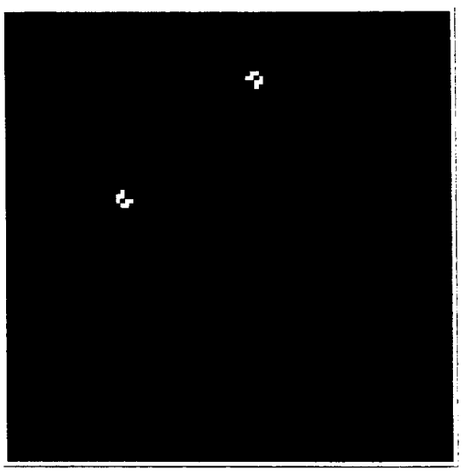


Figura B.3.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

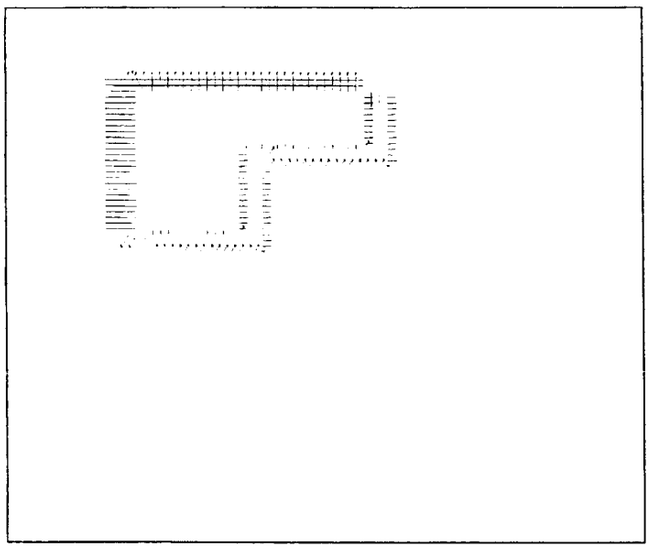


Figura B.3.5. Fluxo ótico da seqüência sintética obtido sem máscara de Cabestaing.

Tempo de processamento: 0,220 s

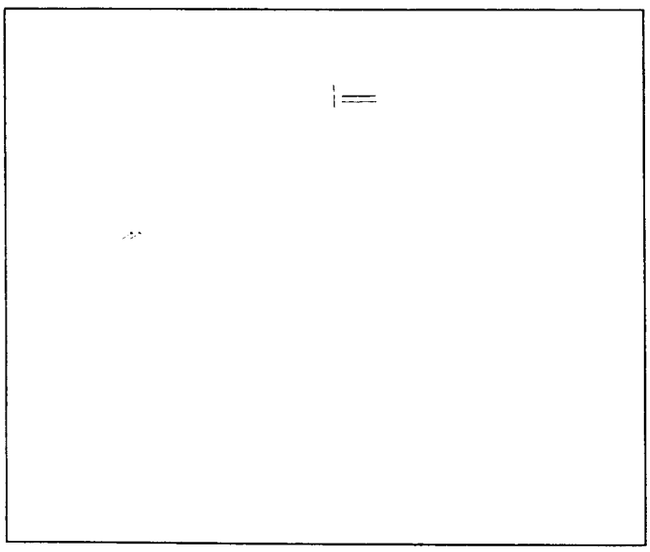


Figura B.3.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 3,392 ms

Erro Médio Quadrático: 0,71686

Relação entre os tempos de processamento: 63,00 vezes

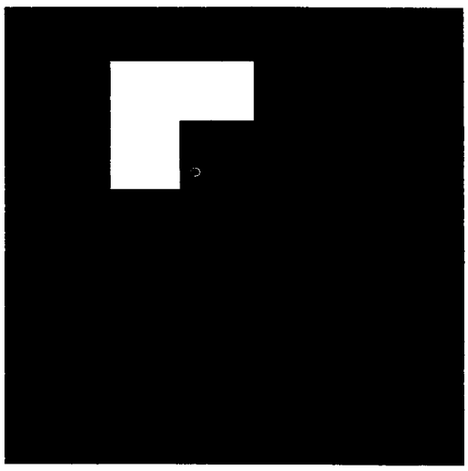


Figura B.4.1. Primeiro quadro de uma seqüência sintética.

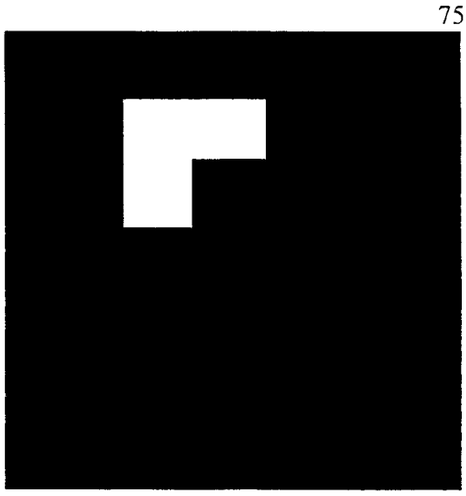


Figura B.4.2. Segundo quadro da seqüência.

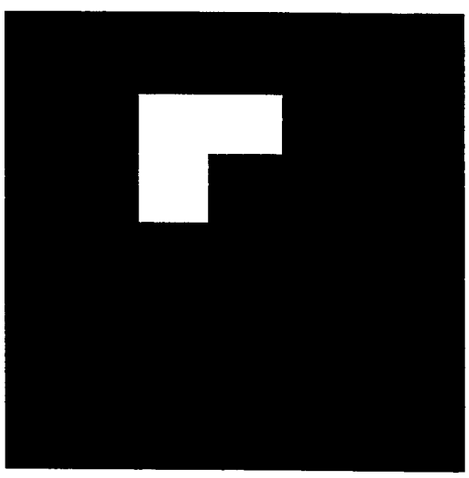


Figura B.4.3. Terceiro quadro da seqüência.

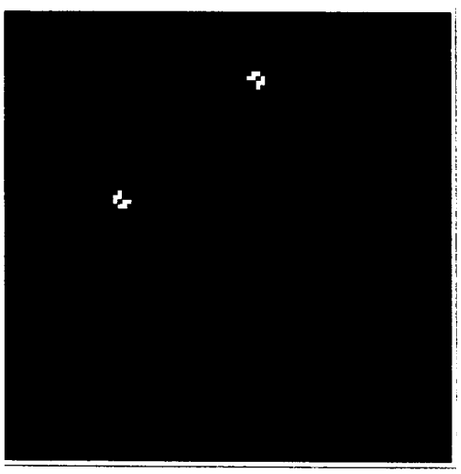


Figura B.4.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

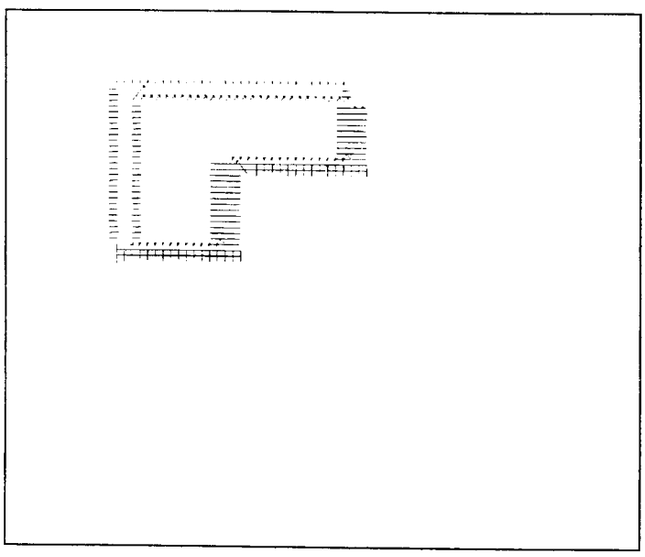


Figura B.4.5. Fluxo ótico da seqüência sintética obtido sem máscara de Cabestaing.

Tempo de processamento: 0,850

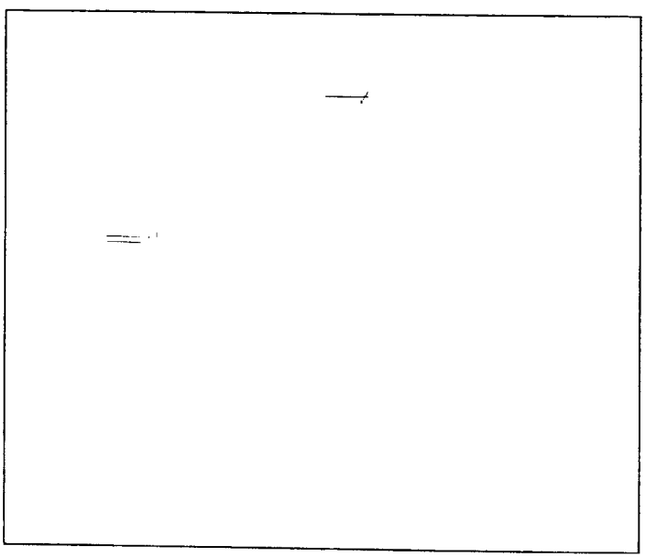


Figura B.4.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 1,336 ms

Erro Médio Quadrático: 0,69183

Relação entre os tempos de processamento: 112,28 vezes

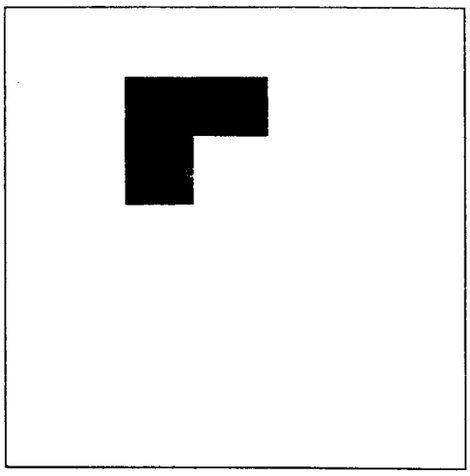


Figura B.5.1. Primeiro quadro de uma seqüência sintética.

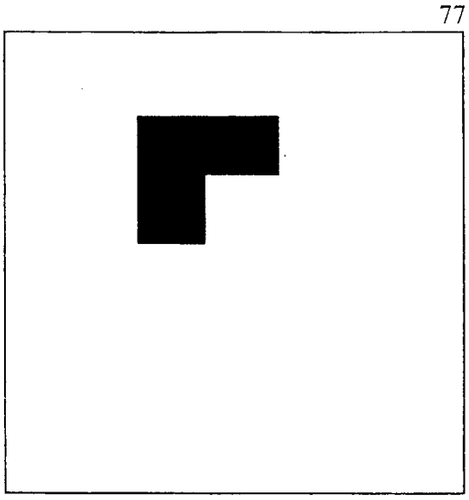


Figura B.5.2. Segundo quadro da seqüência.

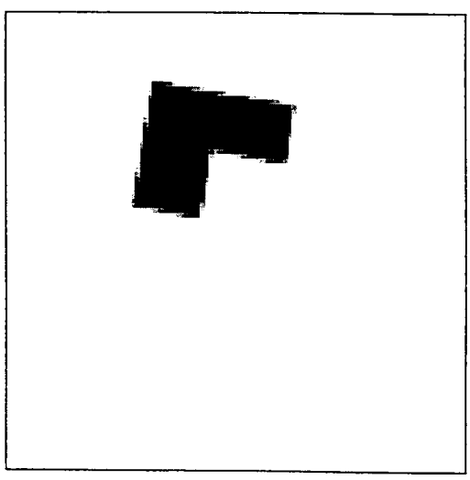


Figura B.5.3. Terceiro quadro da seqüência.

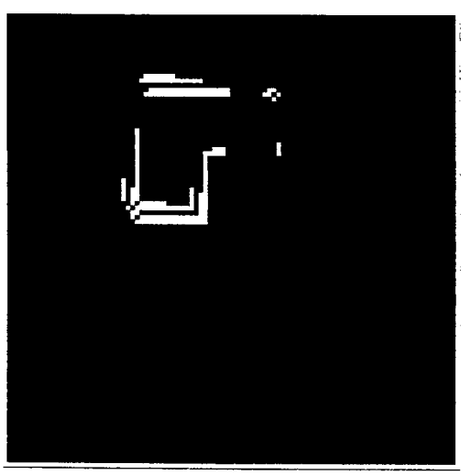


Figura B.5.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

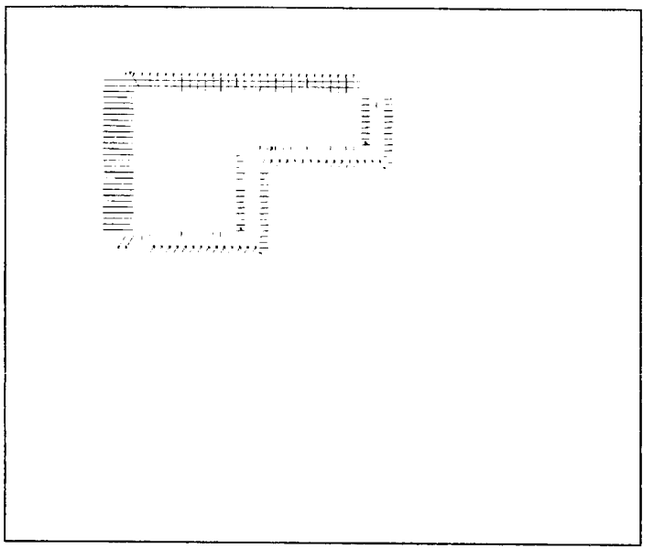


Figura B.5.5. Fluxo ótico da seqüência sintética obtido sem máscara de Cabestaing.

Tempo de processamento: 0,170 s

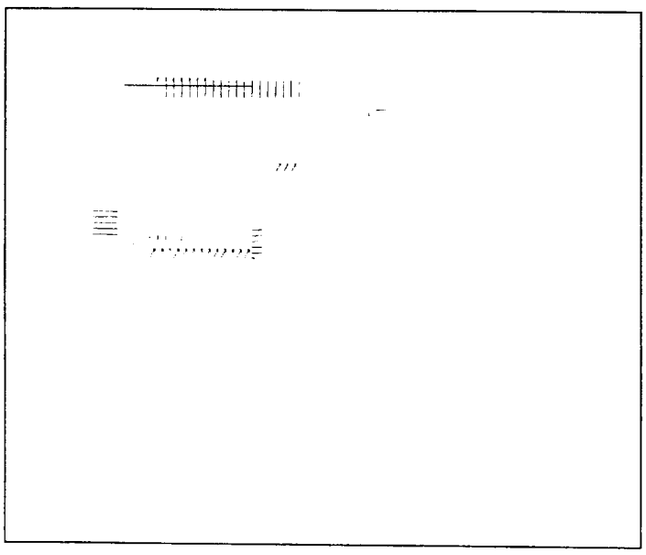


Figura B.5.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 6,621 ms

Erro Médio Quadrático: 0,59323

Relação entre os tempos de processamento: 31,68 vezes

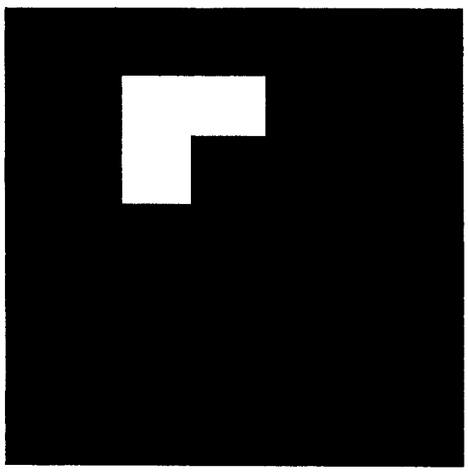


Figura B.6.1. Primeiro quadro de uma seqüência sintética.

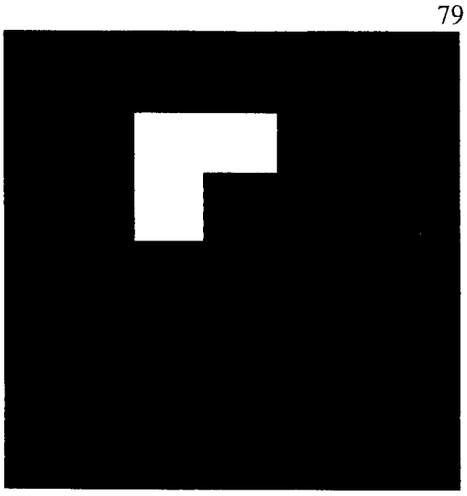


Figura B.6.2. Segundo quadro da seqüência.

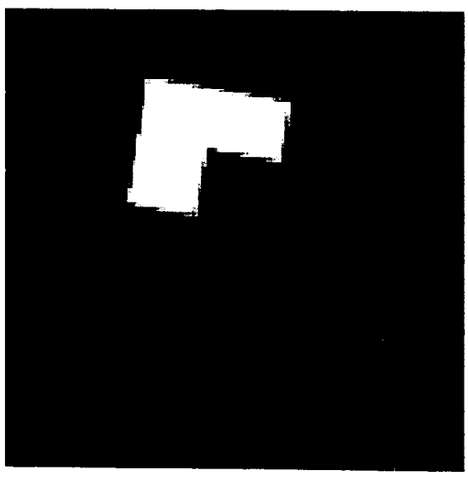


Figura B.6.3. Terceiro quadro da seqüência.

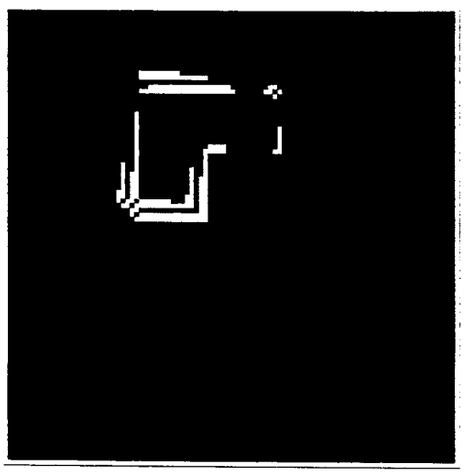


Figura B.6.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

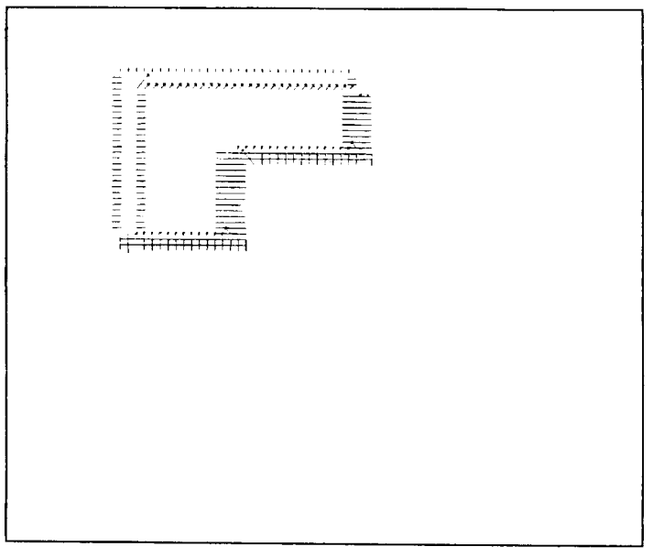


Figura B.6.5. Fluxo ótico da seqüência sintética obtido sem máscara de Cabestaing.

Tempo de processamento: 0,170 s

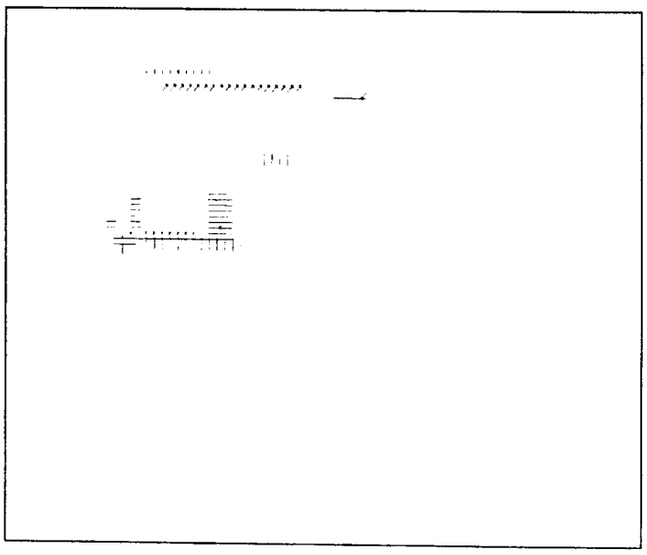


Figura B.6.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 6,160 ms

Erro Médio Quadrático: 0,56128

Relação entre os tempos de processamento: 27,60 vezes

IFSC-USP SERVIÇO DE BIBLIOTECA E INFORMAÇÃO

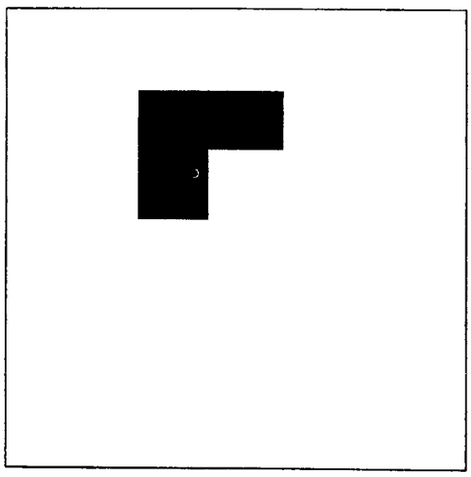


Figura B.7.1. Primeiro quadro de uma seqüência sintética.

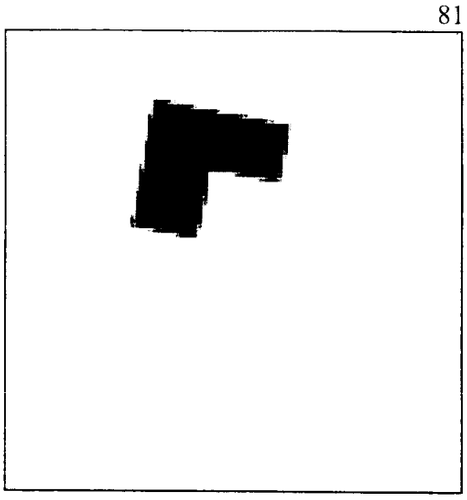


Figura B.7.2. Segundo quadro da seqüência.

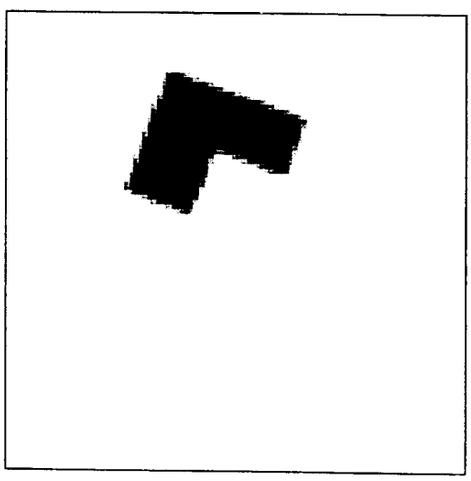


Figura B.7.3. Terceiro quadro da seqüência.

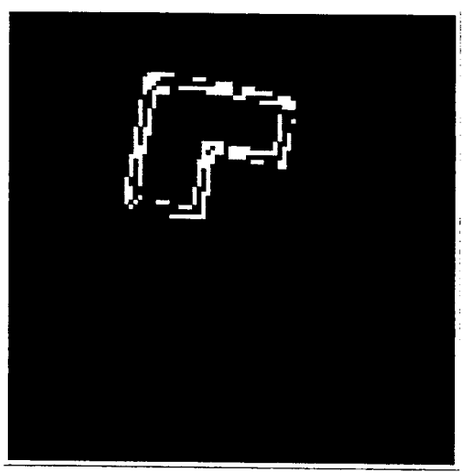


Figura B.7.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

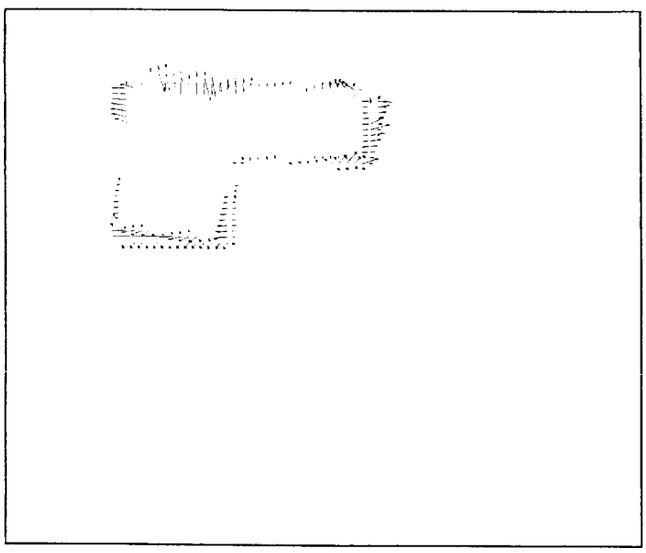


Figura B.7.5. Fluxo ótico da seqüência sintética obtido sem máscara de Cabestaing.

Tempo de processamento: 0,170 s

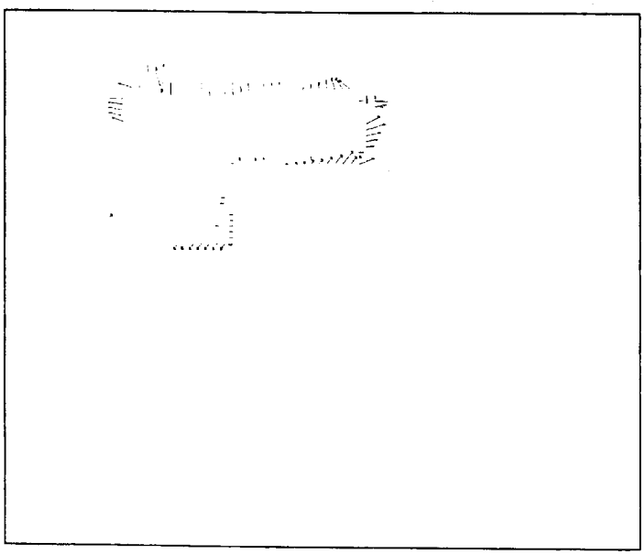


Figura B.7.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 6,621 ms

Erro Médio Quadrático: 0,33513

Relação entre os tempos de processamento: 25,67 vezes

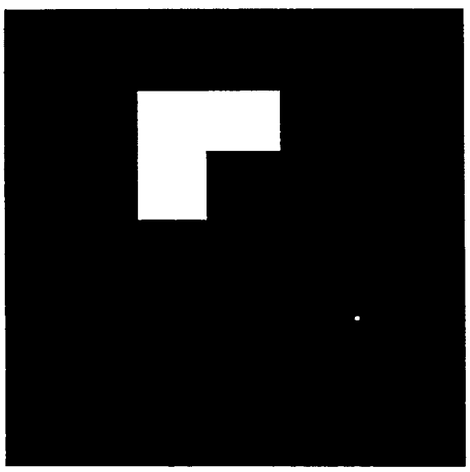


Figura B.8.1. Primeiro quadro de uma seqüência sintética.

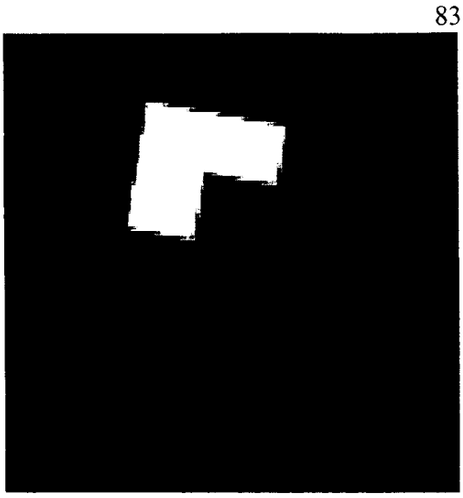


Figura B.8.2. Segundo quadro da seqüência.

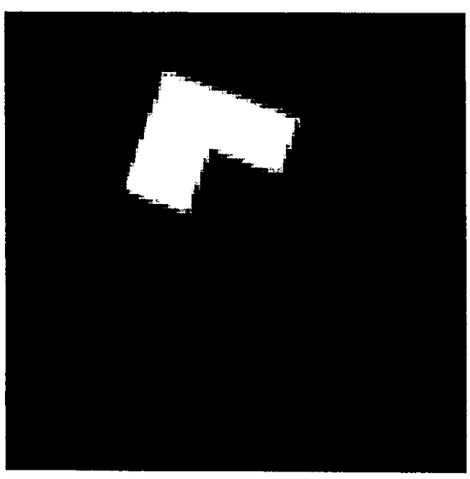


Figura B.8.3. Terceiro quadro da seqüência.

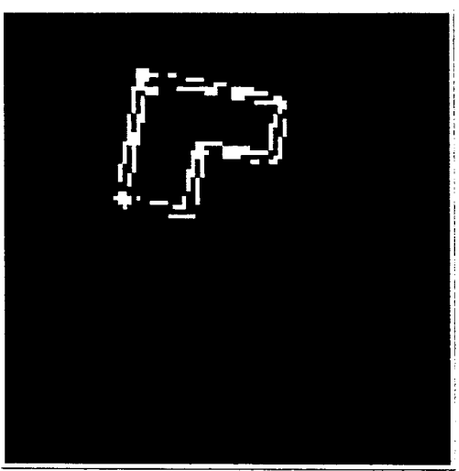


Figura B.8.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

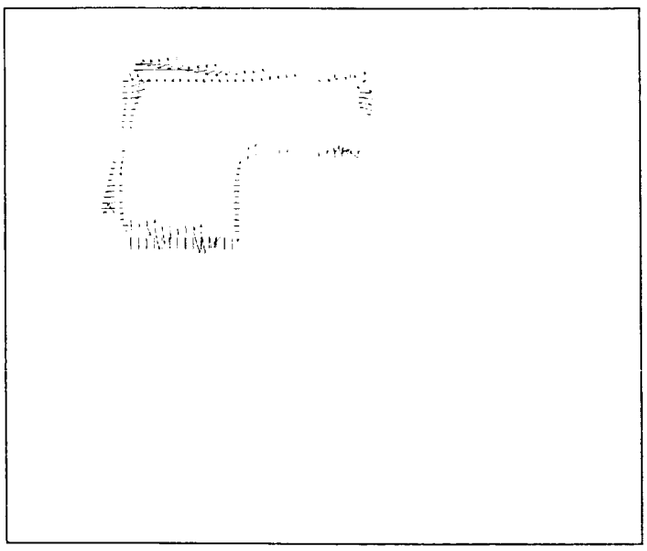


Figura B.8.5. Fluxo ótico da seqüência sintética obtido sem máscara de Cabestaing.

Tempo de processamento: 0,220 s

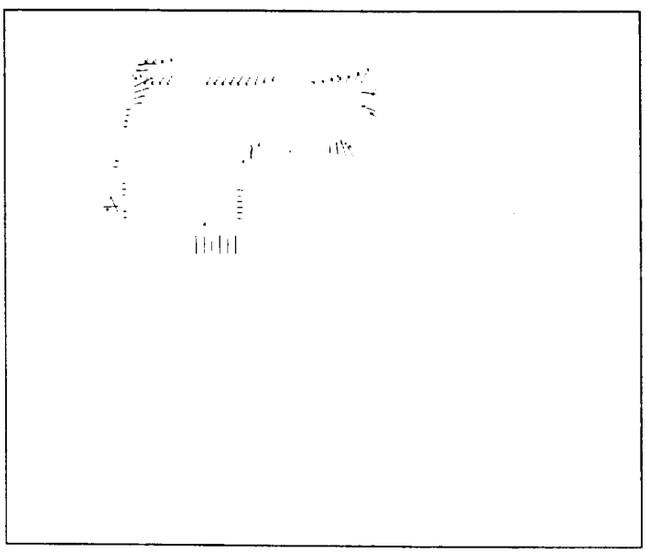


Figura B.8.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 6,405 ms

Erro Médio Quadrático: 0,37289

Relação entre os tempos de processamento: 34,35 vezes

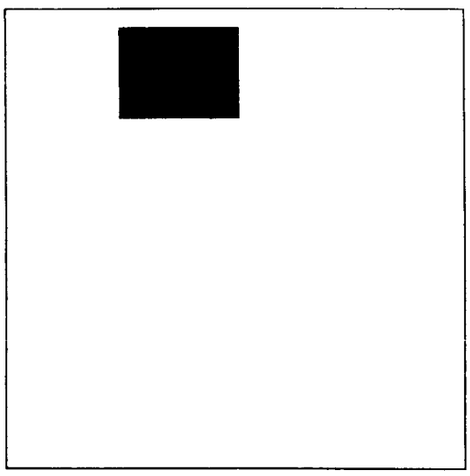


Figura B.9.1. Primeiro quadro de uma seqüência sintética.

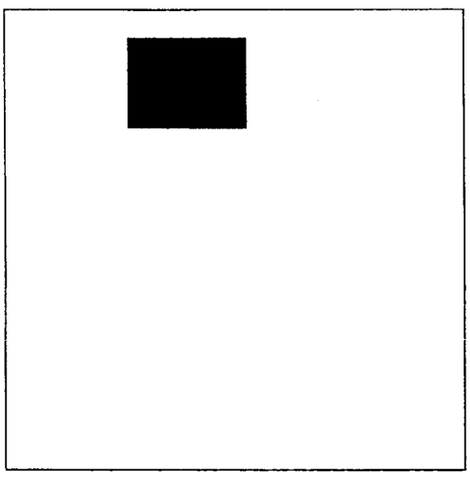


Figura B.9.2. Segundo quadro da seqüência.

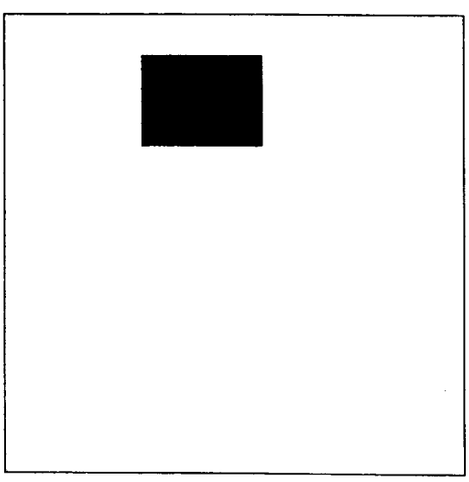


Figura B.9.3. Terceiro quadro da seqüência.

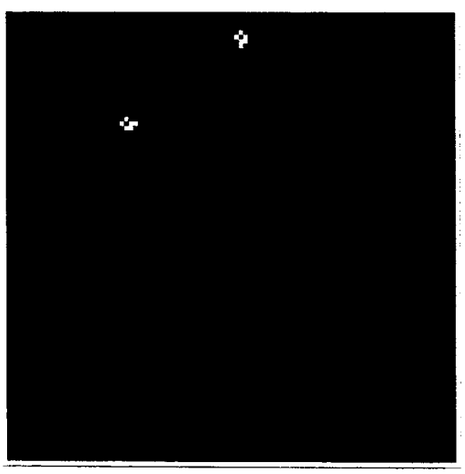


Figura B.9.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

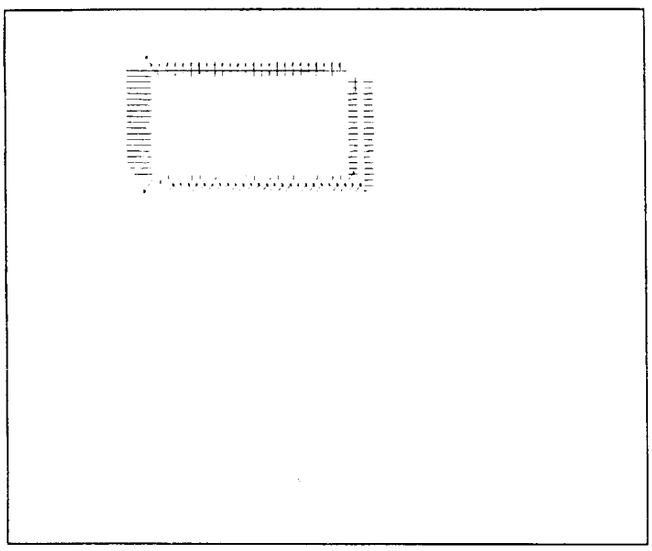


Figura B.9.5. Fluxo ótico da seqüência sintética obtido sem máscara de Cabestaing.

Tempo de processamento: 0,220 s

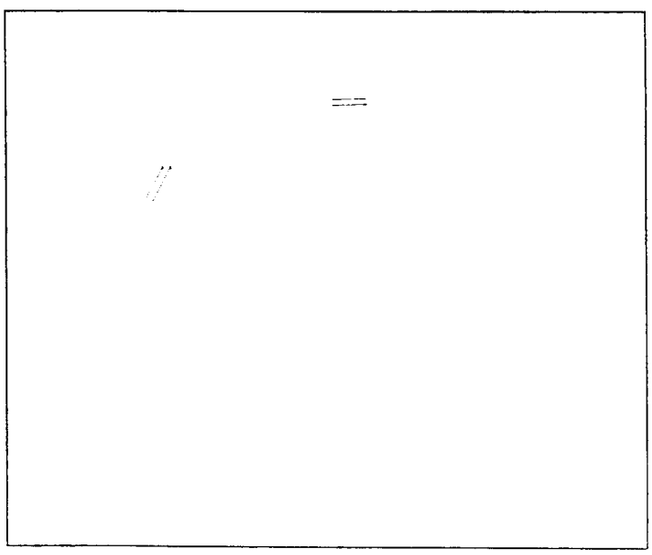


Figura B.9.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 1,332 ms

Erro Médio Quadrático: 0,28057

Relação entre os tempos de processamento: 165,17 vezes

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

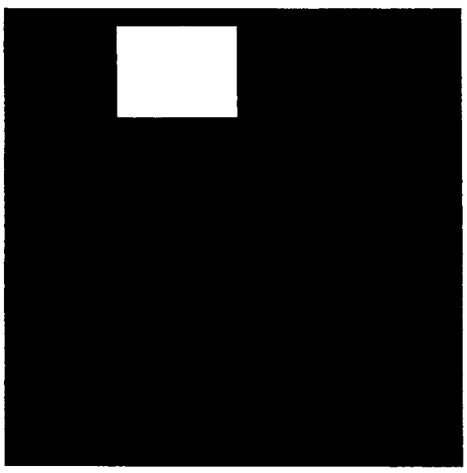


Figura B.10.1. Primeiro quadro de uma seqüência sintética.

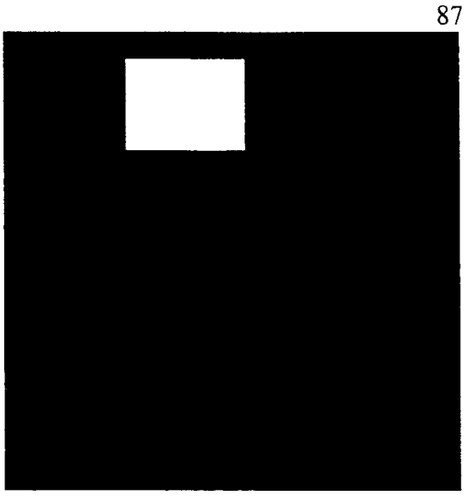


Figura B.10.2. Segundo quadro da seqüência.

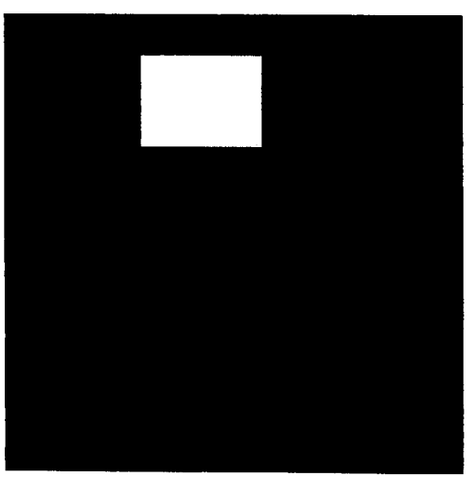


Figura B.10.3. Terceiro quadro da seqüência.

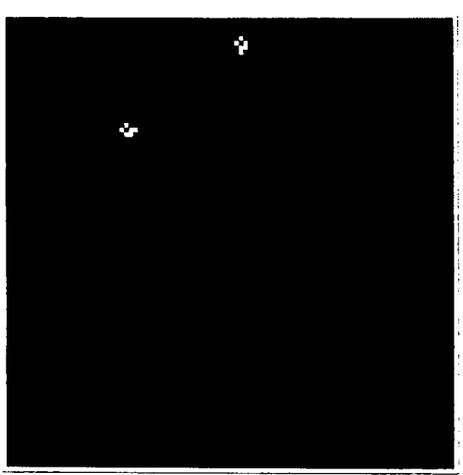


Figura B.10.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

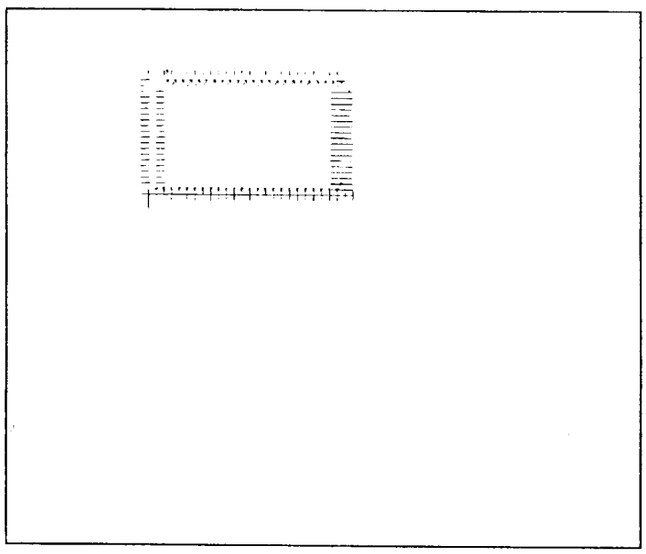


Figura B.10.5. Fluxo ótico da seqüência sintética obtido sem máscara de Cabestaing.

Tempo de processamento: 0,220 s

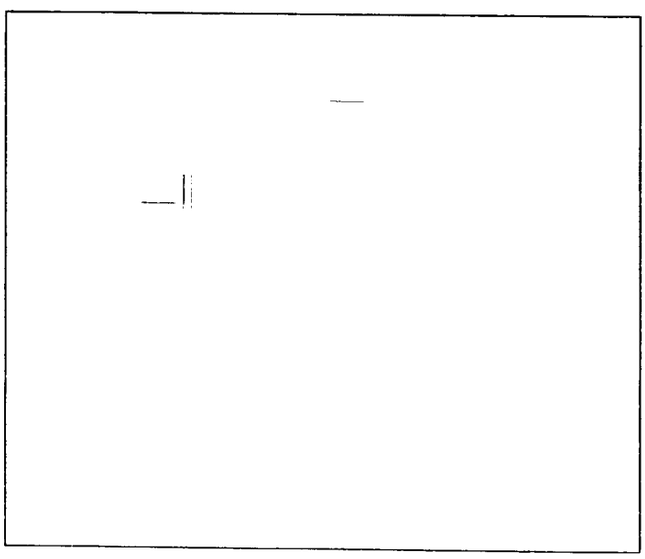


Figura B.10.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 1,254 ms

Erro Médio Quadrático: 0,26808

Relação entre os tempos de processamento: 175,44 vezes

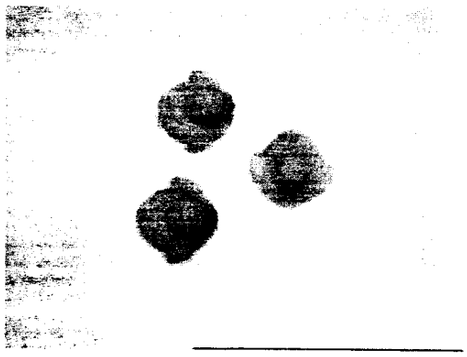


Figura B.11.1. Primeiro quadro de uma seqüência feita com brinquedos (três patos).

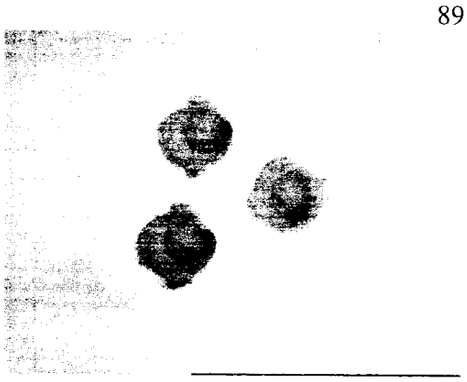


Figura B.11.2. Segundo quadro da seqüência.

Figura B.11.3. Terceiro quadro da seqüência.

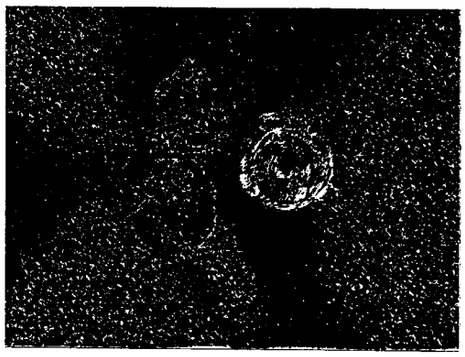


Figura B.11.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

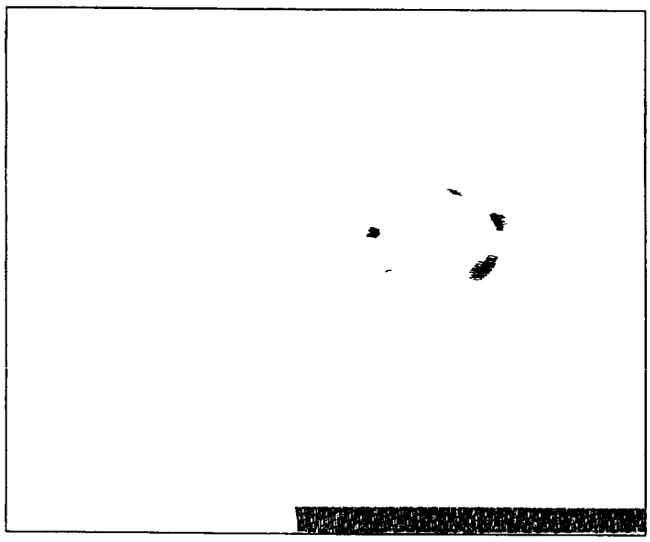


Figura B.11.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,040 s

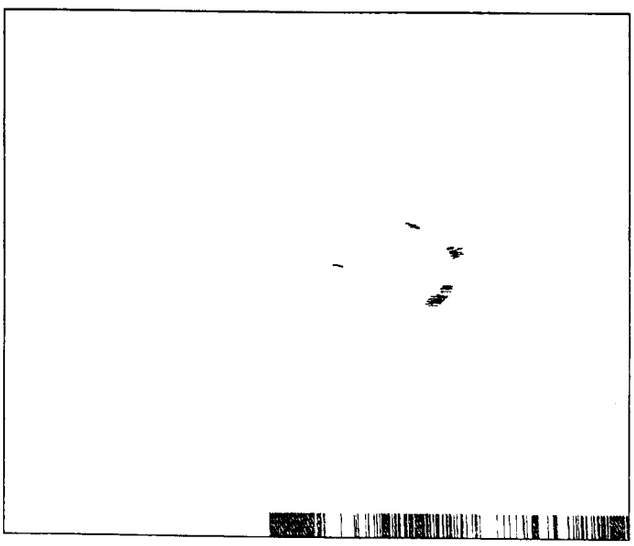


Figura B.11.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,080 s

Erro Médio Quadrático: $9{,}3064.10^{-11}$

Relação entre os tempos de processamento: 75,5 vezes

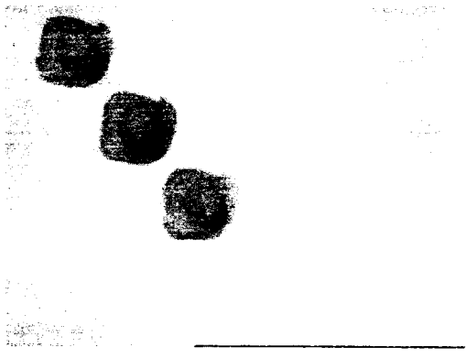


Figura B.12.1. Primeiro quadro de uma seqüência feita com brinquedos (três patos).

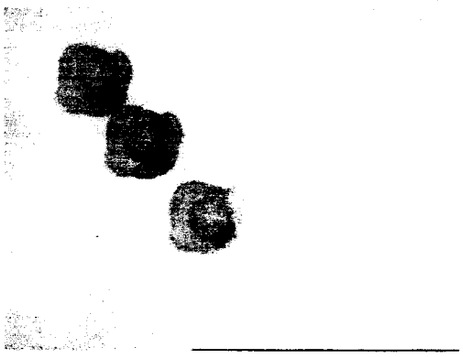


Figura B.12.2. Segundo quadro da seqüência.

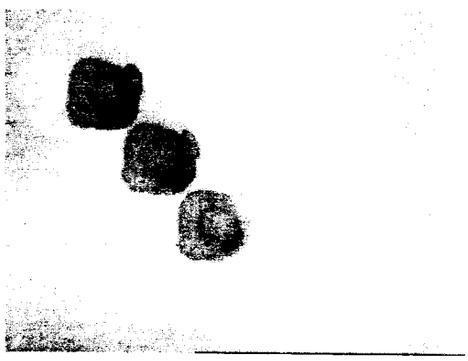


Figura B.12.3. Terceiro quadro da seqüência.

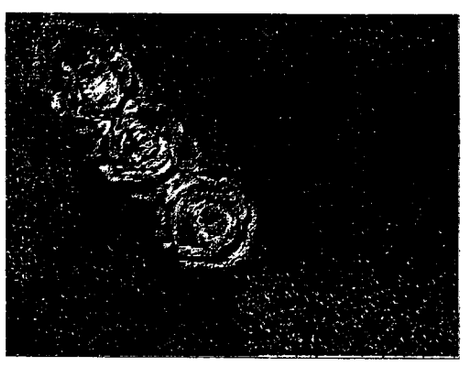


Figura B.12.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

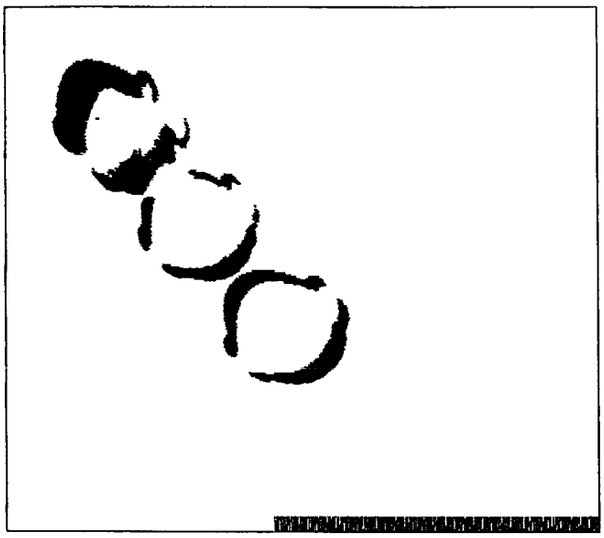


Figura B.12.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,100 s

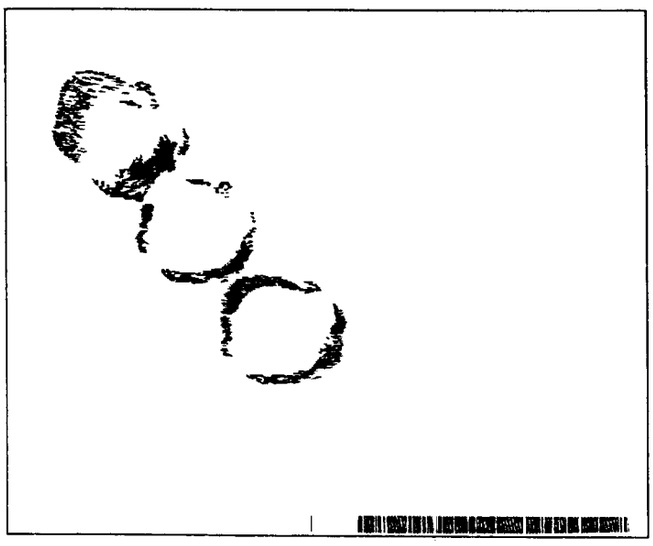


Figura B.12.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,440 s

Erro Médio Quadrático: $2{,}7580.10^{-6}$

Relação entre os tempos de processamento: 13,86 vezes

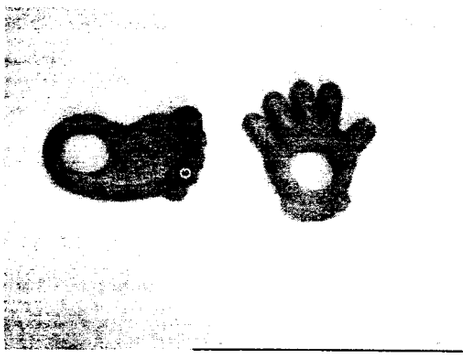


Figura B.13.1. Primeiro quadro de uma seqüência feita com brinquedos (pé e mão).

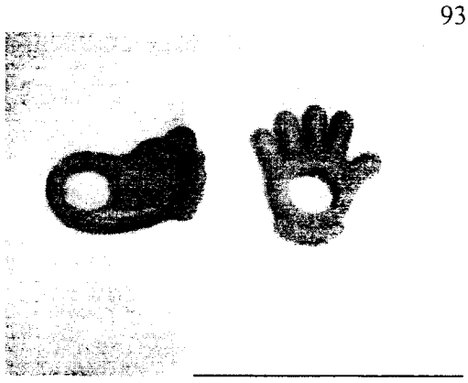


Figura B.13.2. Segundo quadro da seqüência.

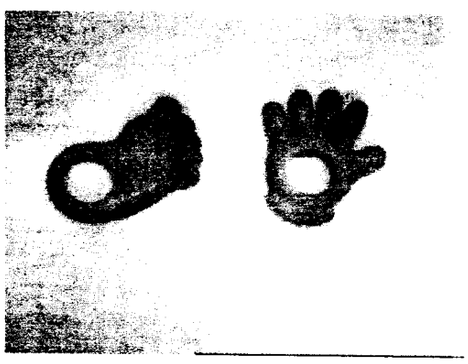


Figura B.13.3. Terceiro quadro da seqüência.

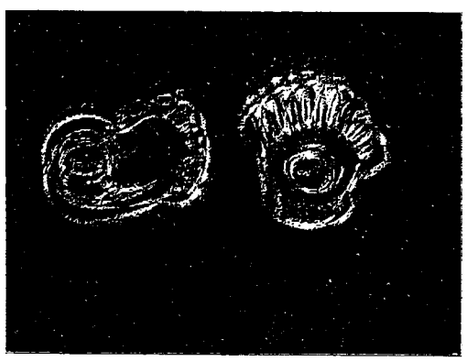


Figura B.13.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

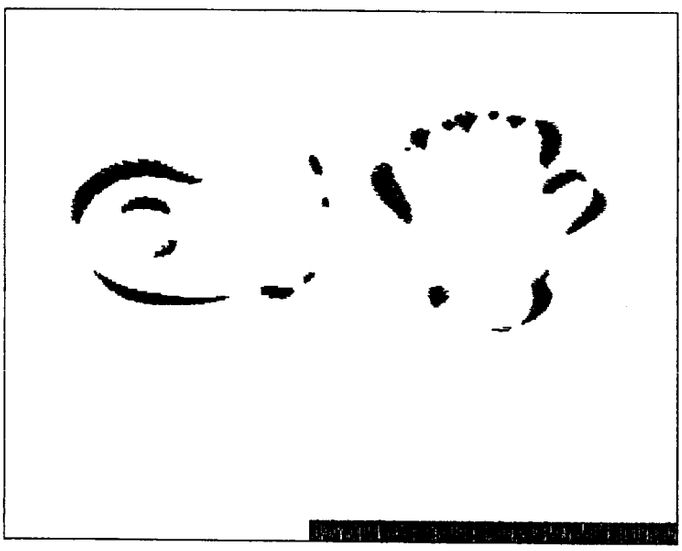


Figura B.13.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,090 s

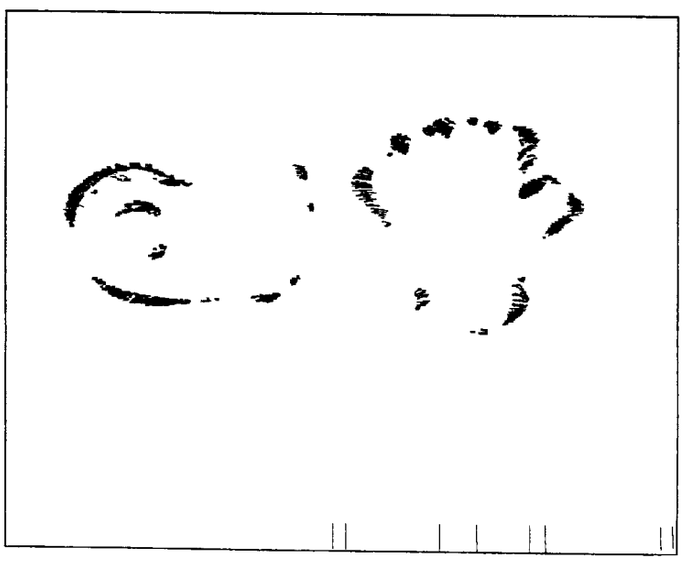


Figura B.13.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,440 s

Erro Médio Quadrático: $1{,}6149.\ 10^{-6}$

Relação entre os tempos de processamento: 13,84 vezes

Figura B.14.1. Primeiro quadro de uma seqüência feita com brinquedos (pé e mão).

Figura B.14.2. Segundo quadro da seqüência.

Figura B.14.3. Terceiro quadro da seqüência.

Figura B.14.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

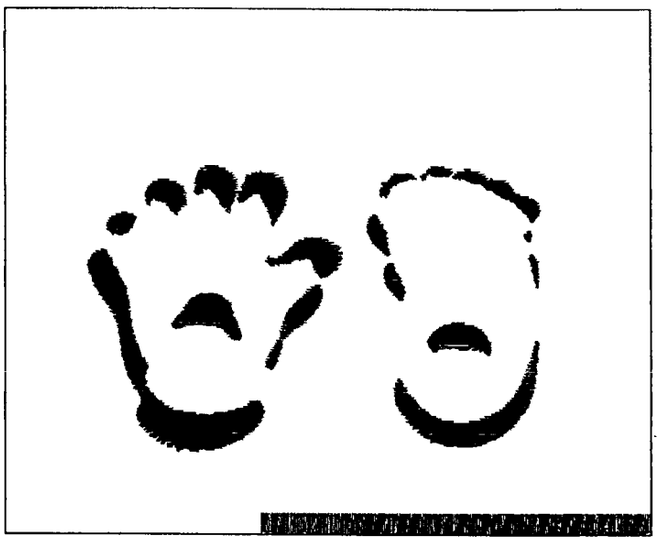


Figura B.14.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,050 s

Figura B.14.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,720 s

Erro Médio Quadrático: $1{,}2706.10^{-5}$

Relação entre os tempos de processamento: 8,40 vezes

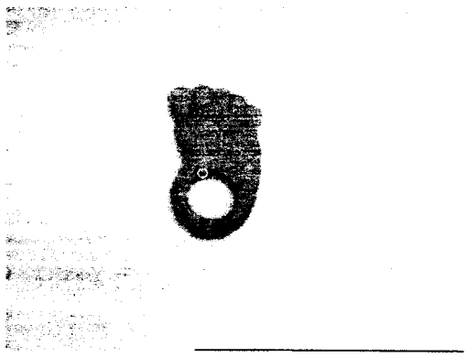


Figura B.15.1. Primeiro quadro de uma seqüência feita com brinquedos (pé).

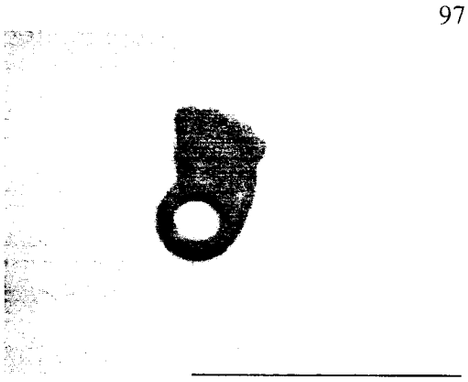


Figura B.15.2. Segundo quadro da seqüência.

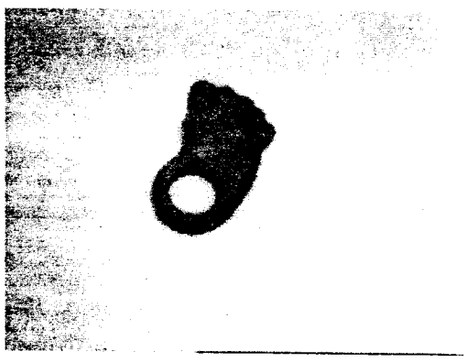


Figura B.15.3. Terceiro quadro da seqüência.

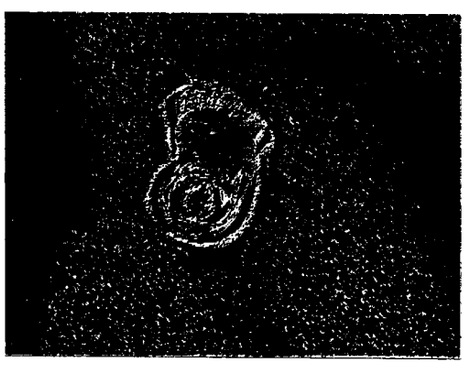


Figura B.15.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

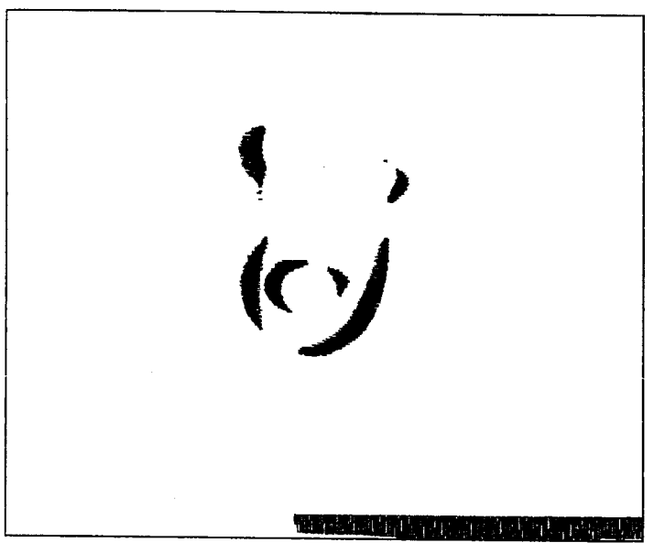


Figura B.15.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,320 s

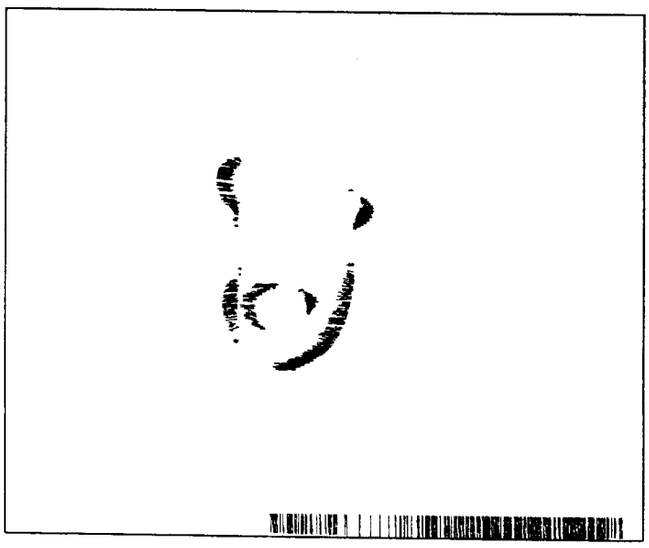


Figura B.15.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,550 s

Erro Médio Quadrático: $9{,}6684.10^{-7}$

Relação entre os tempos de processamento: 11,50 vezes

ICMC-USP SERVIÇO DE BIBLIOTECA E INFORMAÇÃO

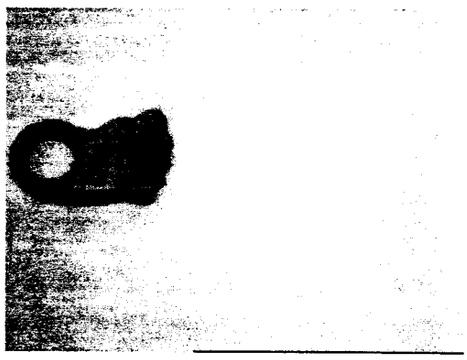


Figura B.16.1. Primeiro quadro de uma seqüência feita com brinquedos (pé).

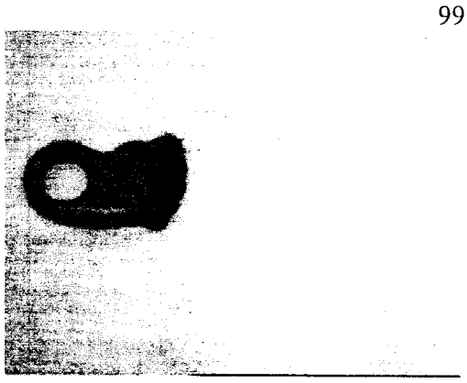


Figura B.16.2. Segundo quadro da seqüência.

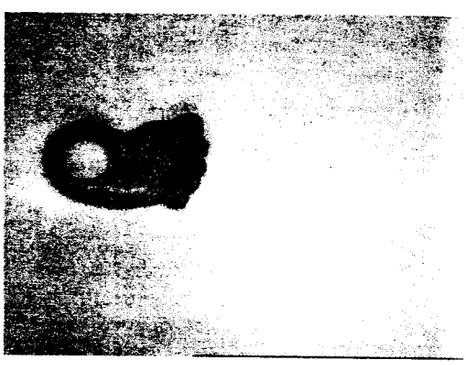


Figura B.16.3. Terceiro quadro da seqüência.

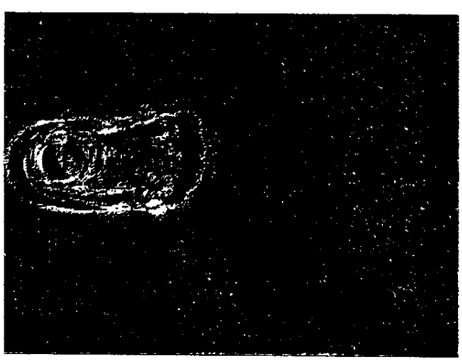


Figura B.16.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

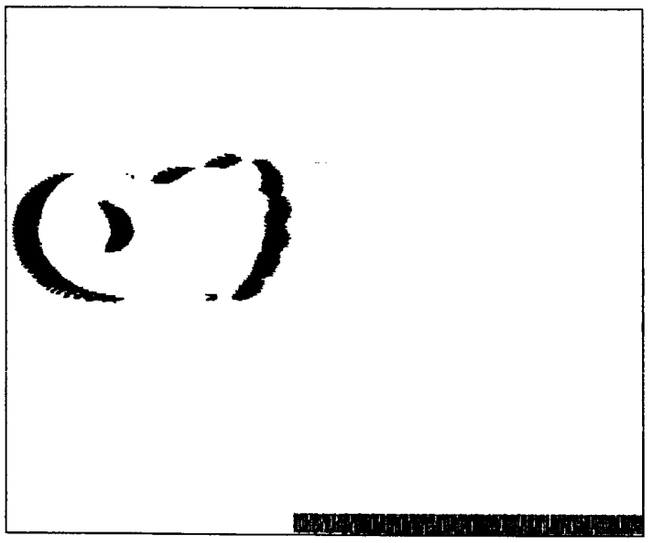


Figura B.16.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,040 s

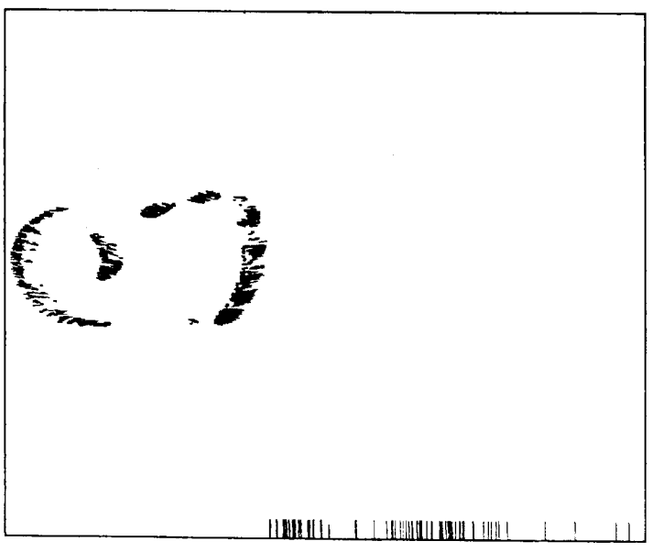


Figura B.16.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,330 s

Erro Médio Quadrático: $8{,}4780.10^{-7}$

Relação entre os tempos de processamento: 18,30 vezes

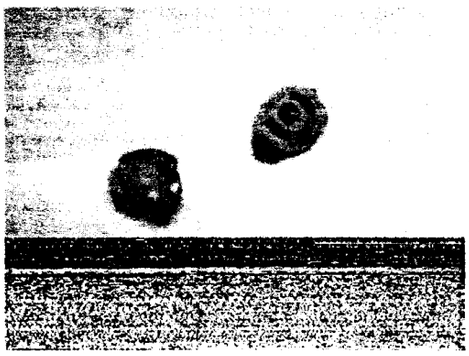


Figura B.17.1. Primeiro quadro de uma seqüência feita com brinquedos (um sapo e uma tartaruga).

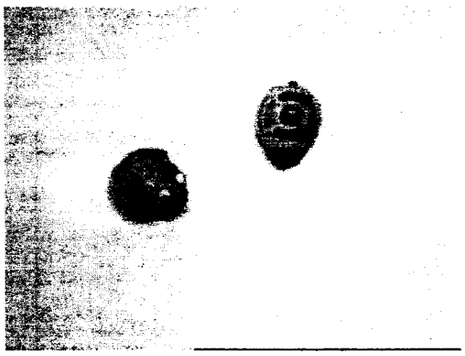


Figura B.17.2. Segundo quadro da seqüência.

Figura B.17.3. Terceiro quadro da seqüência.

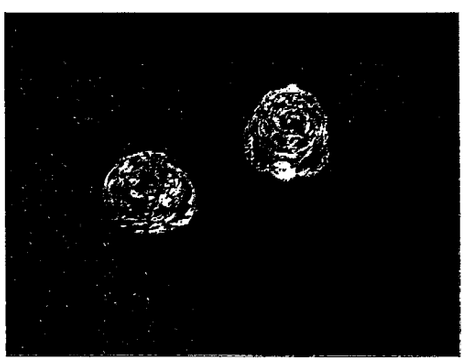


Figura B.17.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

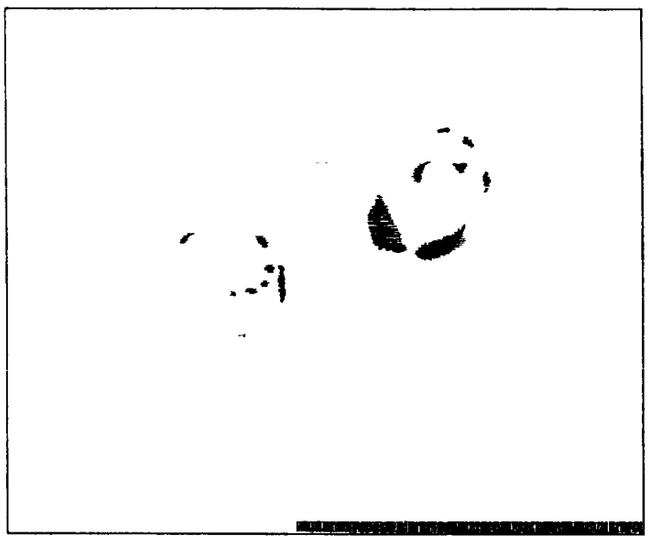


Figura B.17.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,040 s

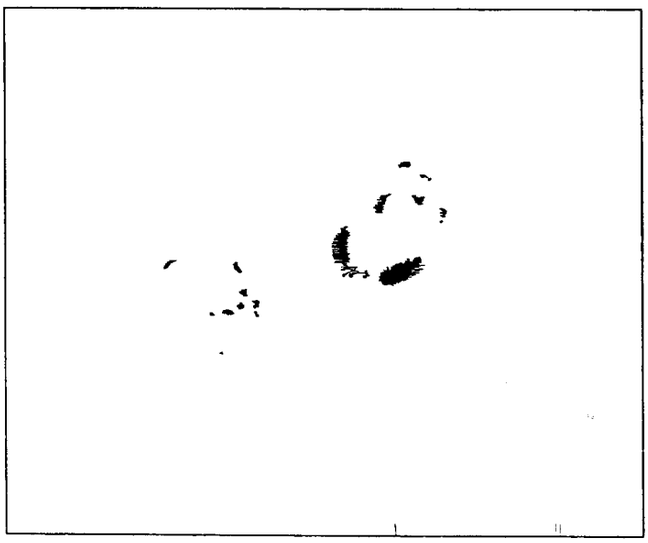


Figura B.17.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,270 s

Erro Médio Quadrático: 0,00012

Relação entre os tempos de processamento: 22,37 vezes

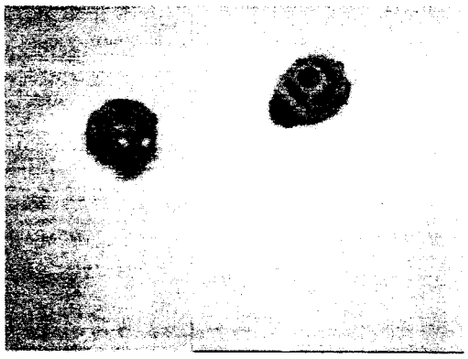


Figura B.18.1. Primeiro quadro de uma seqüência feita com brinquedos (um sapo e uma tartaruga).

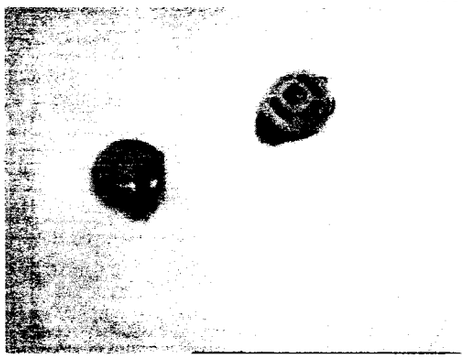


Figura B.18.2. Segundo quadro da seqüência.

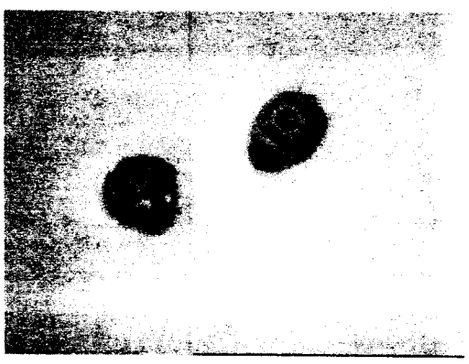


Figura B.18.3. Terceiro quadro da seqüência.

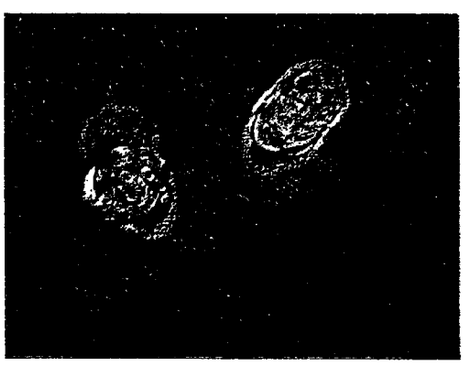


Figura B.18.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

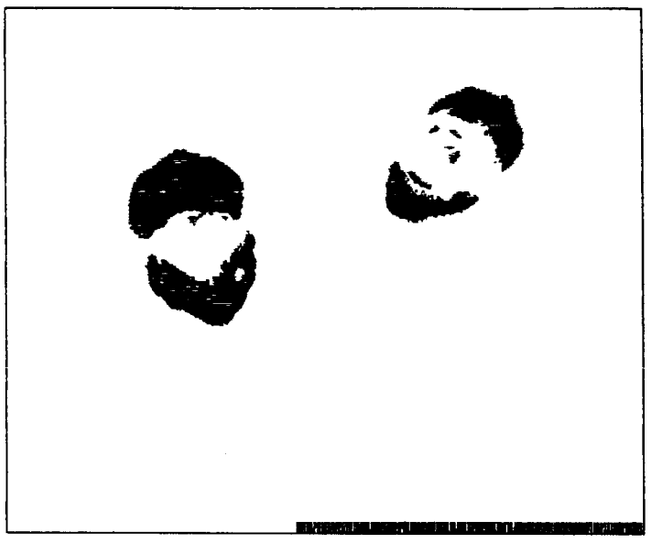


Figura B.18.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,100 s

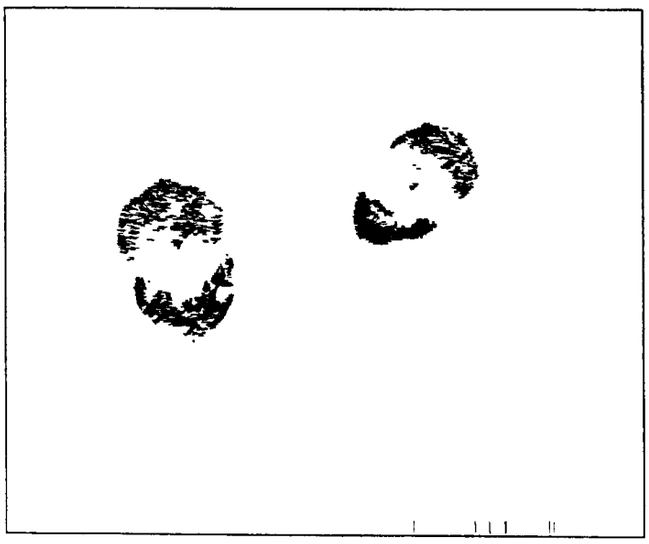


Figura B.18.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,330

Erro Médio Quadrático: 0,00071

Relação entre os tempos de processamento: 18,48 vezes

Figura B.19.1. Primeiro quadro de uma seqüência feita com brinquedos (um relógio, um pato e uma tartaruga).

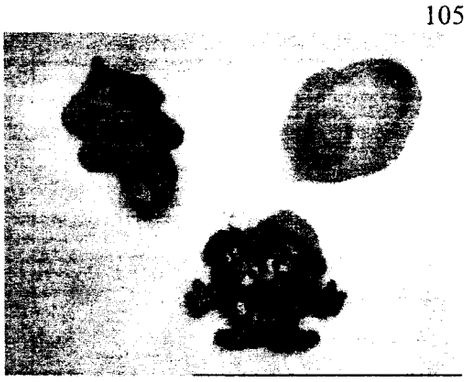


Figura B.19.2. Segundo quadro da seqüência.

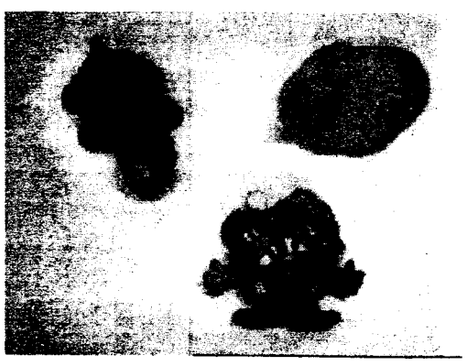


Figura B.19.3. Terceiro quadro da seqüência.

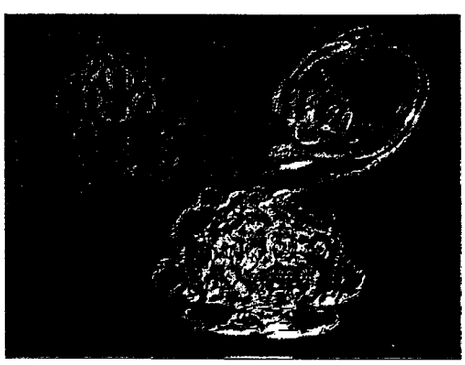


Figura B.19.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

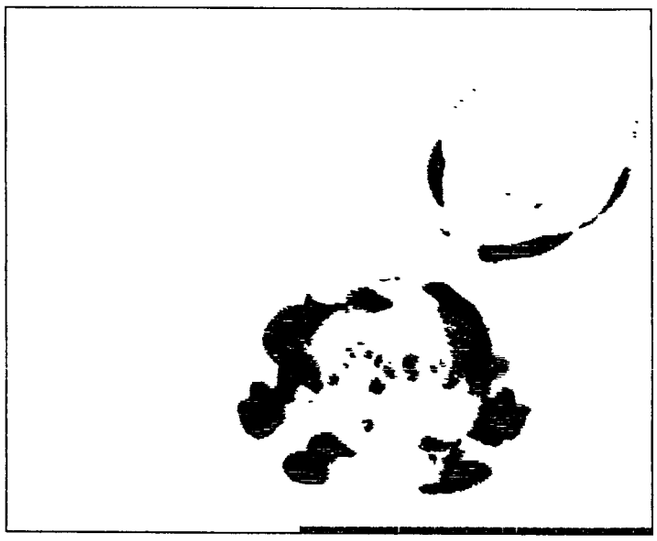


Figura B.19.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 0,490 s

Figura B.19.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,060 s

Erro Médio Quadrático: 0,00178

Relação entre os tempos de processamento: 8,17 vezes

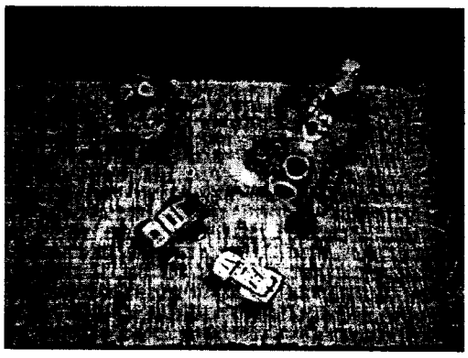


Figura B.20.1. Primeiro quadro de uma seqüência feita com brinquedos (um palhaço, um boneco e dois carrinhos).

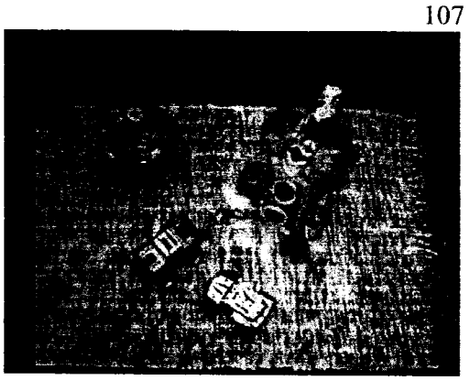


Figura B.20.2. Segundo quadro da seqüência.

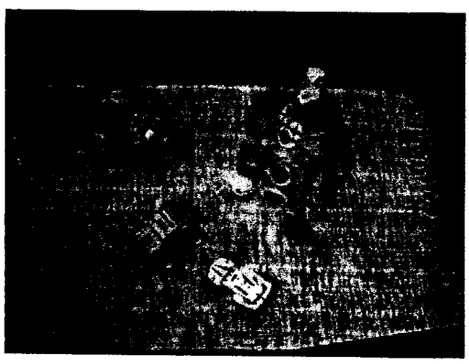


Figura B.20.3. Terceiro quadro da seqüência.

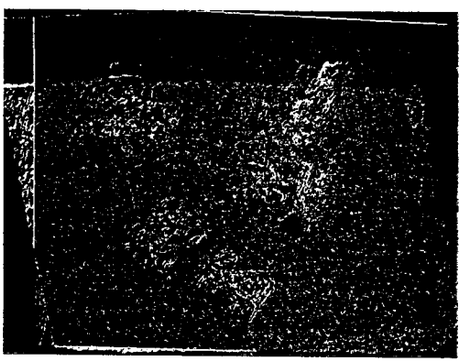


Figura B.20.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

IFSC-USP SERVIÇO DE BIBLIOTECA E INFORMAÇÃO

Figura B.20.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,090 s

Figura B.20.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 1,260 s

Erro Médio Quadrático: 0,00056

Relação entre os tempos de processamento: 8,83 vezes

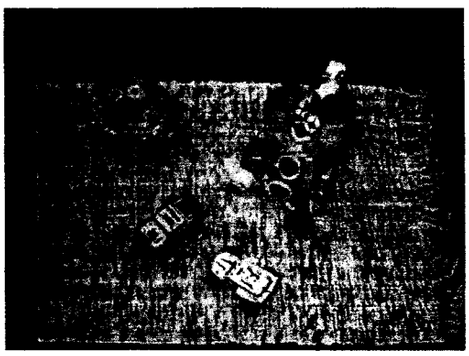


Figura B.21.1. Primeiro quadro de uma seqüência feita com brinquedos (um palhaço, um boneco e dois carrinhos).

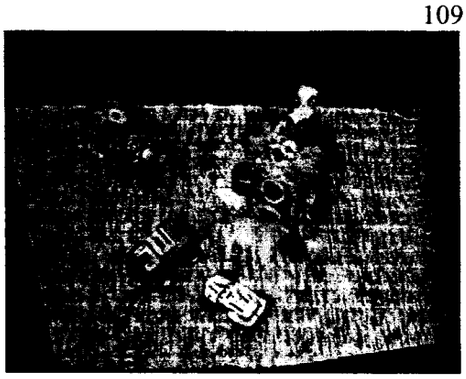


Figura B.21.2. Segundo quadro da seqüência.

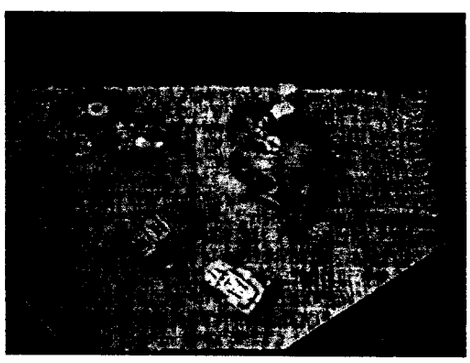


Figura B.21.3. Terceiro quadro da seqüência.

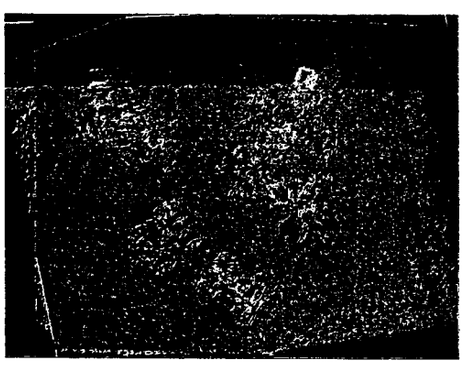


Figura B.21.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

Figura B.21.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,050 s

Figura B.21.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,850 s

Erro Médio Quadrático: 0,00788

Relação entre os tempos de processamento: 7,12 vezes

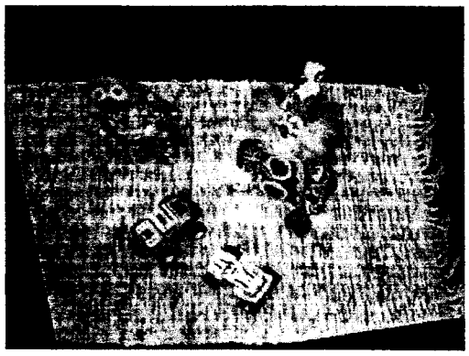


Figura B.22.1. Primeiro quadro de uma seqüência feita com brinquedos (um palhaço, um boneco e dois carrinhos).

Figura B.22.2. Segundo quadro da seqüência.

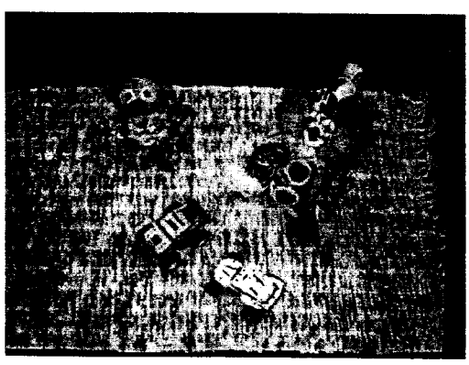


Figura B.22.3. Terceiro quadro da seqüência.

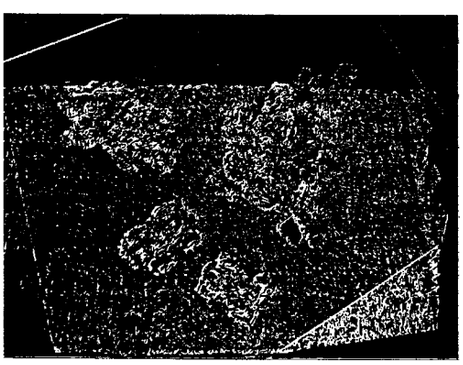


Figura B.22.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

Figura B.22.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 2,900 s

Figura B.22.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 1,090 s

Erro Médio Quadrático: 0,02263

Relação entre os tempos de processamento: 2,66 vezes

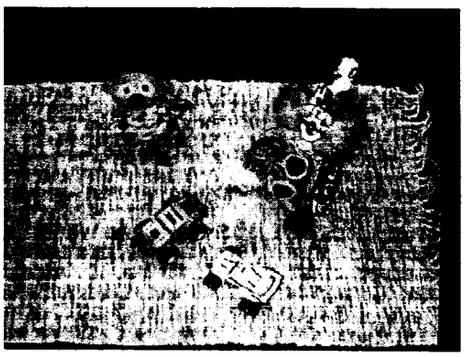


Figura B.23.1. Primeiro quadro de uma seqüência feita com brinquedos (um palhaço, um boneco e dois carrinhos).

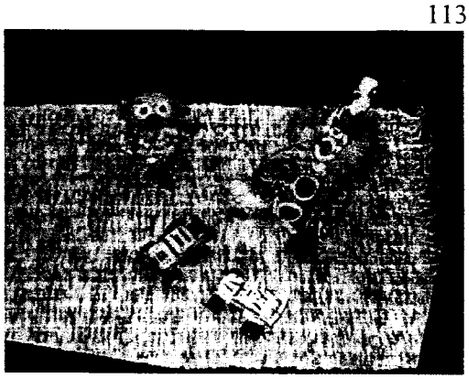


Figura B.23.2. Segundo quadro da seqüência.

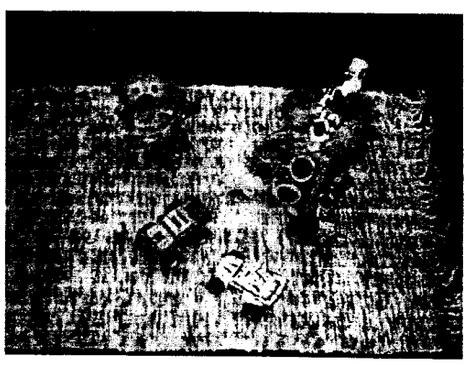


Figura B.23.3. Terceiro quadro da seqüência.

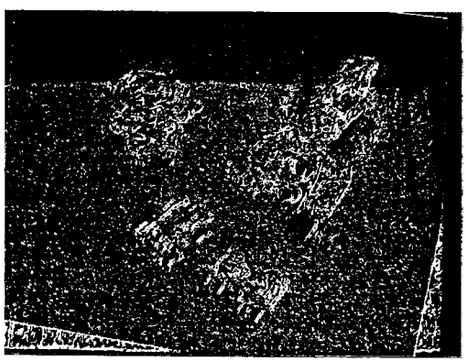


Figura B.23.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

Figura B.23.5. Fluxo ótico da seqüência de brinquedos obtido sem máscara de Cabestaing.

Tempo de processamento: 6,100 s

Figura B.23.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 1,370 s

Erro Médio Quadrático: 0,00088

Relação entre os tempos de processamento: 4,45 vezes

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

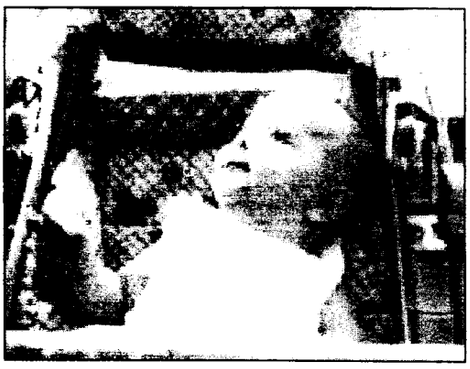


Figura B.24.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.24.2. Segundo quadro da seqüência.

Figura B.24.3. Terceiro quadro da seqüência.

Figura B.20.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

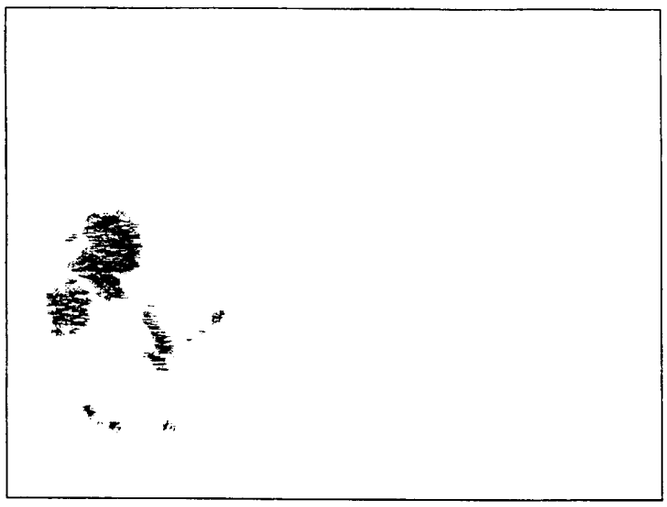


Figura B.24.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,050s

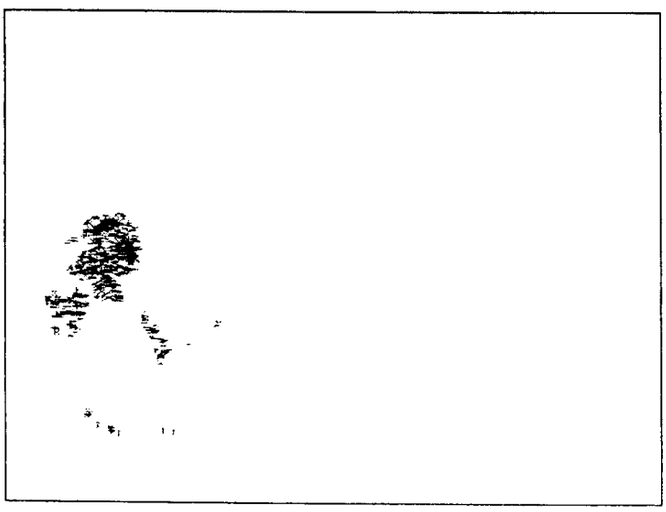


Figura B.24.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,220 s

Erro Médio Quadrático: 0,00194

Relação entre os tempos de processamento: 4,77 vezes

Figura B.25.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.25.2. Segundo quadro da seqüência.

Figura B.25.3. Terceiro quadro da seqüência.

Figura B.25.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

Figura B.25.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,040s

Figura B.25.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,280 s

Erro Médio Quadrático: 0,00194

Relação entre os tempos de processamento: 3,71 vezes

Figura B.26.1. Primeiro quadro de uma seqüência feita com imagens reais.

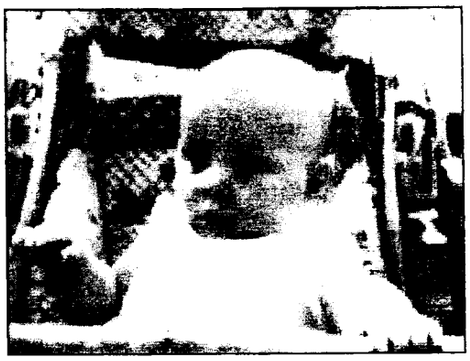


Figura B.26.2. Segundo quadro da seqüência.

Figura B.26.3. Terceiro quadro da seqüência.

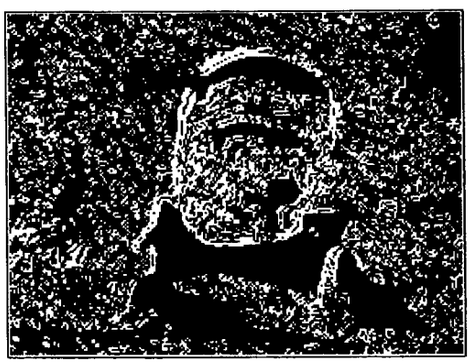


Figura B.26.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

IFSC-USP SERVIÇO DE BI... INFORMAÇÃO

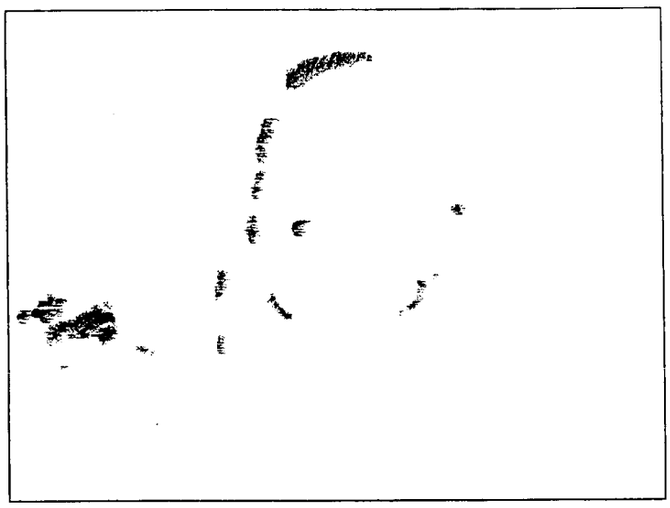


Figura B.26.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,040s

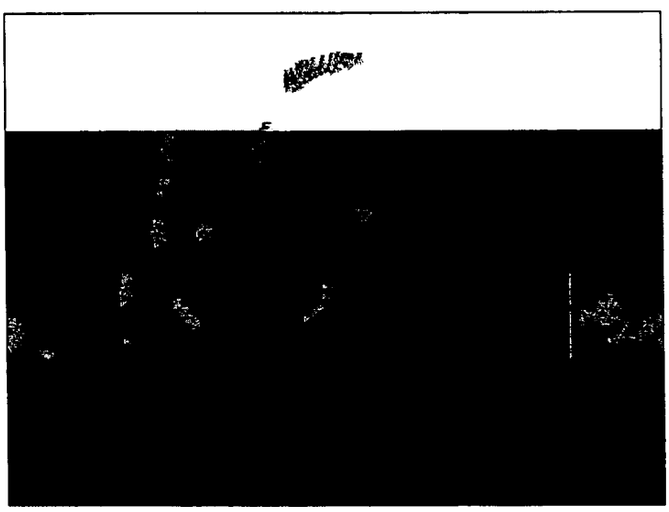


Figura B.26.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,280 s

Erro Médio Quadrático: 0,03371

Relação entre os tempos de processamento: 3,71 vezes

Figura B.27.1. Primeiro quadro de uma seqüência feita com imagens reais.

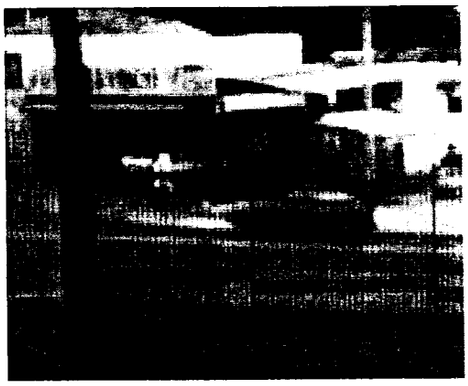


Figura B.27.2. Segundo quadro da seqüência.

Figura B.27.3. Terceiro quadro da seqüência.

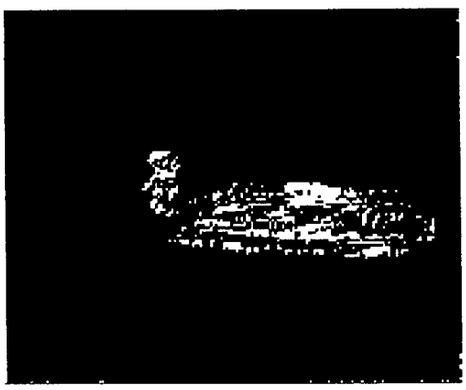


Figura B.27.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

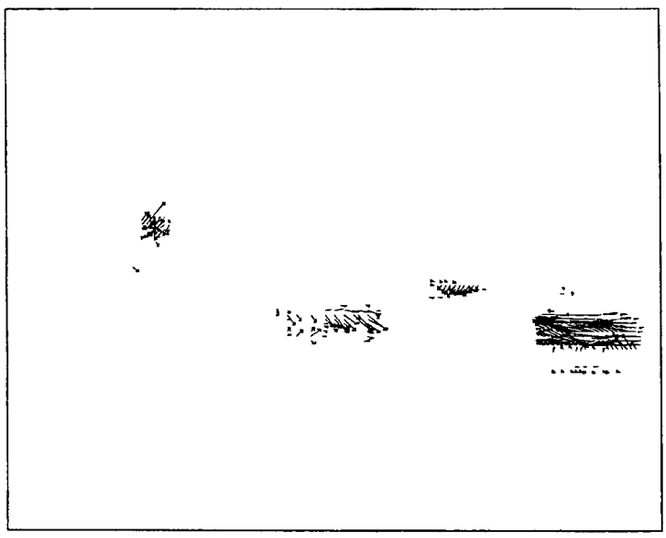


Figura B.27.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 0,490 s

Figura B.27.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,060 s

Erro Médio Quadrático: 0,00559

Relação entre os tempos de processamento: 8,17 vezes

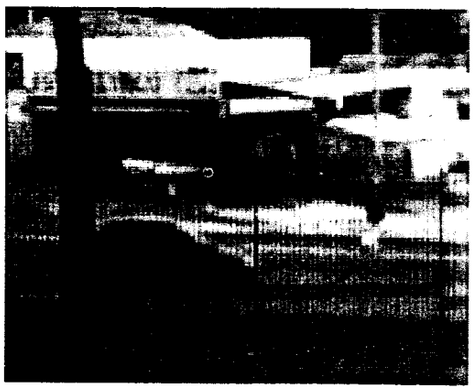


Figura B.28.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.28.2. Segundo quadro da seqüência.

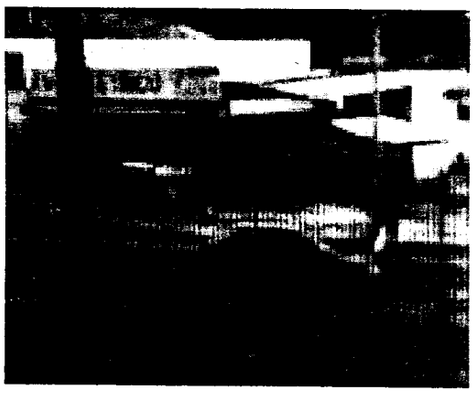


Figura B.28.3. Terceiro quadro da seqüência.

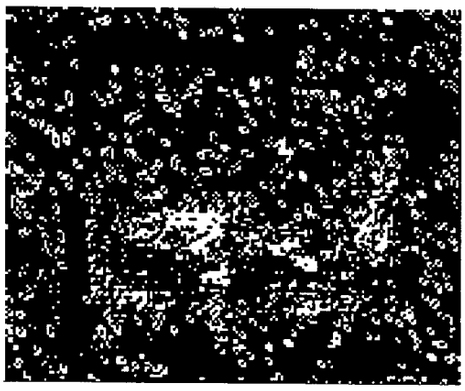


Figura B.28.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

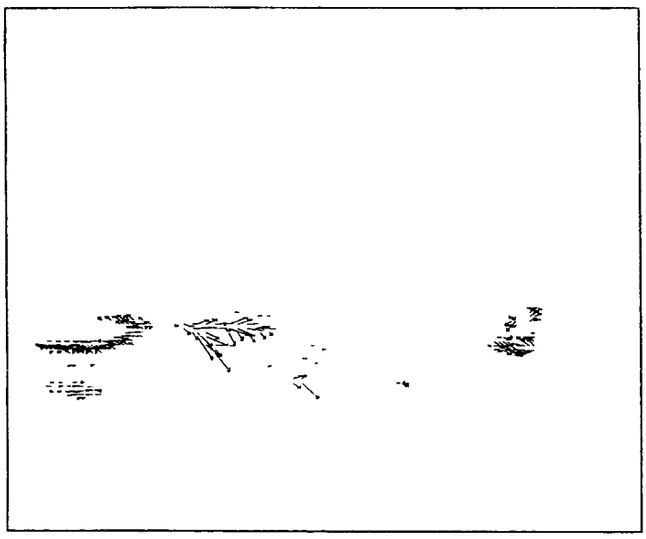


Figura B.28.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 0,490 s

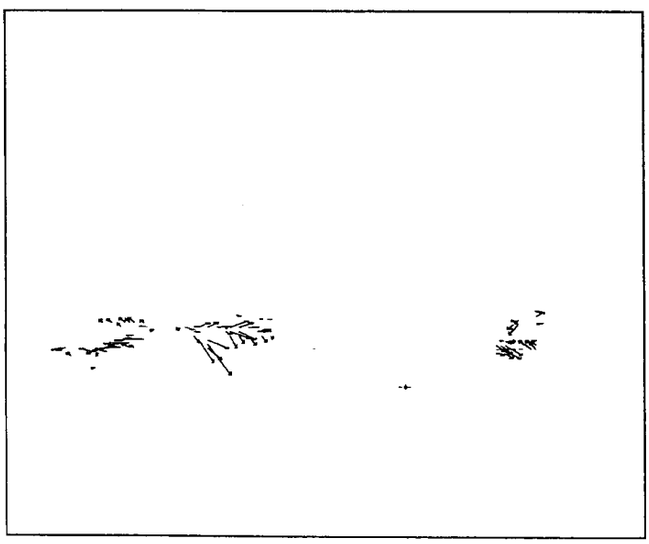


Figura B.28.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,050 s

Erro Médio Quadrático: $4{,}7419.\ 10^{-5}$

Relação entre os tempos de processamento: 9,80 vezes

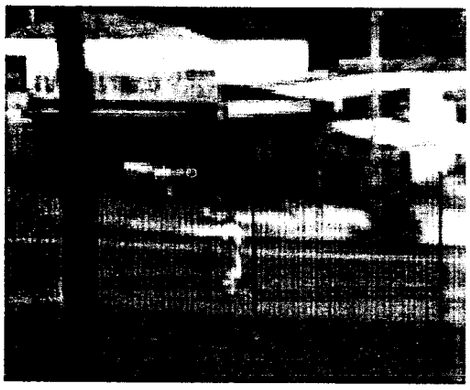


Figura B.29.1. Primeiro quadro de uma seqüência feita com imagens reais.

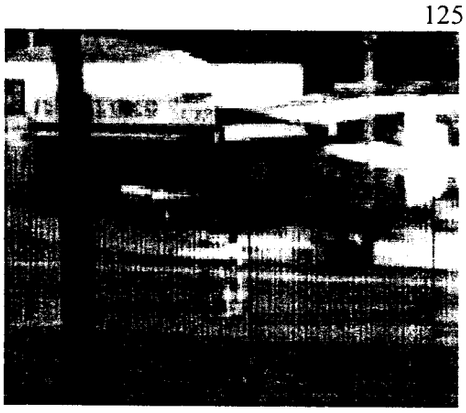


Figura B.29.2. Segundo quadro da seqüência.

Figura B.29.3. Terceiro quadro da seqüência.

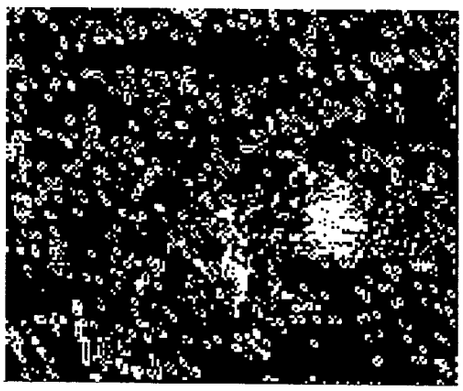


Figura B.29.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

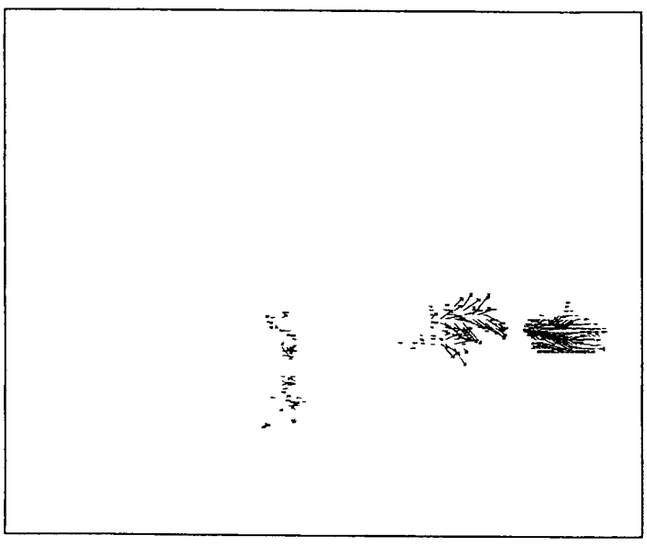


Figura B.29.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 0,490 s

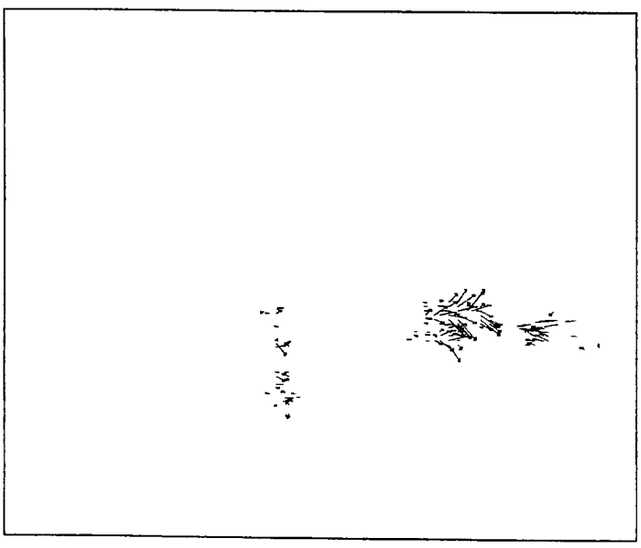


Figura B.29.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,110 s

Erro Médio Quadrático: 0,00232

Relação entre os tempos de processamento: 4,45 vezes

Figura B.30.1. Primeiro quadro de uma seqüência feita com imagens reais.

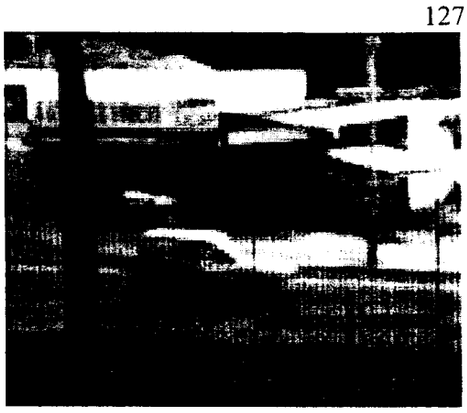


Figura B.30.2. Segundo quadro da seqüência.

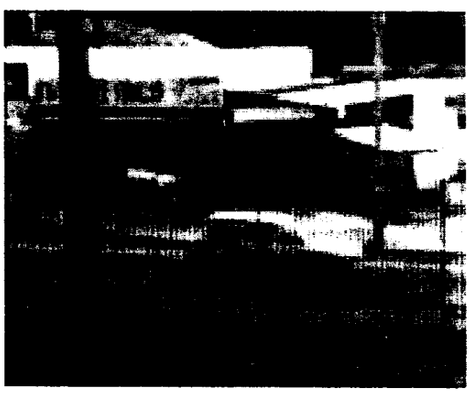


Figura B.30.3. Terceiro quadro da seqüência.

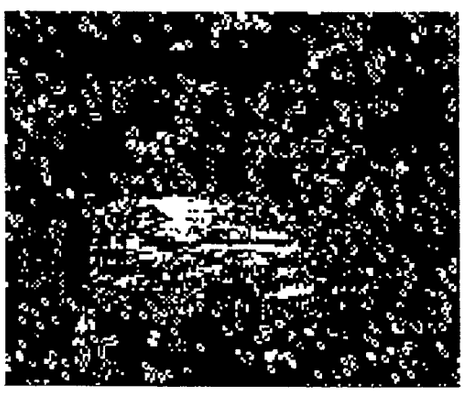


Figura B.30.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

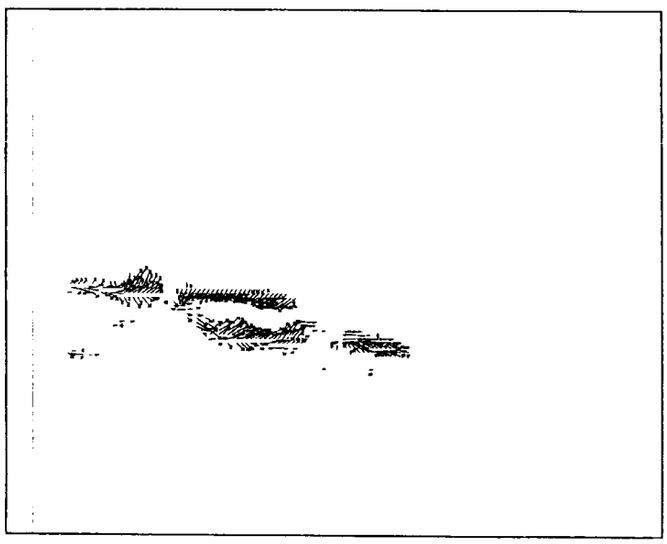


Figura B.30.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,120 s

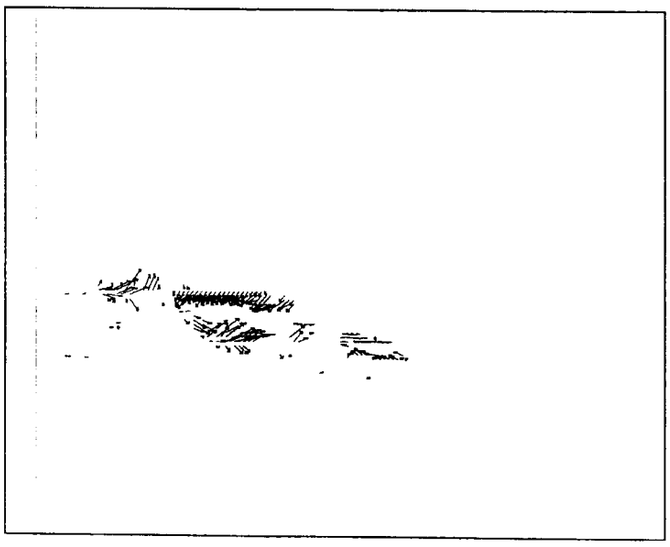


Figura B.30.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,050 s

Erro Médio Quadrático: 0,00133

Relação entre os tempos de processamento: 22,40 vezes

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

Figura B.31.1. Primeiro quadro de uma seqüência feita com imagens reais.

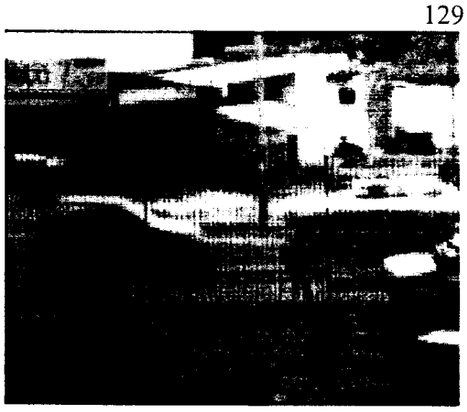


Figura B.31.2. Segundo quadro da seqüência.

Figura B.31.3. Terceiro quadro da seqüência.

Figura B.31.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

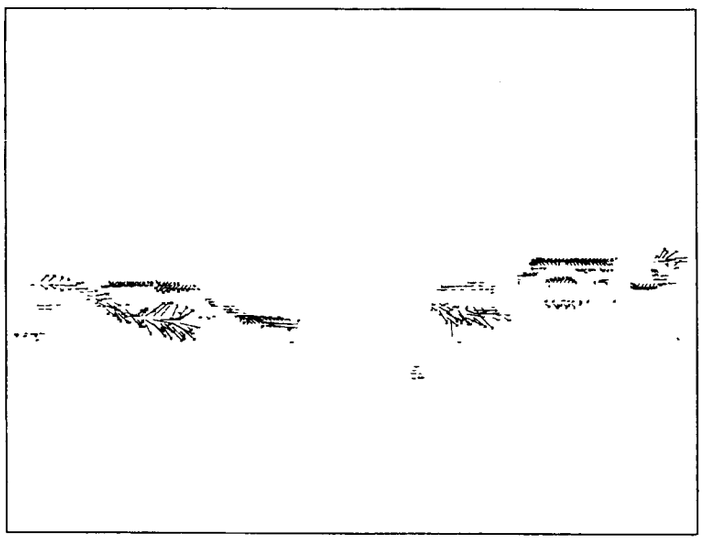


Figura B.31.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 0,500 s

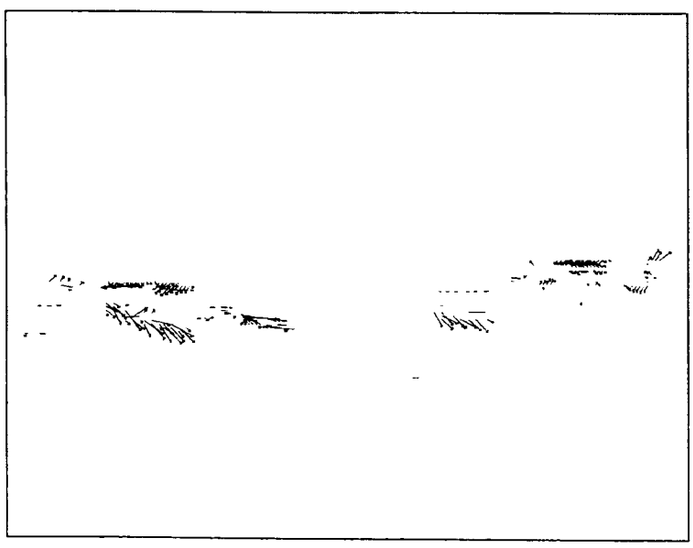


Figura B.31.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,050 s

Erro Médio Quadrático: 0,00332

Relação entre os tempos de processamento: 10,00 vezes

Figura B.32.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.32.2. Segundo quadro da seqüência.

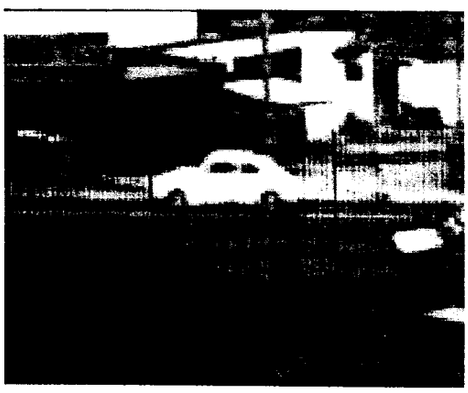


Figura B.32.3. Terceiro quadro da seqüência.

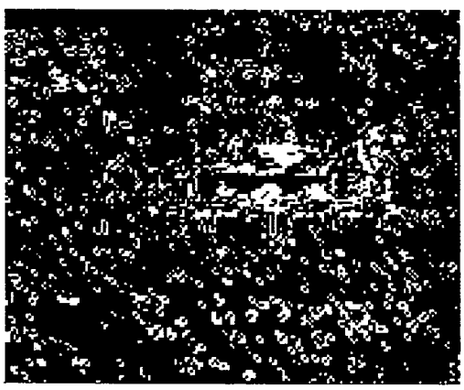


Figura B.32.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

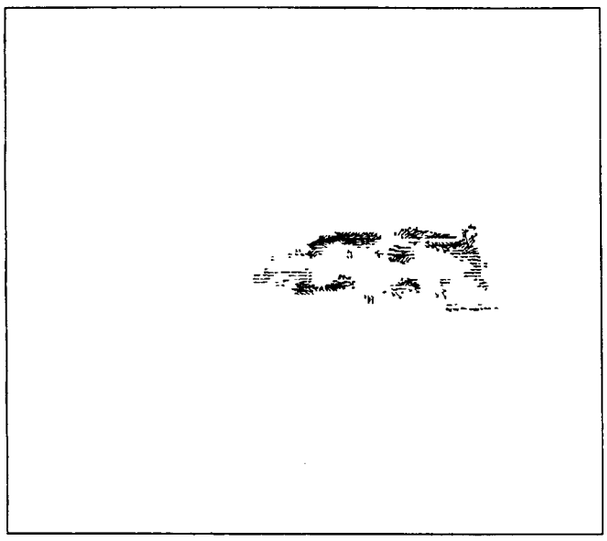


Figura B.32.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 0,490 s

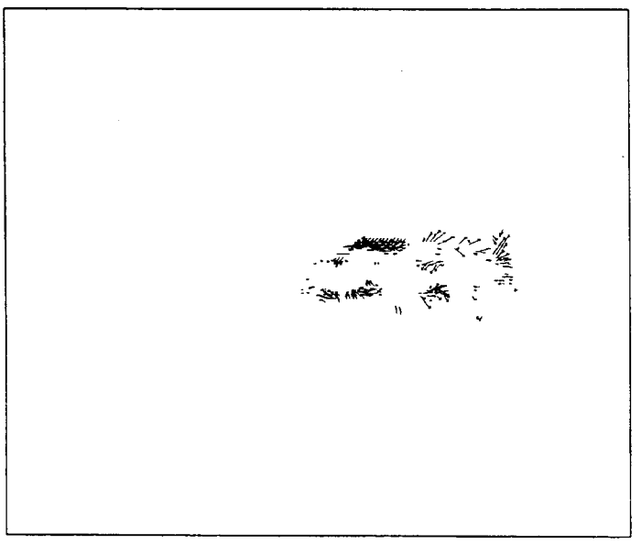


Figura B.32.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,110 s

Erro Médio Quadrático: 0,00166

Relação entre os tempos de processamento: 4,45 vezes

Figura B.33.1. Primeiro quadro de uma seqüência feita com imagens reais.

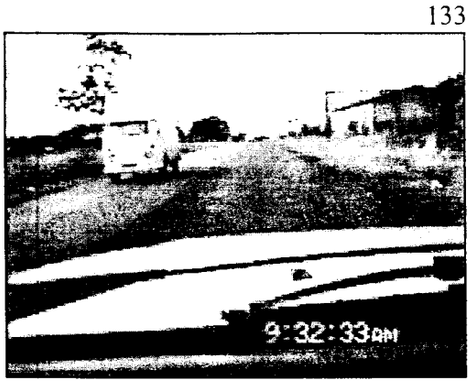


Figura B.33.2. Segundo quadro da seqüência.

Figura B.33.3. Terceiro quadro da seqüência.

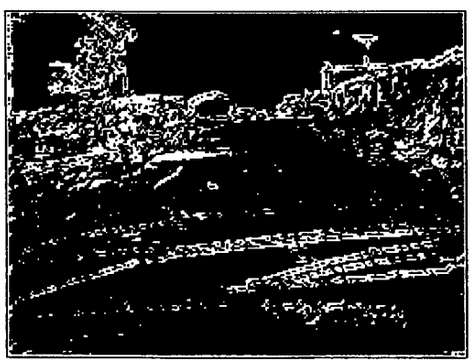


Figura B.33.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

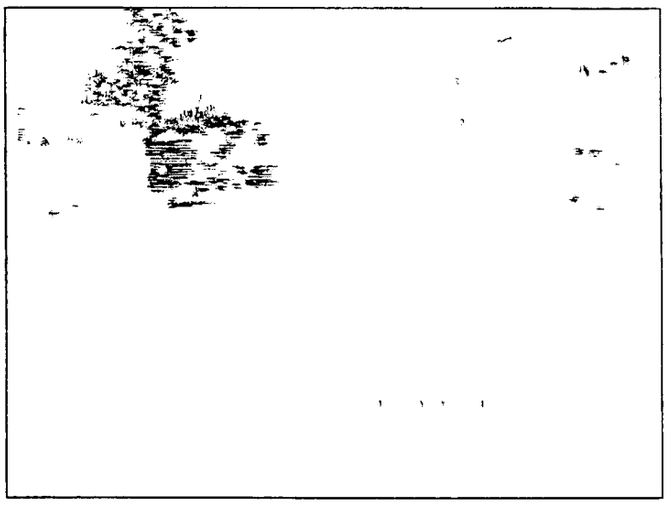


Figura B.33.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,040 s

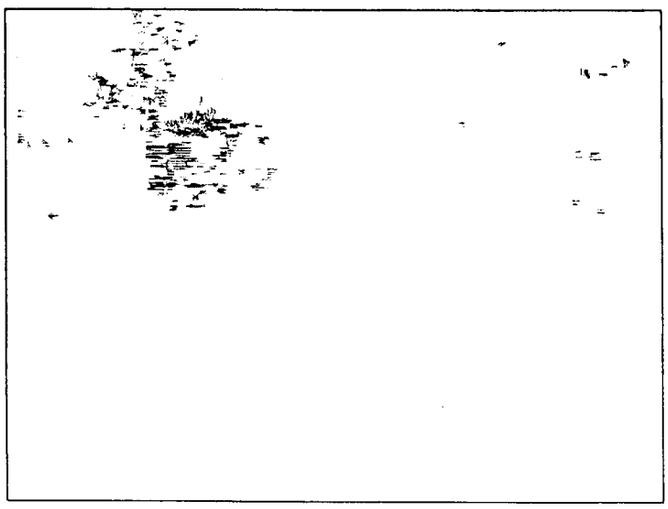


Figura B.33.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,110

Erro Médio Quadrático: 0,00032

Relação entre os tempos de processamento: 9,46 vezes

Figura B.34.1. Primeiro quadro de uma seqüência feita com imagens reais.

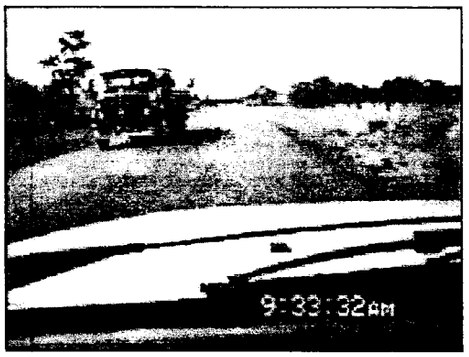


Figura B.34.2. Segundo quadro da seqüência.

Figura B.34.3. Terceiro quadro da seqüência.

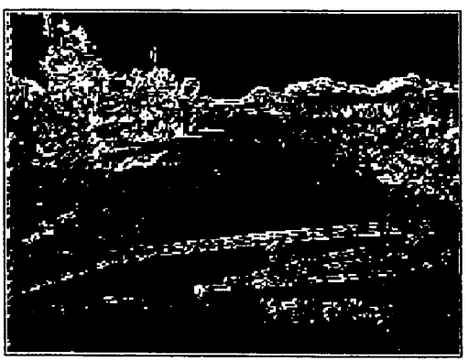


Figura B.34.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

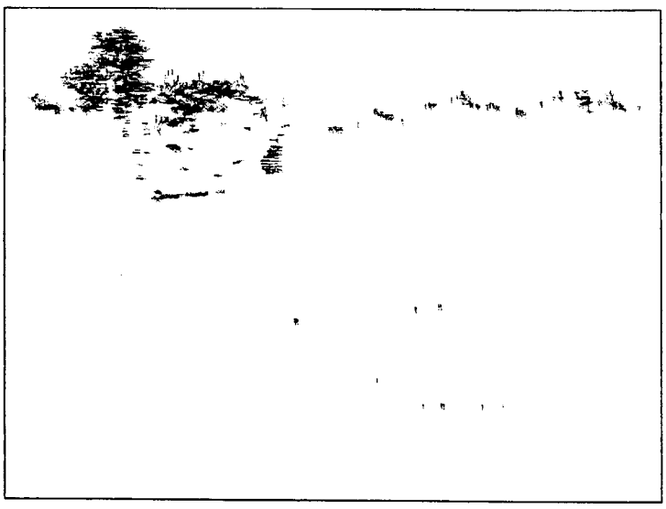


Figura B.34.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,100 s

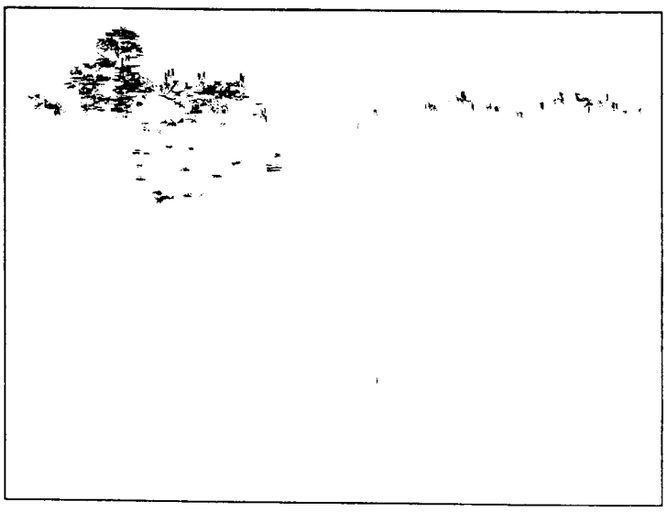


Figura B.34.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,110

Erro Médio Quadrático: 0,03023

Relação entre os tempos de processamento: 10,00 vezes

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

Figura B.35.1. Primeiro quadro de uma seqüência feita com imagens reais.

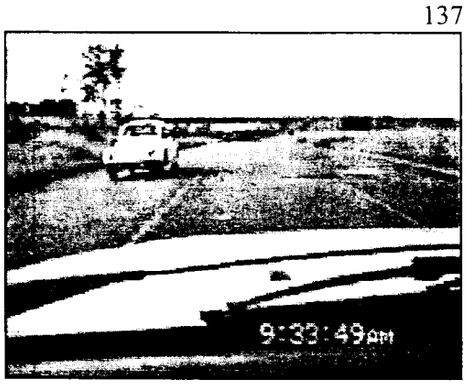


Figura B.35.2. Segundo quadro da seqüência.

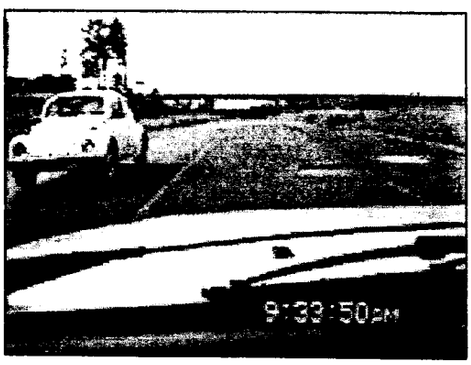


Figura B.35.3. Terceiro quadro da seqüência.

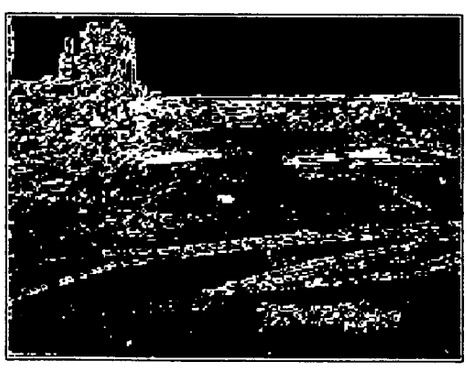


Figura B.35.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

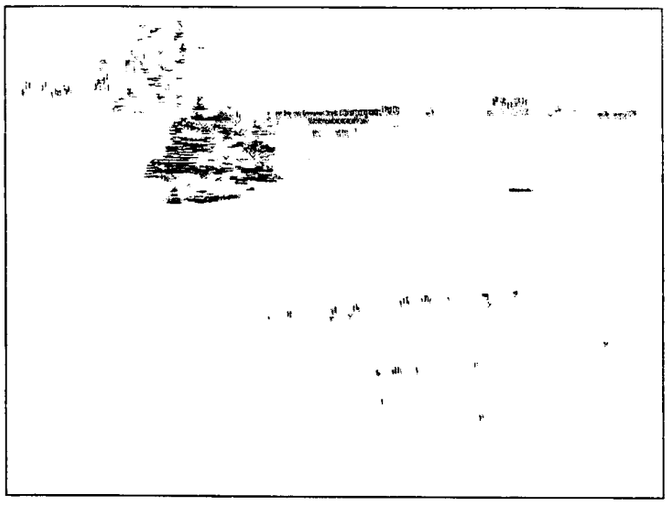


Figura B.35.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,040 s

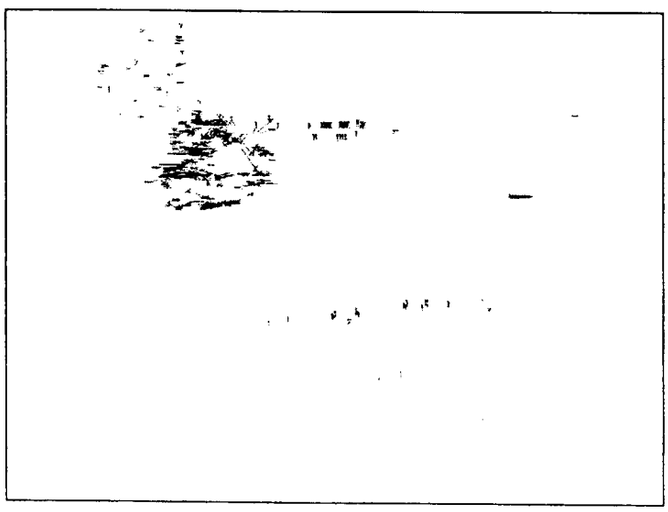


Figura B.35.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,160

Erro Médio Quadrático: 0,00015

Relação entre os tempos de processamento: 6,50 vezes

Figura B.36.1. Primeiro quadro de uma seqüência feita com imagens reais.

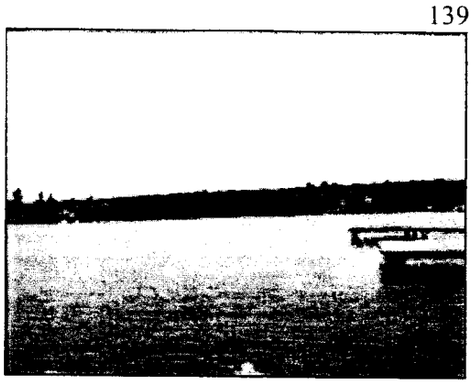


Figura B.36.2. Segundo quadro da seqüência.

Figura B.36.3. Terceiro quadro da seqüência.

Figura B.36.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

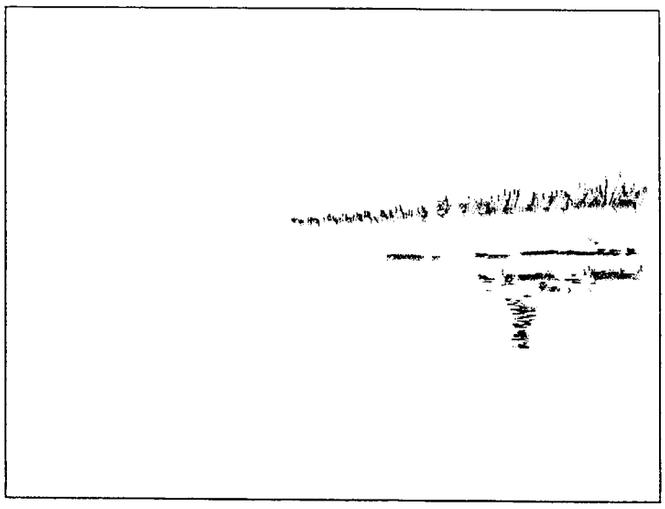


Figura B.36.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,040 s

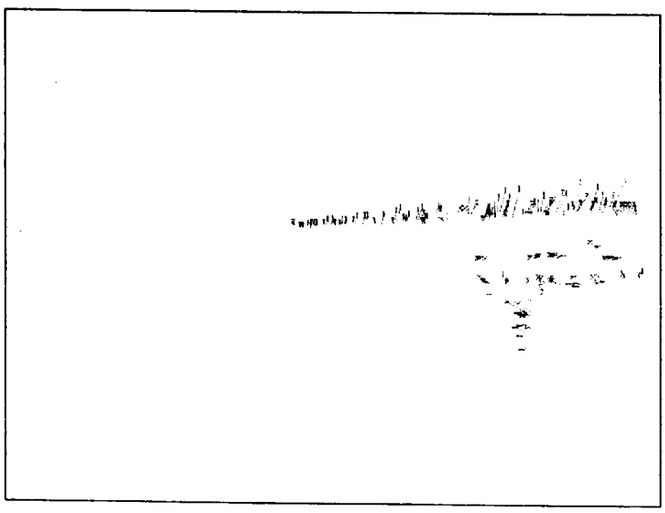


Figura B.36.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,220 s

Erro Médio Quadrático: 0,00012

Relação entre os tempos de processamento: 4,73 vezes

Figura B.37.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.37.2. Segundo quadro da seqüência.

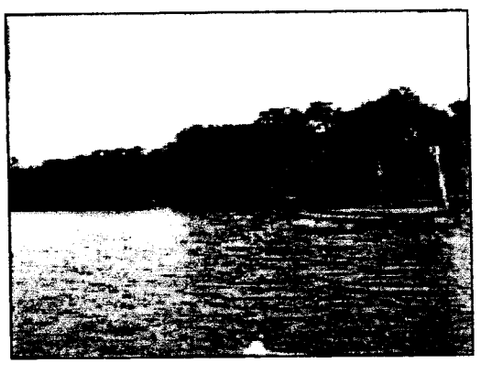


Figura B.37.3. Terceiro quadro da seqüência.

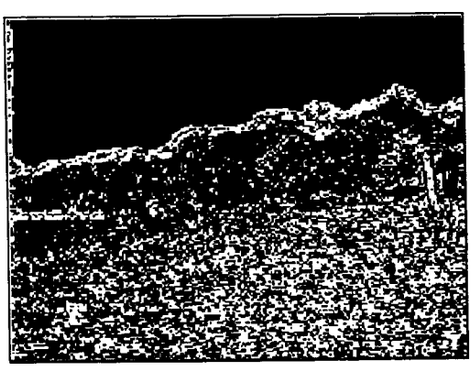


Figura B.37.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

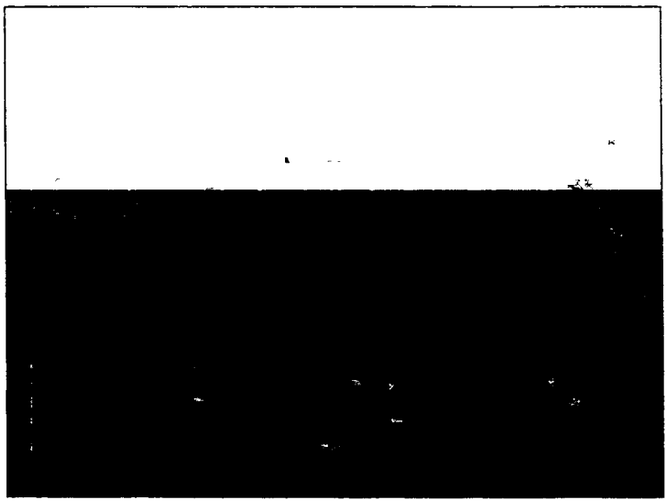


Figura B.37.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,040 s

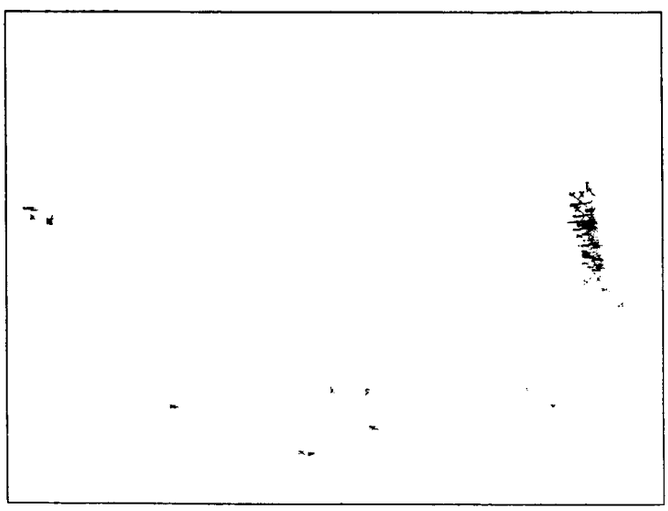


Figura B.37.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,310 s

Erro Médio Quadrático: 1,99300

Relação entre os tempos de processamento: 3,36 vezes

Figura B.38.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.38.2. Segundo quadro da seqüência.

Figura B.38.3. Terceiro quadro da seqüência.

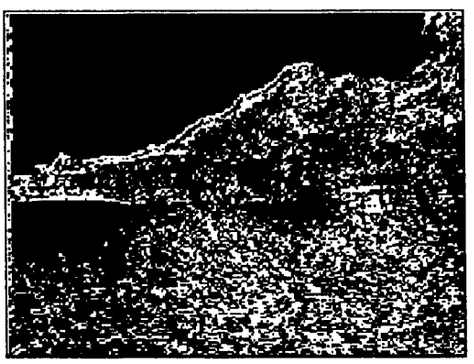


Figura B.38.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

Figura B.38.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,040 s

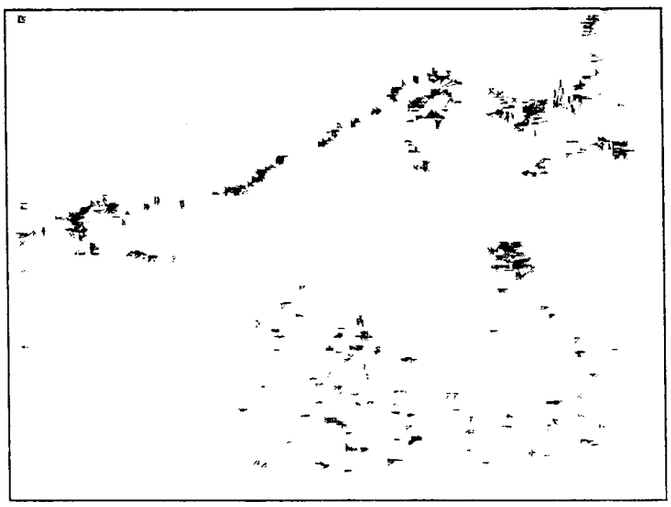


Figura B.38.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,220

Erro Médio Quadrático: 0,02964

Relação entre os tempos de processamento: 4,73 vezes

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

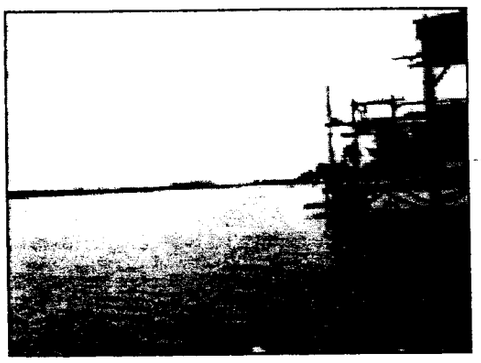


Figura B.39.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.39.2. Segundo quadro da seqüência.

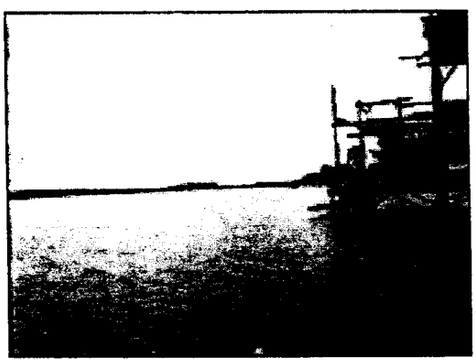


Figura B.39.3. Terceiro quadro da seqüência.

Figura B.39.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

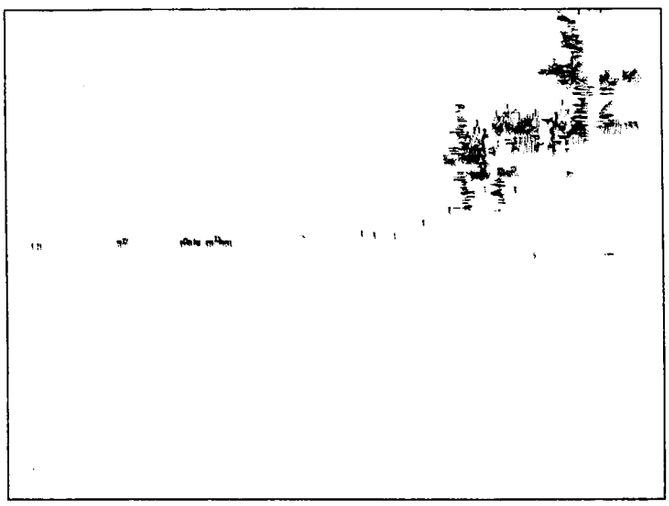


Figura B.39.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,050 s

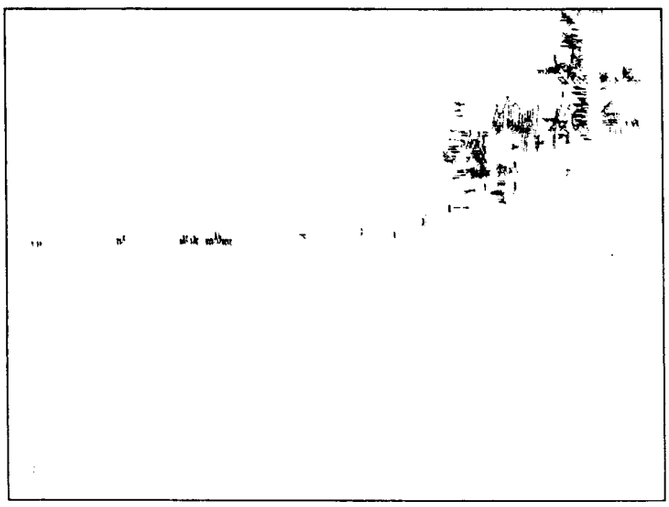


Figura B.39.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,220

Erro Médio Quadrático: 0,00052

Relação entre os tempos de processamento: 4,77 vezes

Figura B.40.1. Primeiro quadro de uma seqüência feita com imagens reais.

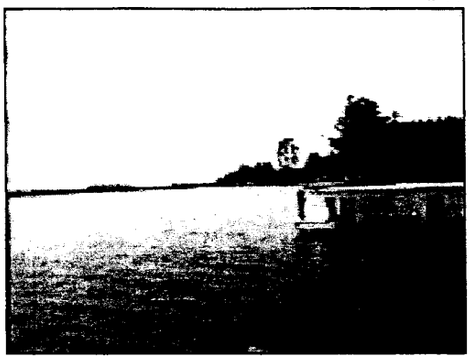


Figura B.40.2. Segundo quadro da seqüência.

Figura B.40.3. Terceiro quadro da seqüência.

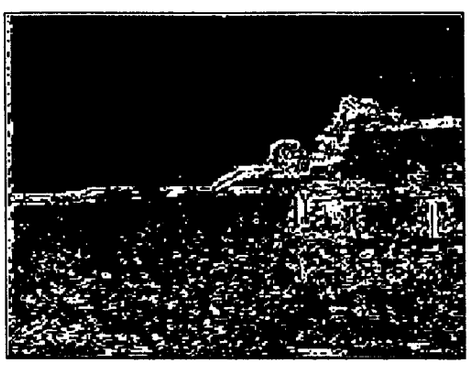


Figura B.40.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

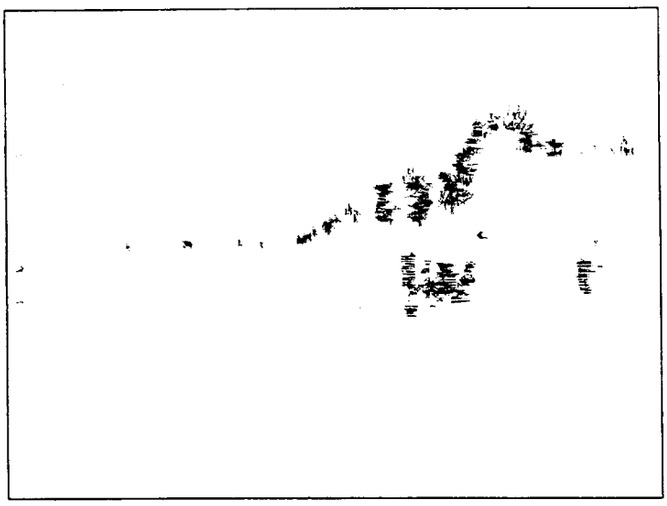


Figura B.40.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,040 s

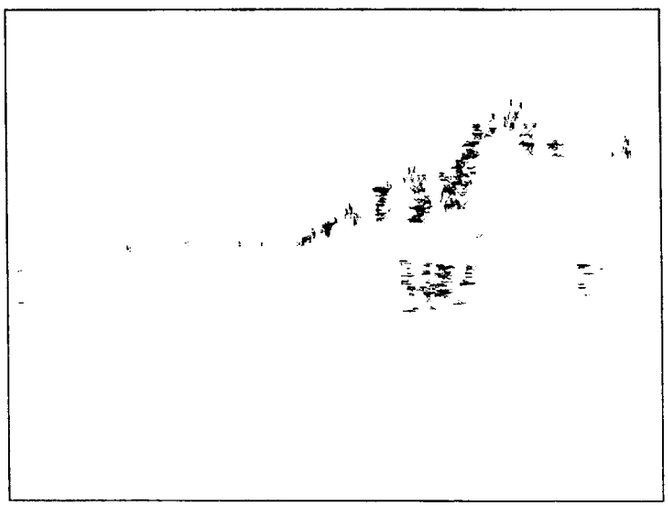


Figura B.40.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,110

Erro Médio Quadrático: 0,02427

Relação entre os tempos de processamento: 9,46 vezes

SERVIÇO DE BIBLIOTECA
INFORMAÇÃO

Figura B.41.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.41.2. Segundo quadro da seqüência.

Figura B.41.3. Terceiro quadro da seqüência.

Figura B.41.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

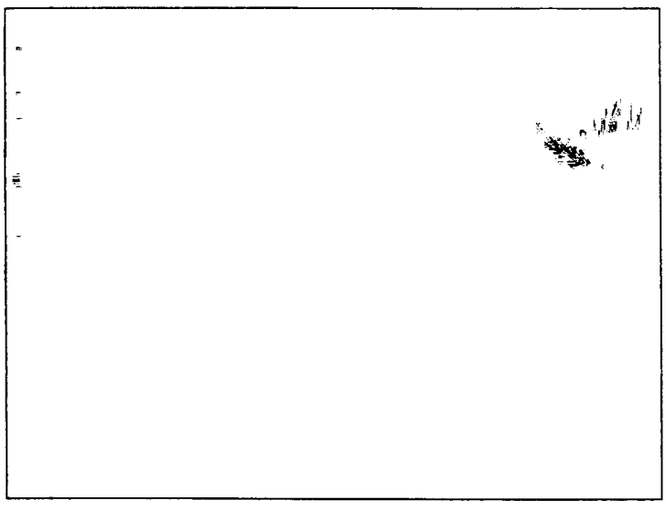


Figura B.41.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,040 s

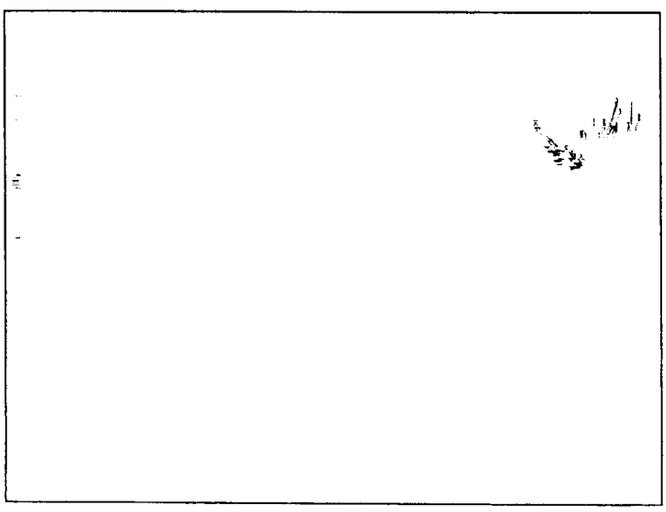


Figura B.41.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,220

Erro Médio Quadrático: 1,30934

Relação entre os tempos de processamento: 4,73 vezes

Figura B.42.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.42.2. Segundo quadro da seqüência.

Figura B.42.3. Terceiro quadro da seqüência.

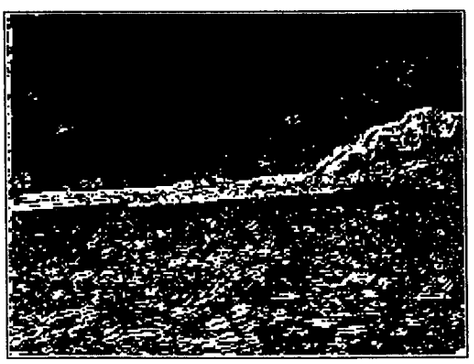


Figura B.42.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

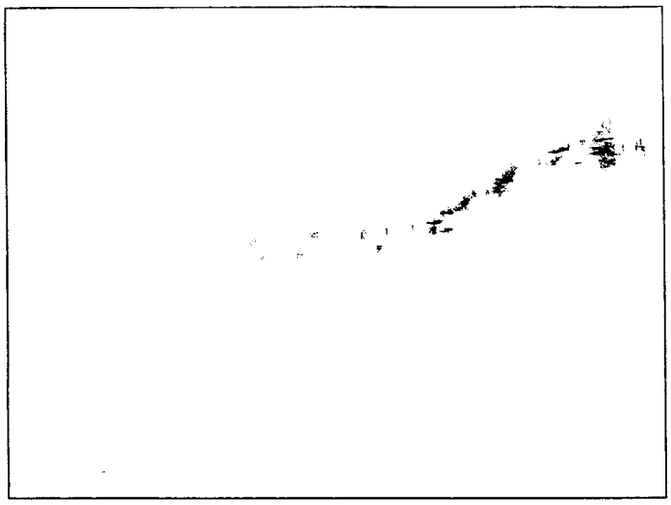


Figura B.42.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,050 s

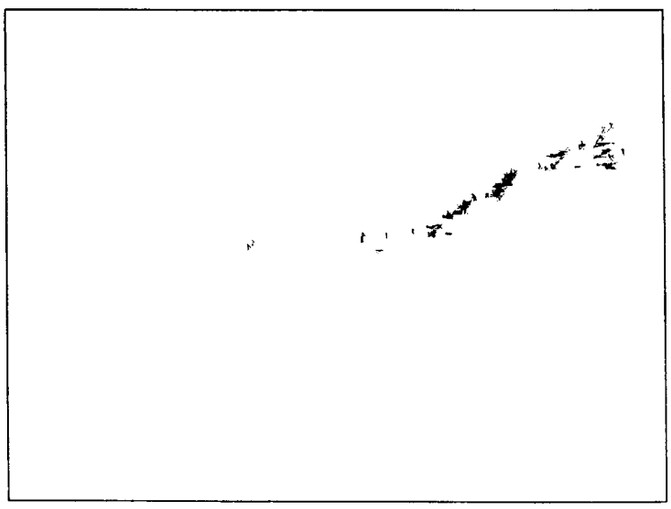


Figura B.42.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,170

Erro Médio Quadrático: 0,00323

Relação entre os tempos de processamento: 6,18 vezes

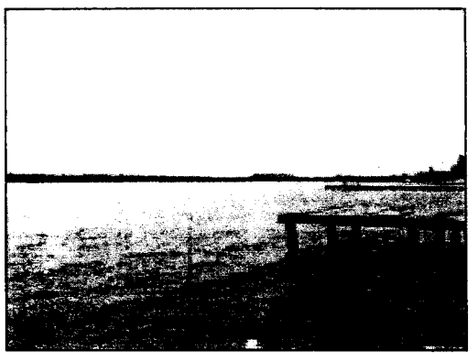


Figura B.43.1. Primeiro quadro de uma seqüência feita com imagens reais.

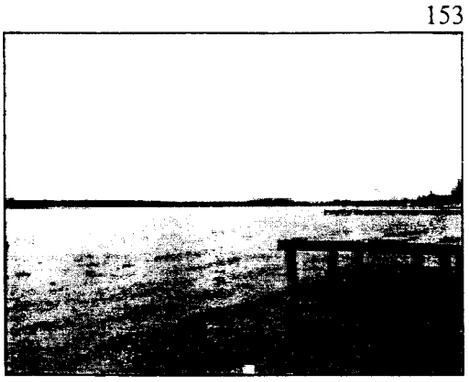


Figura B.43.2. Segundo quadro da seqüência.

Figura B.43.3. Terceiro quadro da seqüência.

Figura B.43.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

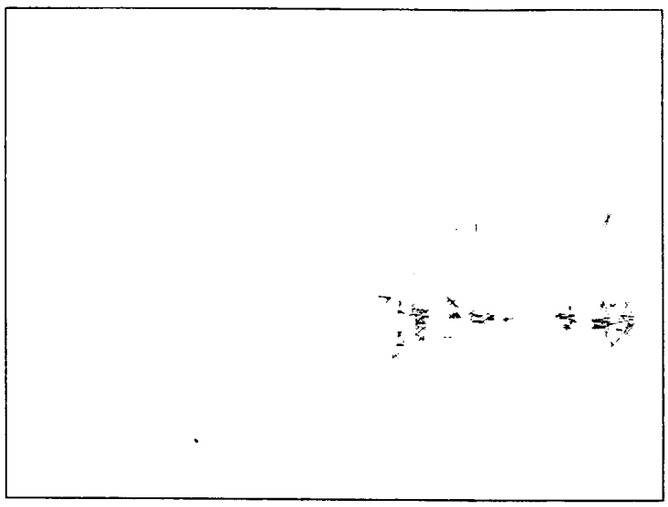


Figura B.43.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,460 s

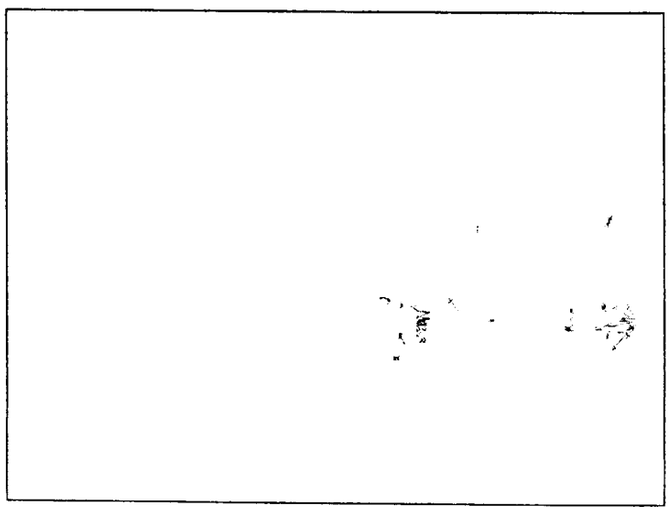


Figura B.43.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,170

Erro Médio Quadrático: 9,88748

Relação entre os tempos de processamento: 8,59 vezes

Figura B.44.1. Primeiro quadro de uma seqüência feita com imagens reais.

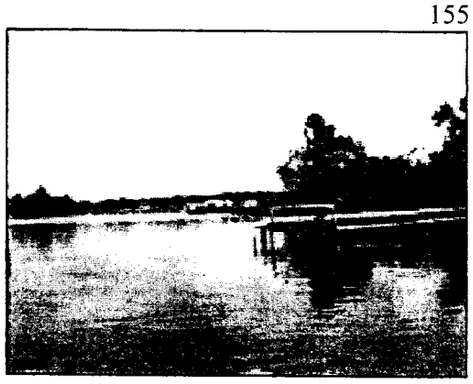


Figura B.44.2. Segundo quadro da seqüência.

Figura B.44.3. Terceiro quadro da seqüência.

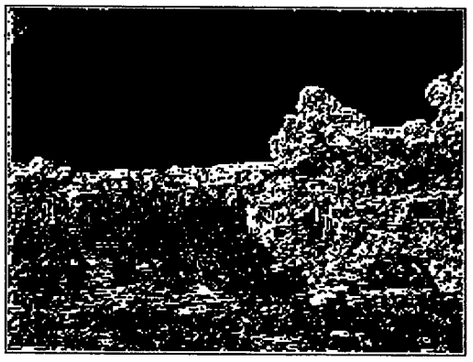


Figura B.44.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

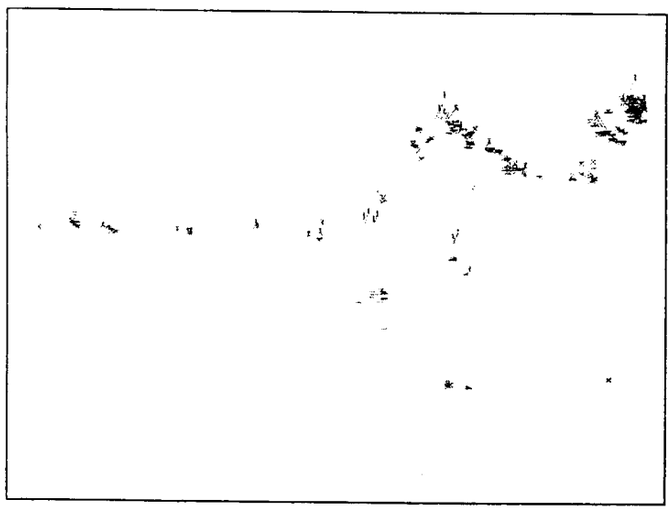


Figura B.44.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,100 s

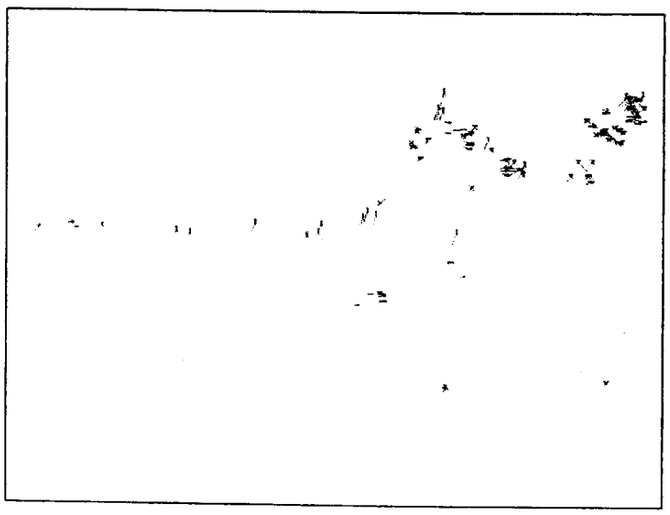


Figura B.44.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,160

Erro Médio Quadrático: 3,56998

Relação entre os tempos de processamento: 6,88 vezes

Figura B.45.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.45.2. Segundo quadro da seqüência.

Figura B.45.3. Terceiro quadro da seqüência.

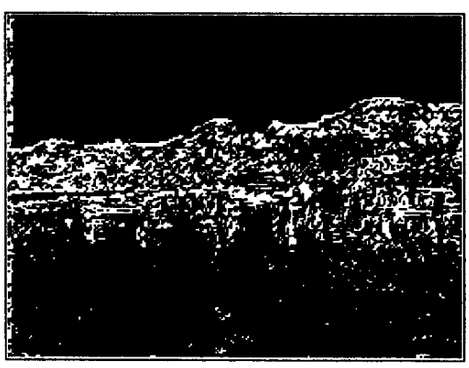


Figura B.45.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

Figura B.47.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,100 s

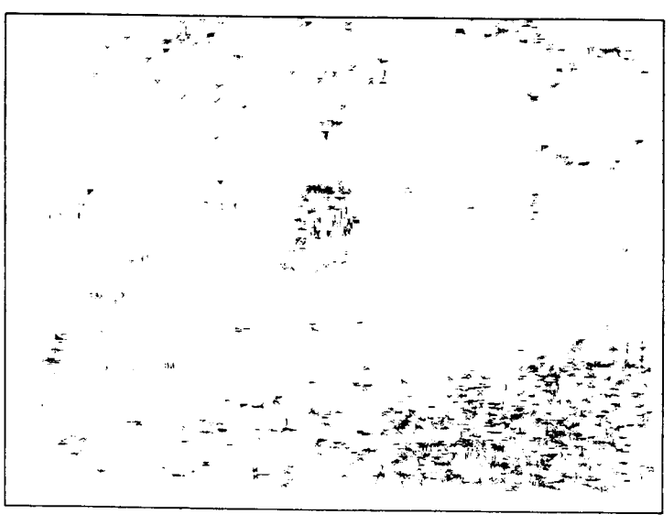


Figura B.47.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,330

Erro Médio Quadrático: 0,00025

Relação entre os tempos de processamento: 3,33 vezes

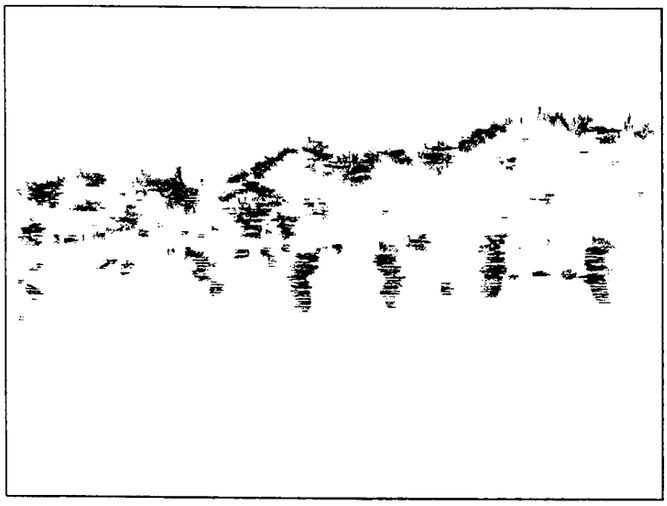


Figura B.45.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,040 s

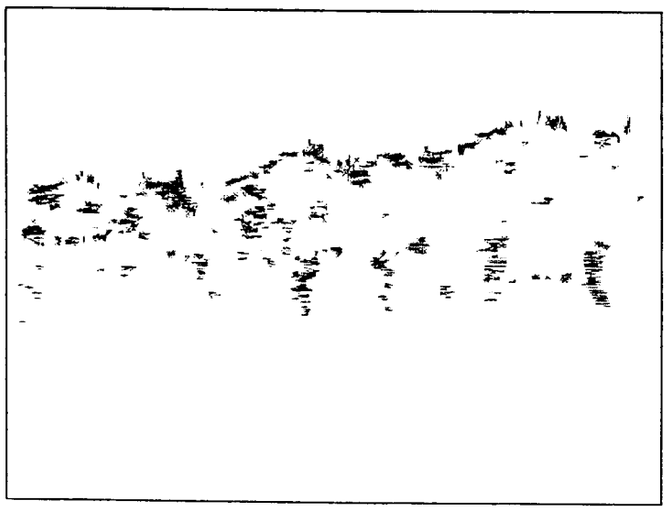


Figura B.45.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,160

Erro Médio Quadrático: 0,04139

Relação entre os tempos de processamento: 6,50 vezes

IFSC-USP

Figura B.46.1. Primeiro quadro de uma seqüência feita com imagens reais.

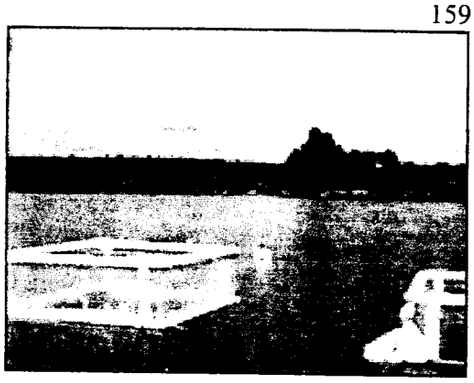


Figura B.46.2. Segundo quadro da seqüência.

Figura B.46.3. Terceiro quadro da seqüência.

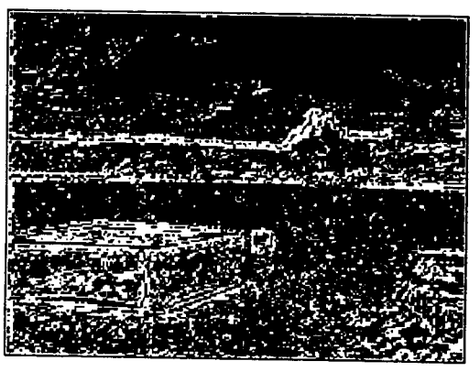


Figura B.46.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

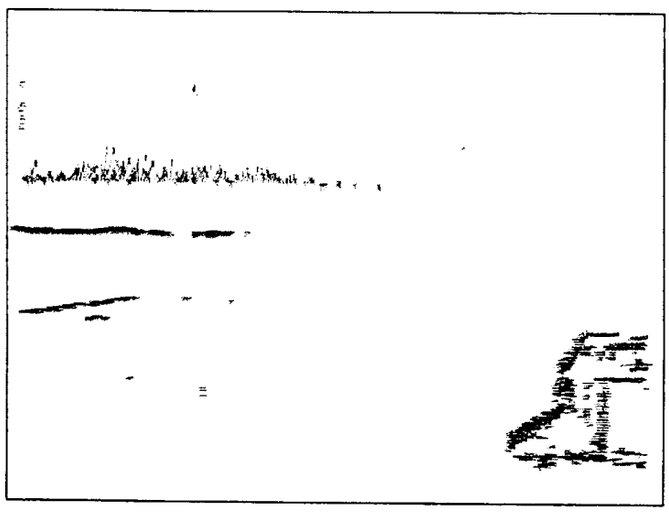


Figura B.46.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,050 s

Figura B.46.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,220

Erro Médio Quadrático: 0,00012

Relação entre os tempos de processamento: 4,77 vezes

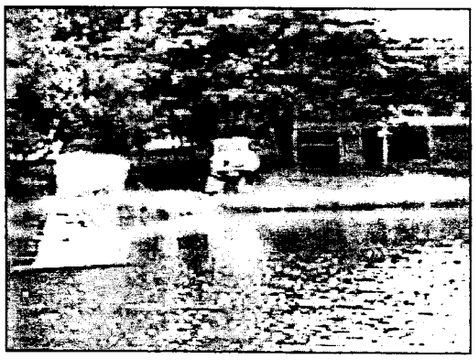


Figura B.47.1. Primeiro quadro de uma seqüência feita com imagens reais.

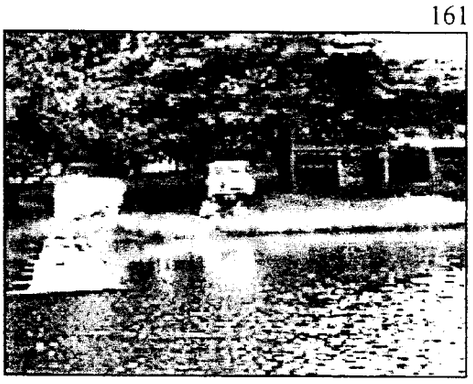


Figura B.47.2. Segundo quadro da seqüência.

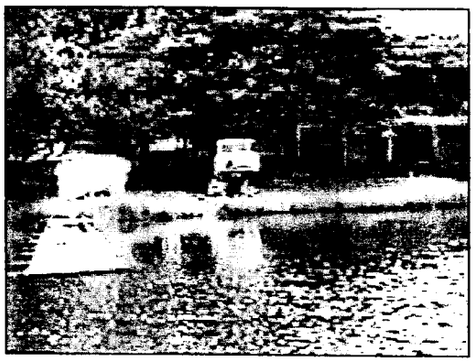


Figura B.47.3. Terceiro quadro da seqüência.

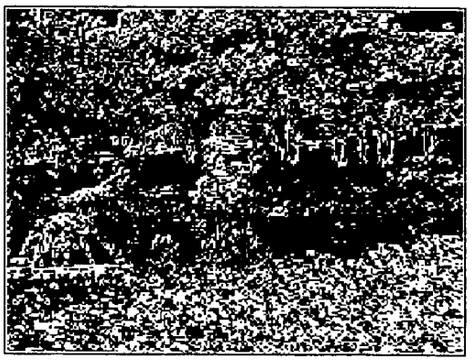


Figura B.47.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

Figura B.47.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 1,100 s

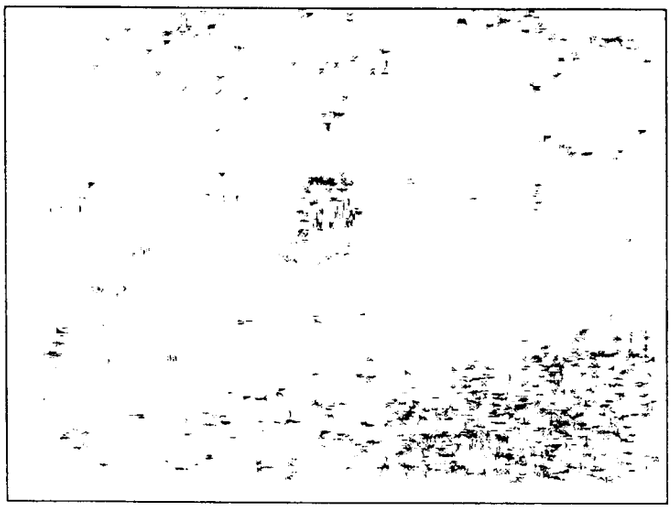


Figura B.47.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,330

Erro Médio Quadrático: 0,00025

Relação entre os tempos de processamento: 3,33 vezes

Figura B.48.1. Primeiro quadro de uma seqüência feita com imagens reais.

Figura B.48.2. Segundo quadro da seqüência.

Figura B.48.3. Terceiro quadro da seqüência.

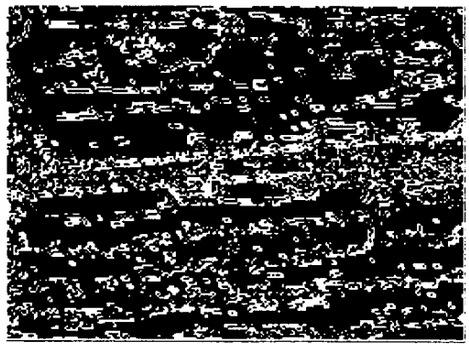


Figura B.48.4. Resultado da aplicação do Método de Cabestaing na devida seqüência.

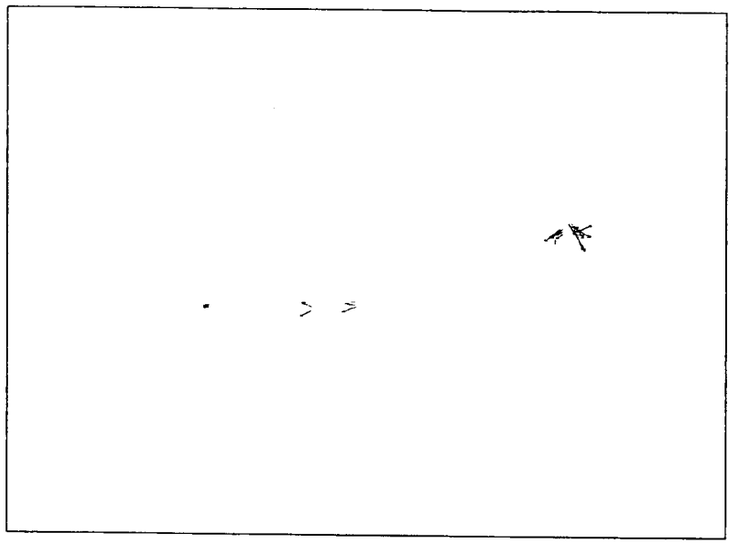


Figura B.48.5. Fluxo ótico da seqüência de imagens reais obtido sem máscara de Cabestaing.

Tempo de processamento: 0,820 s

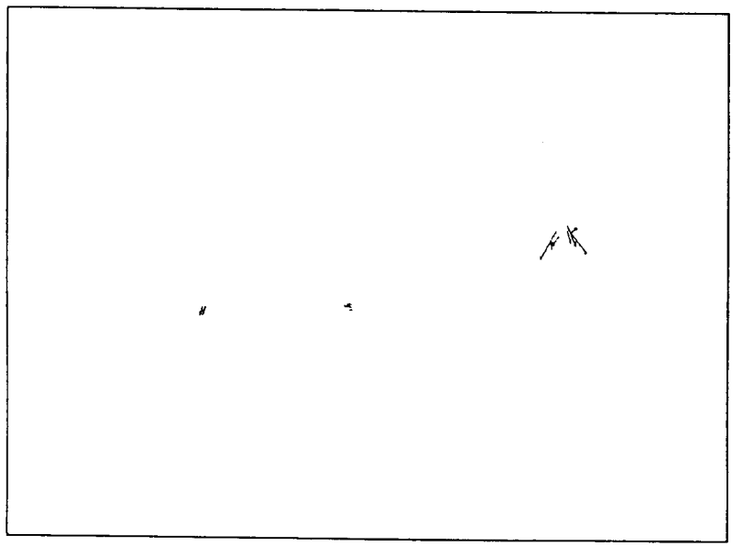


Figura B.48.6. Fluxo ótico obtido com Máscara de Cabestaing.

Tempo de processamento: 0,160 s

Erro Médio Quadrático: 0,91558

Relação entre os tempos de processamento: 5,13 vezes

APÊNDICE C

Listagem do Programa

```
{Este programa calcula o diagrama de agulhas de uma seqüência de dois quadros
através do Fluxo ótico}

unit UnitFlow;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Buttons, ExtDlgs, ComCtrls, Menus, rdtsc;

Const MaxImg = 1024;

type
  velocity_vec = record
     v1,v2 : Real;
    end;

  velocity_mat = Array[0..MaxImg, 0..MaxImg] of velocity_vec;
  TMatriz_Imagem = array [0..MaxImg, 0..MaxImg] of integer;

type
  TForm1 = class(TForm)
    P: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    TabSheet3: TTabSheet;
    AbrirIm: TOpenPictureDialog;
    SalvarIm: TSavePictureDialog;
    ImQ1: TImage;
    ImQ2: TImage;
    ScrollBox1: TScrollBox;
    ImRes: TImage;
    MainMenu1: TMainMenu;
```

```
Imagem1: TMenuItem;
CarregarImagem11: TMenuItem;
CarregarImagem21: TMenuItem;
SalvarResultado1: TMenuItem;
ConfiguraodosParmetros1: TMenuItem;
FluxÓtico1: TMenuItem;
Calcular1: TMenuItem;
N1: TMenuItem;
Sair1: TMenuItem;
Panel2: TPanel;
TabSheet4: TTabSheet;
Imq3: TImage;
CarregarQuadro31: TMenuItem;
PrProcessamento1: TMenuItem;
Cabestaing1: TMenuItem;
Cabestaing21: TMenuItem;
Calculkar21: TMenuItem;
DiagramaAgulhas: TMenuItem;
CalcularErro1: TMenuItem;
TabSobre: TTabSheet;
ScrollBox2: TScrollBox;
IMSobre: TImage;
SobrePorQuadro11: TMenuItem;
Cordovetor1: TMenuItem;
ColorDialog1: TColorDialog;
SalvarSobrePosio1: TMenuItem;
Outros1: TMenuItem;
Velocidadecpu1: TMenuItem;
RelaoSemFiltroComFiltro1: TMenuItem;

procedure ImResMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure ImQ2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure ImQ1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Sair1Click(Sender: TObject);
procedure CarregarImagem11Click(Sender: TObject);
procedure CarregarImagem21Click(Sender: TObject);
procedure SalvarResultado1Click(Sender: TObject);
procedure Calcular1Click(Sender: TObject);
procedure ConfiguraodosParmetros1Click(Sender: TObject);
procedure CarregarQuadro31Click(Sender: TObject);
procedure Cabestaing1Click(Sender: TObject);
procedure Cabestaing21Click(Sender: TObject);
procedure Calculkar21Click(Sender: TObject);
procedure DiagramaAgulhasClick(Sender: TObject);
procedure CalcularErro1Click(Sender: TObject);
procedure SobrePorQuadro11Click(Sender: TObject);
```

IFSC-USP SERVIÇO DE BIBLIOTECA
INFORMAÇÃO

```
    procedure Cordovetor1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure SalvarSobrePosio1Click(Sender: TObject);
    procedure Velocidadecpu1Click(Sender: TObject);
    procedure RelaoSemFiltroComFiltro1Click(Sender: TObject);

  private
    { Private declarations }

  public
    { Public declarations }
    corvetor : Tcolor;

    procedure Estimate_Gradient(var Dx1:real;var Dx2:real;var Dt:real;
                          i:integer;j:integer);

    { Gauss-Seidel interactive numeric method to evaluate the velocity
      vector at pixels coordinates x1 and x2 for frames ImQ1 and ImQ2 }

    function Gauss_Seidel(Dx1:real;Dx2:real;Dt:real;alpha:real;
                    vel:velocity_vec):velocity_vec;

    procedure ComputeOpticalFlow;

    procedure ComputeOpticalFlowFiltro;

    procedure ImageResult(var Mat_vel: velocity_mat);

    procedure SubtraiImagem(im1, im2 : TImage; var ImF :TMatriz_Imagem);

    procedure DetectabordasSobel (var IM1, Bordas : TMatriz_Imagem);
  end;

  var
    Form1: TForm1;
    Mat_vel: velocity_mat;
    Precision: Real;
    Alpha: Real;
    FEV:Integer;
    Mag:Integer;
    Im21, Im32, Bordas21, Bordas32 : TMatriz_Imagem;
    t1, t2 : TDateTime;

    tick1, tick2,
    TComFiltro, TSemFiltro : int64;  //calculo de tempo de processamento


    { Arquivos temporarios }
    FILE1 : file of velocity_vec;
```

```
  FILE2 : file of velocity_vec;

  { Erros a serem calculados }
  eqm,reqm : real;

implementation

uses UOpcoes, ULInha;

{$R *.DFM}


function TForm1.Gauss_Seidel(Dx1:real;Dx2:real;Dt:real;alpha:real;
                  vel:velocity_vec):velocity_vec;
var
 num,denom : real;

begin
  with vel do
   num := Dx1*v1 + Dx2*v2 + Dt;
  denom := alpha*alpha + Dx1*Dx1 + Dx2*Dx2;

  Gauss_Seidel.v1 := vel.v1 - Dx1*num/denom;
  Gauss_Seidel.v2 := vel.v2 - Dx2*num/denom;

end;

procedure TForm1.Estimate_Gradient(var Dx1:real;var Dx2:real;var Dt:real;
                        i:integer;j:integer);
var temp : real;

function Pixel_Int(Im:TImage;i:integer;j:integer):Byte;
 begin
  Pixel_Int := abs(GetRValue(Im.Canvas.Pixels[i,j]));
 end;

begin
  temp := Pixel_Int(Form1.ImQ1,i+1,j);
  Dx1:= temp;
  temp := Pixel_Int(Form1.ImQ1,i,j);
  Dx1:= Dx1 - temp;
  temp := Pixel_Int(Form1.ImQ1,i+1,j+1);
  Dx1:= Dx1 + temp;
  temp := Pixel_Int(Form1.ImQ1,i,j+1);
  Dx1:= Dx1 - temp;
  temp := Pixel_Int(Form1.ImQ2,i+1,j);
  Dx1:= Dx1 + temp;
  temp := Pixel_Int(Form1.ImQ2,i,j);
  Dx1:= Dx1 - temp;
```

```
  temp := Pixel_Int(Form1.ImQ2,i+1,j+1);
  Dx1 := Dx1+temp;
  temp:= Pixel_Int(Form1.ImQ2,i,j+1);
  Dx1:=Dx1+temp;
  Dx1 := Dx1/4;

  Dx2:=( Pixel_Int(Form1.ImQ1,i,j+1)
     - Pixel_Int(Form1.ImQ1,i,j)
     + Pixel_Int(Form1.ImQ1,i+1,j+1)
     - Pixel_Int(Form1.ImQ1,i+1,j)
     + Pixel_Int(Form1.ImQ2,i,j+1)
     - Pixel_Int(Form1.ImQ2,i,j)
     + Pixel_Int(Form1.ImQ2,i+1,j+1)
     - Pixel_Int(Form1.ImQ2,i+1,j) )/4;

  Dt:=( Pixel_Int(Form1.ImQ2,i,j)
    - Pixel_Int(Form1.ImQ1,i,j)
    + Pixel_Int(Form1.ImQ2,i+1,j)
    - Pixel_Int(Form1.ImQ1,i+1,j)
    + Pixel_Int(Form1.ImQ2,i,j+1)
    - Pixel_Int(Form1.ImQ1,i,j)
    + Pixel_Int(Form1.ImQ2,i+1,j+1)
    - Pixel_Int(Form1.ImQ1,i+1,j+1) )/4;
end;

procedure TForm1.ImageResult(var Mat_vel: velocity_mat);
 var
  Maior_x,Maior_y:Real;
  Fat_Norm_x,Fat_Norm_y:Real;
  X_orig,Y_orig,X_offset,Y_offset:Integer;
  i,j:Integer;

begin
  // Encontra maiores componentes x e y de
  // velocidade em Mat_vel

  Maior_x := Mat_vel[0,0].v1;
  Maior_y := Mat_vel[0,0].v2;
  for i:= 1 to ImQ2.Width-2 do
   for j:= 1 to ImQ2.Height-2 do
    begin
     if (Maior_x < Mat_vel[i,j].v1)
      then
       Maior_x := Mat_vel[i,j].v1;
     if (Maior_y < Mat_vel[i,j].v2)
      then
       Maior_y := Mat_vel[i,j].v2;
    end;
```

```
   // Determina fator de normalização a partir das maiores coordenadas
//   ShowMessage('Maior X = '+FloattoStr(Maior_x));
//   ShowMessage('Maior Y = '+FloattoStr(Maior_y));
   Fat_Norm_x := Mag / Maior_x;
   Fat_Norm_y := Mag / Maior_y;
   with Imres do
    begin
     Width := ImQ1.Width*Mag;
     Height:= ImQ2.Height*Mag;
     Picture.BitMap.Width := ImQ1.Width*Mag;
     Picture.BitMap.Height:= ImQ2.Height*Mag;
     Canvas.Brush.color:=ClWhite;
     // Limpa área do canvas de branco antes de usar
     canvas.FillRect(rect(0,0,ImRes.Width-1,ImRes.Height-1));
     Canvas.Pen.color := CorVetor;
     ScrollBox1.Horzscrollbar.range:=Width;
     ScrollBox1.Vertscrollbar.range:=Height;
    end;
  For i:=0 to ImQ1.Width-2 do
   For j:= 0 to ImQ1.Height-2 do
    begin
     // Encontra a posição de origem do vetor (i,j)
     X_orig := i*Mag+Mag div 2;
     Y_orig := j*Mag+Mag div 2;
     // Calcula o deslocamento Normalizado em relacao a origem
     X_offset := FEV*Round(Mat_vel[i,j].v1*Fat_Norm_X); // MMSS
     Y_offset := FEV*Round(Mat_vel[i,j].v2*Fat_Norm_Y); // MMSS
     ImRes.Canvas.Pixels[X_orig,Y_orig] := CorVetor;
     ImRes.Canvas.MoveTo(X_orig,Y_orig);
     ImRes.Canvas.Lineto(X_orig+X_offset,Y_orig+Y_offset);

     // Se houve Fluxo óticodesenho a seta.
     if (X_offset <> 0) or (Y_offset <> 0) then
      FLInha.Seta(X_orig,Y_orig,
            X_orig+X_offset,Y_orig+Y_offset,
            4, 6, ImRes);

     //caption := IntToStr(X_offset) + " + IntToStr(Y_offset);
    end;
  ImRes.UpDate;
end;

procedure TForm1.ComputeOpticalFlow;
var
 i,j : integer;
 Dx1,Dx2,Dt: Real;
 vel: velocity_vec;

function Conv_Crit(val1:real;val2:real):real;
```

```
 begin
   Conv_Crit := abs(val1 - val2);
 end;

begin

 for i:= 0 to Form1.ImQ2.Width-2 do
  for j:= 0 to Form1.ImQ2.Height-2 do
   begin
    vel.v1:=0.01;
    vel.v2:=0.01;
    Estimate_Gradient(Dx1,Dx2,Dt,i,j);
    repeat
     Mat_vel[i,j].v1:=vel.v1;
     Mat_vel[i,j].v2:=vel.v2;
     vel:= Gauss_Seidel(Dx1,Dx2,Dt,alpha,Mat_vel[i,j]);
    until ( (Conv_Crit(Mat_vel[i,j].v1,vel.v1)<= precision)
         AND (Conv_Crit(Mat_vel[i,j].v2,vel.v2) <= precision) );
   end;

   tick2 := cpugettick;
//   t2 := now;

   ImageResult(Mat_vel); { Constroi o campo vetorial de velocidades }

end;

procedure TForm1.ComputeOpticalFlowFiltro;
var
 i,j, cont, cont2 : integer;
 Dx1,Dx2,Dt: Real;
 vel: velocity_vec;

function Conv_Crit(val1:real;val2:real):real;
 begin
   Conv_Crit := abs(val1 - val2);
 end;

begin
 cont := 0;
 cont2 := 0;

 for i:= 0 to Form1.ImQ2.Width-2 do
  for j:= 0 to Form1.ImQ2.Height-2 do
   if Bordas32[i,j] = 0 then
    begin
      Mat_vel[i,j].v1:=0;
      Mat_vel[i,j].v2:=0;
      cont := cont + 1;
```

```
    end
   else
    begin
     vel.v1:=0.01;
     vel.v2:=0.01;
     Estimate_Gradient(Dx1,Dx2,Dt,i,j);
     repeat
      Mat_vel[i,j].v1:=vel.v1;
      Mat_vel[i,j].v2:=vel.v2;
      vel:= Gauss_Seidel(Dx1,Dx2,Dt,alpha,Mat_vel[i,j]);
     until ( (Conv_Crit(Mat_vel[i,j].v1,vel.v1)<= precision)
        AND (Conv_Crit(Mat_vel[i,j].v2,vel.v2) <= precision) );
         cont2 := cont2 + 1;
    end;

  tick2 := cpugettick;
//  t2 := now;

  ImageResult(Mat_vel); { Constroi o campo vetorial de velocidades }
  showMessage(inttostr(cont));
  showMessage(inttostr(cont2));
end;

procedure TForm1.SubtraiImagem(im1, im2 : TImage; var ImF :TMatriz_Imagem);
var i,j, v1, v2 : integer;
begin
 for i:= 0 to Form1.ImQ2.Width-1 do
  for j:= 0 to Form1.ImQ2.Height-1 do
   begin
    v1 := GetRValue(im1.canvas.pixels[i,j]);
    v2 := GetRValue(im2.canvas.pixels[i,j]);
    Imf[i,j] := abs (v1 - v2);
   end;
end;

procedure TForm1.DetectabordasSobel (var IM1, Bordas : TMatriz_Imagem);
var i,j : integer;
begin
 for i:= 1 to Form1.ImQ2.Width-2 do
  for j:= 1 to Form1.ImQ2.Height-2 do
   begin
    Bordas[i,j] := ((Im1[i-1,j+1] + 2 * Im1[i, j +1] + Im1[i+1,j+1]) -
                (Im1[i-1,j-1] + 2 * Im1[i, j -1] + Im1[i+1,j-1]) +

                (Im1[i+1,j+1] + 2 * Im1[i+1, j ] + Im1[i+1,j-1]) -
                (Im1[i-1,j+1] + 2 * Im1[i-1, j ] + Im1[i-1,j-1])) div 2;
   end;
end;
```

SC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO

```
{=========================================}
procedure TForm1.ImResMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  //Coords.Visible := True;
  Panel2.Caption := 'X =' + IntToStr(x) + '  Y =' + IntToStr(y);
end;

procedure TForm1.ImQ2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
//  Coords.Visible := True;
  Panel2.Caption := 'X =' + IntToStr(x) + '  Y =' + IntToStr(y);
end;

procedure TForm1.ImQ1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  Visible := True;
  Panel2.Caption := 'X =' + IntToStr(x) + '  Y =' + IntToStr(y);
end;

procedure TForm1.Sair1Click(Sender: TObject);
begin
close;
end;

procedure TForm1.CarregarImagem11Click(Sender: TObject);
begin
If AbrirIm.execute then
  ImQ1.picture.bitmap.LoadFromFile(AbrirIm.FileName);
end;

procedure TForm1.CarregarImagem21Click(Sender: TObject);
begin
If AbrirIm.execute then
  ImQ2.picture.bitmap.LoadFromFile(AbrirIm.FileName);
end;

procedure TForm1.SalvarResultado1Click(Sender: TObject);
begin
If SalvarIm.execute then
  ImRes.picture.bitmap.savetofile(SalvarIm.filename);
end;

procedure TForm1.Calcular1Click(Sender: TObject);
var //H1, M1, S1, MSec1 : word;
  // H2, M2, S2, MSec2 : word;
```

```
  i,j: Integer;

begin
screen.cursor := CrHourGlass;

with FOpcoes do
 begin
  Mag:= StrtoInt(Edit3.Text);
  Precision:= Strtofloat(Edit1.Text);
  Alpha:= Strtofloat(Edit2.Text);
  FEV := StrtoInt(Edit4.Text);
 end;

//t1 := now;
tick1 := cpugettick;
ComputeOpticalFlow;



{Arquivo para armazenar matriz }
{ para calculo do erro }
{ entre o calculo via cabestaing }
{ e fluxo óticoconvencional }
{ File2 -> Dados do cabestaing  }
{ File1 -> Dados do fluxo óticoconvencional }

AssignFile(File1,'$TMP1');
rewrite(File1);

for i := 1 to ImQ2.Width-2 do
  for j := 1 to ImQ2.Height-2 do
    write(File1,Mat_Vel[i,j]);
closefile(file1);

showmessage(format('Tempo = %d microsegundos', [cpucalcmks(tick2,tick1)]));
showmessage(format('Tempo = %d milisegundos', [cpucalcms(tick2,tick1)]));
TSemFiltro := cpucalcmks(tick2,tick1);


//DecodeTime(t1, H1, M1, S1, MSec1);
//DecodeTime(t2, H2, M2, S2, MSec2);

//ShowMessage(TimeToStr(t1) + ':' + INtTostr(Msec1) + ' - ' +
//          TimeToStr(t2) + ':' + INtTostr(Msec2));

screen.cursor := CrDefault;
end;

procedure TForm1.ConfiguraodosParmetros1Click(Sender: TObject);
```

```
begin
 FOpcoes.Show;
end;

procedure TForm1.CarregarQuadro31Click(Sender: TObject);
begin
If AbrirIm.execute then
   ImQ3.picture.bitmap.LoadFromFile(AbrirIm.FileName);
end;

procedure TForm1.Cabestaing1Click(Sender: TObject);
var i,j, maior, cor : integer;
   fator : real;
begin
 SubtrailmagemCimq1, imq2, Im21);
 SubtrailmagemCimq2, imq3, Im32);

 DetectabordasSobel(im21, Bordas21);
 DetectabordasSobel(im32, Bordas32);

 for i:= 1 to Form1.ImQ2.Width-2 do
  for j:= 1 to Form1.ImQ2.Height-2 do
     Bordas32[i,j] := Bordas32[i,j] + Bordas21[i,j];

 maior := bordas32[1,1];

 for i:= 1 to Form1.ImQ2.Width-2 do
  for j:= 1 to Form1.ImQ2.Height-2 do
    if bordas32[i,j] > maior then maior := bordas32[i,j];

 fator := 255 / maior;

  with Imres do
   begin
    Width := ImQ1.Width;
    Height:= ImQ2.Height;
    Picture.BitMap.Width := ImQ1.Width;
    Picture.BitMap.Height:= ImQ2.Height;
    Canvas.Brush.color:=ClWhite;
    // Limpa área do canvas de branco antes de usar
    canvas.FillRect(rect(0,0,ImRes.Width-1,ImRes.Height-1));
    Canvas.Pen.color := CorVetor;
    ScrollBox1.Horzscrollbar.range:=Width;
    ScrollBox1.Vertscrollbar.range:=Height;
   end;

 for i:= 1 to Form1.ImQ2.Width-2 do
  for j:= 1 to Form1.ImQ2.Height-2 do
    begin
```

IFSC-USP SERVIÇO DE BIBLIOTECA E INFORMAÇÃO

```
      cor := round (fator * bordas32[i,j]);
      if cor > 0 then cor := 255;
      IMres.canvas.pixels[i,j] := rgb(cor, cor, cor);
     end;
 ImRes.update;
end;

procedure TForm1.Cabestaing21Click(Sender: TObject);
var i,j, maior, cor : integer;
   fator : real;
begin
 SubtraiImagem(imq1, imq2, Im21);
 SubtraiImagem(imq2, imq3, Im32);

 DetectabordasSobel(im21, Bordas21);
 DetectabordasSobel(im32, Bordas32);

 for i:= 1 to Form1.ImQ2.Width-2 do
  for j:= 1 to Form1.ImQ2.Height-2 do
    Bordas32[i,j] := Bordas32[i,j] * Bordas21[i,j];

 maior := bordas32[1,1];

 for i:= 1 to Form1.ImQ2.Width-2 do
  for j:= 1 to Form1.ImQ2.Height-2 do
   if bordas32[i,j] > maior then maior := bordas32[i,j];

 fator := 255 / maior;

  with Imres do
   begin
    Width := ImQ1.Width;
    Height:= ImQ2.Height;
    Picture.BitMap.Width := ImQ1.Width;
    Picture.BitMap.Height:= ImQ2.Height;
    Canvas.Brush.color:=ClWhite;
    // Limpa área do canvas de branco antes de usar
    canvas.FillRect(rect(0,0,ImRes.Width-1,ImRes.Height-1));
    Canvas.Pen.color := CorVetor;
    ScrollBox1.Horzscrollbar.range:=Width;
    ScrollBox1.Vertscrollbar.range:=Height;
   end;

 for i:= 1 to Form1.ImQ2.Width-2 do
  for j:= 1 to Form1.ImQ2.Height-2 do
    begin
     cor := round (fator * bordas32[i,j]);
     if cor > 0 then cor := 255
            else cor := 0 ;
```

```
        bordas32[i,j] := cor;
        IMres.canvas.pixels[i,j] := rgb(cor, cor, cor);
      end;
  ImRes.update;

end;

procedure TForm1.Calculkar21Click(Sender: TObject);
var //H1, M1, S1, MSec1 : word;
    //H2, M2, S2, MSec2 : word;
    i,j : integer;

begin
screen.cursor := CrHourGlass;
with FOpcoes do
 begin
   Mag:= StrtoInt(Edit3.Text);
   Precision:= Strtofloat(Edit1.Text);
   Alpha:= Strtofloat(Edit2.Text);
   FEV := StrtoInt(Edit4.Text);
 end;

//t1 := now;

tick1 := cpugettick;
ComputeOpticalFlowFiltro;


{ Arquivo temporario }
{para armazenar matriz }
{ para calculo do erro }
{ entre o calculo via cabestaing }
{ e fluxo óticoconvencional }
{ File2 -> Dados do cabestaing  }
{ File1 -> Dados do fluxo óticoconvencional }

AssignFile(File2,'$TMP2');
rewrite(File2);
for i:= 1 to ImQ2.Width-2 do
  for j:= 1 to ImQ2.Height-2 do
    write(File2,Mat_Vel[i,j]);
closefile(file2);

//DecodeTime(t1, H1, M1, S1, MSec1);
//DecodeTime(t2, H2, M2, S2, MSec2);

showmessage(format('Tempo = %d microsegundos', [cpucalcmks(tick2,tick1)]));
showmessage(format('Tempo = %d milisegundos', [cpucalcms(tick2,tick1)]));
TComFiltro := cpucalcmks(tick2,tick1);
```

```
//ShowMessage(TimeToStr(t1) + ':' + INtTostr(Msec1) + ' - ' +
//          TimeToStr(t2) + ':' + INtTostr(Msec2));

screen.cursor := CrDefault;
end;

procedure TForm1.DiagramaAgulhasClick(Sender: TObject);
begin
  FLinha.ShowModal;
end;


{ Compara os dados amazenados nos
{ arquivos temporarios FILE 1 e FILE2 }

procedure TForm1.CalcularErro1Click(Sender: TObject);

function modulo(v : velocity_vec) : real;
 begin
   modulo := (sqr(v.v1*v.v1 + v.v2*v.v2));
 end;

var
 v1,v2 : velocity_vec;
 acum_dif : real;
 acum_quad  : real;
 i,j : integer;

begin

AssignFile(File1,'$TMP1');
AssignFile(File2,'$TMP2');
FileMode := 0;   { Readonly }
reset(File1);
reset(File2);

acum_dif := 0;
acum_quad := 0;

for i:= 1 to ImQ2.Width-2 do
 for j:= 1 to ImQ2.Height-2 do
  begin
    read(File1,v1);
    read(File2,v2);
    acum_dif := acum_dif + sqr(modulo(v1) - modulo(v2));
    acum_quad := acum_quad + sqr(modulo(v1));
  end;
```

```
{ erro quadratico medio }
 eqm := acum_dif /((ImQ2.Width- 2)*(ImQ2.Height-2));

{ raiz quadrada do erro quadratico medio }

 reqm := sqrt(acum_dif/acum_quad);

{ Mostra resultados }

 ShowMessage('eqm =' + FloatTostr(eqm));
 ShowMessage('reqm =' + FloatTostr(reqm));

 { Fecha arquivos temporarios }

 closefile(file1);
 closefile(file2);

 { remove arquivos temporarios do disco }
 deletefile('$TMP1');
 deletefile('$TMP2');

end;

procedure TForm1.SobrePorQuadro11Click(Sender: TObject);
var i, j : integer;
   ret1 : Trect;
begin
 IMSobre.Width := ImQ1.Width*Mag;
 IMSobre.Height:= ImQ1.Height*Mag;

 IMSobre.Picture.BitMap.Width := ImQ1.Width*Mag;
 IMSobre.Picture.BitMap.Height:= ImQ1.Height*Mag;

 ret1 := rect(0,0, imsobre.width -1, imsobre.height -1);
 imsobre.Canvas.StretchDraw(ret1, imq1.picture.bitmap);

 for i := 0 to imres.width - 1 do
  for j := 0 to imres.Height - 1 do
    begin
     if imres.Canvas.pixels[i,j] = CorVetor then
       imsobre.Canvas.pixels[i,j] := corvetor;
    end;
end;

procedure TForm1.Cordovetor1Click(Sender: TObject);
begin
 if colordialog1.Execute then
   CorVetor := colordialog1.Color;
end;
```

```
procedure TForm1.FormShow(Sender: TObject);
begin
  CorVetor := CLRed;
end;

procedure TForm1.SalvarSobrePosio1Click(Sender: TObject);
begin
If SalvarIm.execute then
   ImSobre.picture.bitmap.savetofile(SalvarIm.filename);
end;

procedure TForm1.Velocidadecpu1Click(Sender: TObject);
begin
  showmessage(inttostr(cpuGetSpeed));
end;

procedure TForm1.RelaoSemFiltroComFiltro1Click(Sender: TObject);
begin
  showMessage('Relação de desempenho Sem Filtro / Com Filtro  :' +
          Floattostr(TSemFiltro/TComFiltro));
end;

end.
```

IFSC-USP SERVIÇO DE BIBLIOTECA INFORMAÇÃO