

Um Algoritmo Eficiente para Estimação de Movimentos Complexos

LÍRIDA A. B. NAVINER¹
JEAN-FRANÇOIS NAVINER²
ELMAR MELCHER¹
JOÃO MARQUES DE CARVALHO¹

¹Universidade Federal da Paraíba
Departamento de Engenharia Elétrica -LAPS
58100 - Campina Grande - PB, Brasil
lirida@dee.ufpb.br

²Ecole Nationale Supérieure des Télécommunications
Département Electronique
75634 - Paris CEDEX 13, França
naviner@elec.enst.fr

Abstract. This paper deals with an efficient motion estimation algorithm for complex displacements. The approach is based on differential method using parametric modelling. Multiresolution and multiprediction are used in order to improve the estimation and save the computation power. The results obtained with this algorithm show that it is particularly efficient for temporal interpolation.

1 Introdução

Geralmente, em sistemas de codificação de imagens, procede-se a uma sub-amostragem temporal. Nem todas as imagens são transmitidas a fim de obter maior compressão [MPEG4 (1994)], [Recomm (1991)] (ver Figura 1). A recuperação da seqüência de imagens original é feita no decodificador através de uma sobre-amostragem. Para a sobre-amostragem, utilizam-se técnicas de interpolação. Nos casos em que as cenas das imagens contêm movimento, a interpolação deve ocorrer na direção desse movimento, para compensá-lo [Barros (1994)]. Muitas técnicas foram desenvolvidas para analisar o movimento em seqüências de imagens [Bouthémy, (1988)], [Cafforio (1976)], [Bergmann (1993)], [Musmann (1985)]. As características dessas técnicas podem ser resumidas como segue:

- Os métodos por **transformada de Fourier** são baseados na propriedade dessa transformada que relaciona um deslocamento no domínio espacial com uma mudança de fase no domínio das frequências. Esses métodos apresentam a desvantagem de fornecer apenas um vetor por imagem, mas são úteis para indicar o movimento global da cena [Wu (1989)].
- Os métodos por **block-matching** permitem a detecção do movimento de vários objetos em uma cena. Eles são simples e podem se traduzir em operações

regulares, o que os torna interessantes do ponto de vista da implementação *hardware*, mas eles são limitados à estimação de movimentos do tipo translacional [Thoma (1989)], [Colavin (1991)], [Komarek (1989)].

- Os métodos **diferenciais**, normalmente aplicados em um esquema de recursividade, envolvem operações mais complexas e menos regulares que aquelas dos métodos por *block-matching*. Eles são, entretanto, potencialmente mais poderosos, uma vez que permitem a modelagem de movimentos complexos, isto é, resultantes de desde uma simples translação até rotações ou deformações complexas de objetos [Sanson (1991)]. Eles apresentam ainda a característica de uma procura inteligente de movimento, através da consideração das características locais da imagem [Limb (1975)], [Cafforio (1976)].

Essas reflexões conduziram ao desenvolvimento de um algoritmo baseado em um método diferencial. De fato, são comuns as situações em que as seqüências de imagens apresentam vários objetos se deslocando ao mesmo tempo. Além disso, os movimentos podem ser de todos os tipos (rotação, homotetias, etc.).

Este artigo trata do estudo de um algoritmo de estimação de movimentos complexos. O algoritmo proposto e algumas considerações arquiteturais correspondentes são mostrados nas seções 2 e 3, respectivamente. A seção 4 concerne a validação desse

algoritmo. Na seção 5, algumas conclusões são apresentadas.

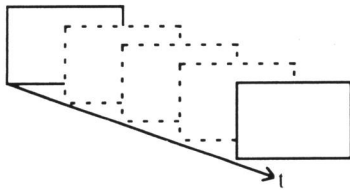


Figura 1: Compressão de imagens por sub-amostragem temporal: as linhas pontilhadas indicam os quadros não enviados.

2 Estimação de Movimentos Complexos

O algoritmo de estimação de movimentos complexos apresentado aqui foi desenvolvido no *Centre Commun d'Etudes de Télédiffusion et Télécommunications* em Rennes, França [Sanson (1991)], [Barros (1993)]. Trata-se de um algoritmo que associa modelagem a posteriori do fluxo ótico e estimação direta do modelo através de mínimos quadrados. A estimação é feita de modo pontual, sob a condição que o campo siga um modelo linear para uma determinada região da imagem. O deslocamento de um pixel é dado por:

$$\mathbf{d} = \mathbf{A}(\mathbf{r} - \mathbf{r}_c) + \mathbf{T} \quad (1)$$

onde:

- \mathbf{T} é um vetor de translação correspondendo ao deslocamento médio da região considerada
- \mathbf{A} é uma matriz 2x2 que descreve a deformação da região considerada
- \mathbf{r}_c é o centro da deformação
- \mathbf{r} é a posição do pixel cujo deslocamento está sendo medido

Sendo um método diferencial, este algoritmo é baseado no desenvolvimento de primeira ordem de Taylor para a luminância espaço-temporal sobre a trajetória. De acordo com a hipótese de invariância da luminância I sobre essa trajetória, tem-se:

$$\begin{cases} \nabla I_t(\mathbf{r}, t) = 0 \\ \partial I_t(\mathbf{r}, t) = \nabla I_t(\mathbf{r}, t) + \nabla I_t(\mathbf{r}, t) \nabla \mathbf{r}_t \end{cases} \quad (2)$$

O desenvolvimento de Taylor supõe a luminância espaço-temporal derivável. Isto é,

- os pixels cujos movimentos estão sendo estimados deslocam-se a baixa velocidade com relação à frequência de amostragem temporal.

- as imagens são suficientemente filtradas passa-baixas.

Para a estimação de movimentos em seqüências reais, deve-se recorrer à recursividade [Netravali (1978)]. Partindo-se da hipótese de uma boa predição, a equação (2) é utilizada em várias iterações para corrigir/atualizar esse valor. O problema corresponde a uma equação com duas incógnitas. Uma solução possível é a componente do deslocamento perpendicular ao contorno local [Choquet (1988)]. Na prática, trata-se de procurar a variação mínima da luminância sobre a trajetória dada na equação (3).

$$\mathbf{d}_\perp = \nabla I_t(\mathbf{r}, t) \frac{\nabla \mathbf{r}_t}{|\nabla \mathbf{r}_t|^2} \quad (3)$$

Como critério para minimização, utiliza-se uma função crescente do valor absoluto ao quadrado dessa variação, definida por:

$$\begin{cases} \sum_{\text{bloco}} |TFD|^2 \\ TFD = I(\mathbf{r}, t) - I(\mathbf{r} - \mathbf{d}, t - 1) \end{cases} \quad (4)$$

TFD (*transformed frame difference*) é a diferença de luminância entre uma região da imagem no instante t e a região equivalente - após estimação de movimentos - dessa região no instante $t-1$. Uma vez que a confiabilidade desses resultados é diretamente ligada à dimensão dos gradientes, em vez de atribuir a mesma importância a todos os pixels, a participação do deslocamento em cada ponto é ponderada por uma função crescente do gradiente no ponto deslocado.

Esse problema de mínimos quadrados ponderados leva a um sistema de equações lineares cuja solução é apresentada na equação (5) [Barros (1993)]. Note-se que se trata de um enfoque paramétrico, isto é, não se procura o movimento propriamente dito, mas os parâmetros do modelo que o descreve (a matriz de deformação \mathbf{A} e o vetor de translação \mathbf{T}).

O algoritmo se aplica em um esquema de recursividade hierárquica. Segmentando a imagem em blocos de tamanhos decrescentes, segundo um *quadtree* com recobrimento, geram-se N níveis, N sendo um parâmetro a determinar. A um dado nível, para cada bloco (bloco filho), os modelos dos quatro blocos conexos (blocos pais) são utilizados como predições iniciais (ver Figura 2). Além dessas quatro predições, uma quinta predição, correspondendo à ausência de movimento, é utilizada.

$$\begin{cases} \partial \mathbf{T}_{k+1} = \frac{-\sum_r TFD(\mathbf{r}, \mathbf{d}_k) \nabla I_r(\mathbf{r}, \mathbf{d}_k) - \partial \mathbf{A} \sum_r \nabla I_r(\mathbf{r}, \mathbf{d}_k)^2 \partial \mathbf{r}}{\sum_r |\nabla I_r(\mathbf{r}, \mathbf{d}_k)|^2} \\ \partial \mathbf{A}_{k+1} = \frac{-\sum_r TFD(\mathbf{r}, \mathbf{d}_k) \nabla I_r(\mathbf{r}, \mathbf{d}_k) \partial \mathbf{r}^T - \partial \mathbf{T} \sum_r \nabla I_r(\mathbf{r}, \mathbf{d}_k)^2 \partial \mathbf{r}^T}{\sum_r |\nabla I_r(\mathbf{r}, \mathbf{d}_k)|^2 \partial \mathbf{r} \partial \mathbf{r}^T} \end{cases} \quad (5)$$

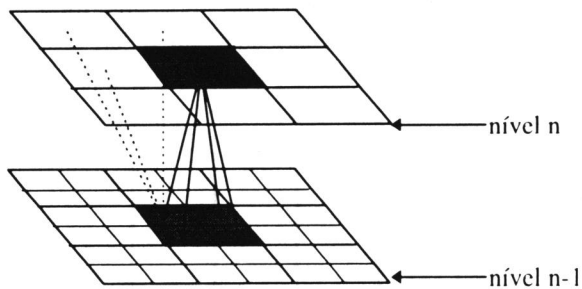


Figura 2: Recursividade hierarquica e *quadtree* com recobrimento. As linhas pontilhadas indicam o recobrimento das predições iniciais

A redução da complexidade de cálculo é obtida com análise multi-resolução. A imagem é filtrada/subamostrada de um fator dois em cada direção sucessivamente, gerando uma pirâmide (ver Figura 3). A estimação de movimentos começa no topo da pirâmide. A multi-resolução permite uma detecção mais rápida de movimentos grandes: um movimento detectado a um dado nível da pirâmide corresponde a um movimento duas vezes maior no nível imediatamente inferior (nível de resolução imediatamente superior).

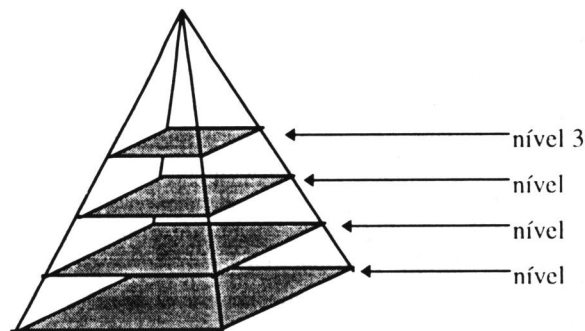


Figura 3: Exemplo de pirâmide para quatro níveis de resolução.

Com multi-resolução, os blocos processados são todos de mesmo tamanho: o tamanho daqueles na imagem com resolução completa. Quando da passagem de parâmetros de um nível a outro, os valores das translações devem ser multiplicados por dois, para compensar a multi-resolução.

3 Considerações Arquiteturais

Seja o fluxo de pixels no sistema para estimação de movimentos mostrado na Figura 4. Os pixels ativos chegam a uma frequência F_p (correspondendo à taxa D_p). O algoritmo inclui o processamento da imagem a várias resoluções, gerando um aumento para a taxa, que passa a ser D_M . A atualização de um jogo de parâmetros exige a consideração de uma vizinhança de tamanho V^2 para as interpolações (casos em que o estimador conduz a um vetor de coordenadas fracionárias), levando a uma taxa entre memória-estimador D_{ME} . Essa é a taxa na qual o circuito recebe dados para realizar a equação (5). D_E é a taxa na qual o estimador fornece os parâmetros do modelo do movimento \mathbf{A} e \mathbf{T} .

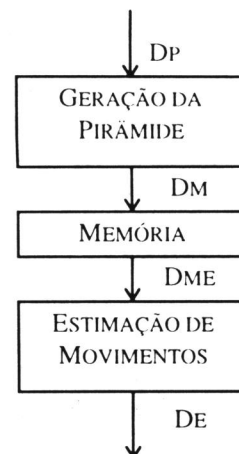


Figura 4: Fluxo de pixels no sistema de estimação de movimentos.

Considerando imagens em formato CCIR e quatro níveis de resolução, o processamento em tempo real implica no cálculo dessas equações em 7,3 ns! Uma tradução direta do algoritmo em arquitetura sem nenhuma otimização com a tecnologia atual não permite atingir tamanho desempenho. Compromissos com *pipeline*, paralelização e partição são indispensáveis [Barros (1993)]. As técnicas de *pipeline* e paralelização são ilustradas na Figura 5.

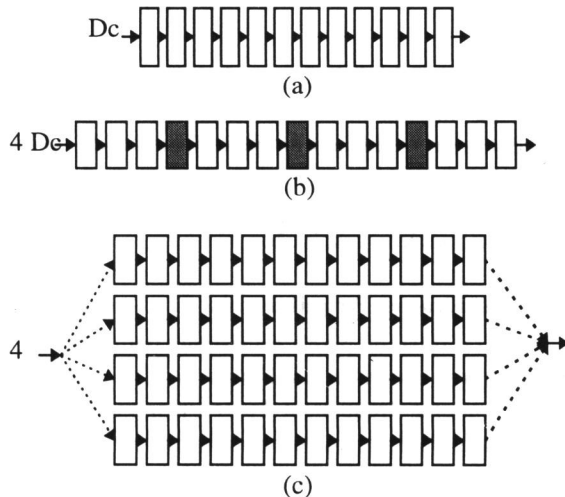


Figura 5: Pipeline e paralelização.

Seja D_C a taxa na qual o circuito calcula atualização de um jogo de parâmetros. Com a introdução de P barreiras de *pipeline*, aumenta-se a taxa D_C de um fator P . O aumento resultante para a área do circuito é:

$$S_{C(P)} = [1 + H_{C(P_i)}] S_C, \quad i = 1 \dots P \quad (6)$$

onde $H_C(P_i)$ é a altura do caminho crítico na posição de inserção da barreira de *pipeline*. Supondo caminhos críticos regulares e igualmente distribuídos pelo *pipeline*, a expressão acima equivale a:

$$S_{C(P)} = \left[1 + \frac{(P-1)}{T_C} \right] S_C \quad (7)$$

onde T_C é o tamanho do caminho crítico, isto é, o número máximo de somadores (ou equivalentes) a serem atravessados no circuito.

Em caso de paralelização, para o mesmo fator de aumento de taxa P , tem-se:

$$S_{C(P)} = P S_C \quad (8)$$

A vantagem do *pipeline* sobre a paralelização é evidente e a utilização dessa última só deve ser feita quando a aplicação de *pipeline* se torna impossível: por exemplo, se o valor de P necessário for superior a T_C .

Outra solução é a partição. Nesse caso, vários circuitos são utilizados. A aplicação dessa solução se justifica quando a área do circuito com *pipeline*/paralelização é tal que ela compromete o rendimento, isto é, quando o circuito se torna tão grande que os riscos de erros de fabricação a tornam proibitiva.

4 Condições de Simulação

Para avaliar a qualidade do algoritmo, as simulações foram realizadas com um objetivo de reamostragem temporal (ver Figura 6).

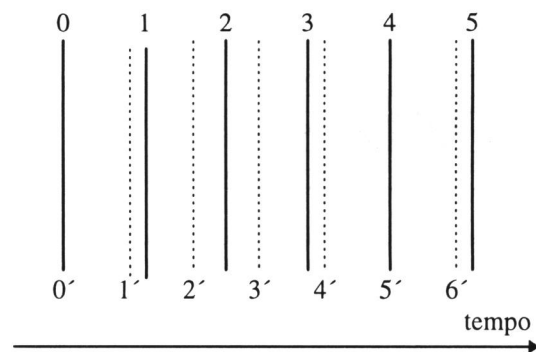


Figura 6: Interpolação temporal.

No caso de imagens em formato progressivo, a imagem ausente é reconstruída a partir das imagens precedente e subsequente: a imagem 2' é gerada a partir das imagens 1 e 2. Um formato entrelaçado na entrada implica em uma reconstrução (e estimação de movimentos) baseada em quadros de mesma paridade: o quadro 2' (quadro par) é gerado a partir dos quadros (pares) 0 e 2; o quadro 3' é gerado a partir dos quadros (ímpares) 1 e 3.

Fatores de interpolação não inteiros foram escolhidos a fim de obter várias posições relativas dos quadros/imagens a reconstruir com relação aos quadros/imagens existentes na seqüência.

Como critério objetivo de qualidade, escolheu-se o erro médio quadrático global da imagem. Duas condições foram consideradas para a detecção da divergência do algoritmo em um determinado nível da pirâmide:

- o erro é superior a um valor pré-definido no final das N iterações.
- o erro é superior a um valor pré-definido a qualquer uma das iterações.

Para a escolha do jogo de parâmetros a ser associado a um dado bloco, dois casos foram considerados:

- o jogo é aquele da última iteração.
- o jogo é aquele correspondente ao menor erro.

Incorporou-se a flexibilidade quanto ao tamanho dos blocos, que inicialmente era fixo e igual a 8x8 pixels. Mudou-se ainda a estratégia de associação dos jogos de parâmetros: em vez de se utilizar apenas o jogo de parâmetros resultante da segmentação dos blocos, utilizam-se também aqueles resultantes de estimativas sobre os blocos vizinhos.

A segmentação é feita para blocos de 16x16 pixels. Quatro níveis de resolução foram utilizados. No nível 0 (imagem com resolução completa), após todos os cálculos de jogos de parâmetros, os blocos são subdivididos, gerando blocos de 8x8 pixels, e uma redistribuição dos jogos é feita de acordo com o parágrafo precedente. As subdivisões e as redistribuições prosseguem até blocos de tamanho 2x2.

4.1 Seqüências Testadas

As seqüências testadas incluem « Mobile & Calendar », « Train », « Table Tennis », « Voiture » e « Manège ». Essas imagens correspondem à norma CCIR 4:2:2 (720x576 pixels ativos, 220 níveis de cinza, formato entrelaçado, freqüência de amostragem 13,5 Mhz).

A seqüência « Mobile & Calendar » apresenta fortes gradientes, movimentos pequenos do tipo translação e rotação. A seqüência « Train » se caracteriza por movimentos de grande amplitude. Casos de oclusão são bastante freqüentes na seqüência « Manège ».

As Figuras 7 e 8 ilustram os resultados obtidos para a seqüência « Table Tennis ». Nessa seqüência, podem ser identificados, além das translações, movimentos causados por homotetias e rotações.

4.2 Comentários

Durante as simulações, observou-se que o aumento do tamanho dos blocos implicava em uma diminuição da qualidade das imagens reconstruídas (critério subjetivo, visual). Isso porque os efeitos de bloco se tornavam mais visíveis. Por outro lado, a utilização de blocos muito pequenos tornava difícil a estimativa de movimentos de grande amplitude. Deve-se lembrar que o sentido « grande demais » ou « pequeno demais » é função do contexto e da atividade dos objetos nas seqüências.

A subdivisão dos blocos para a redistribuição mencionada na seção precedente levou a resultados substancialmente melhores, diminuindo os efeitos de bloco resultantes da segmentação.

As simulações feitas com as seqüências ditas « difíceis » evidenciaram a existência de problemas residuais. Esses problemas (inerentes aos métodos de estimativa de movimentos) são resultantes da hipótese inicial. O modelo não considera as variações temporais de luminância de um objeto, pois a única causa suposta para todas as variações de luminância é o movimento.

O processamento de aparecimento/occlusão de objetos exige um conhecimento do contexto da cena. A substituição de uma segmentação arbitrária em blocos por uma segmentação contextual deve conduzir à redução do número de vetores mal estimados. Trabalhos nesse sentido estão atualmente sendo realizados.

5 Conclusões

Neste artigo, apresentou-se um novo algoritmo de estimativa de movimentos. Baseado no método diferencial, esse algoritmo possibilita a estimativa de movimentos complexos tais como aqueles resultantes de rotação e *zoom*, além de translação. O esquema hierárquico com multi-resolução utilizado leva a um ganho em complexidade de cálculo. Os resultados obtidos usando esse algoritmo indicam a sua superioridade com relação a algoritmos do tipo *block-matching*, graças à minimização dos efeitos de bloco.

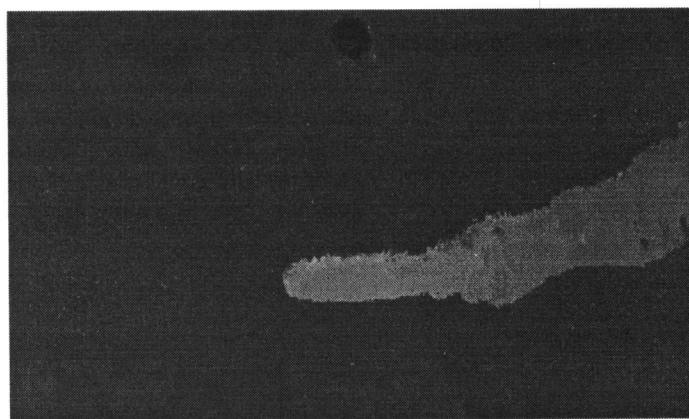
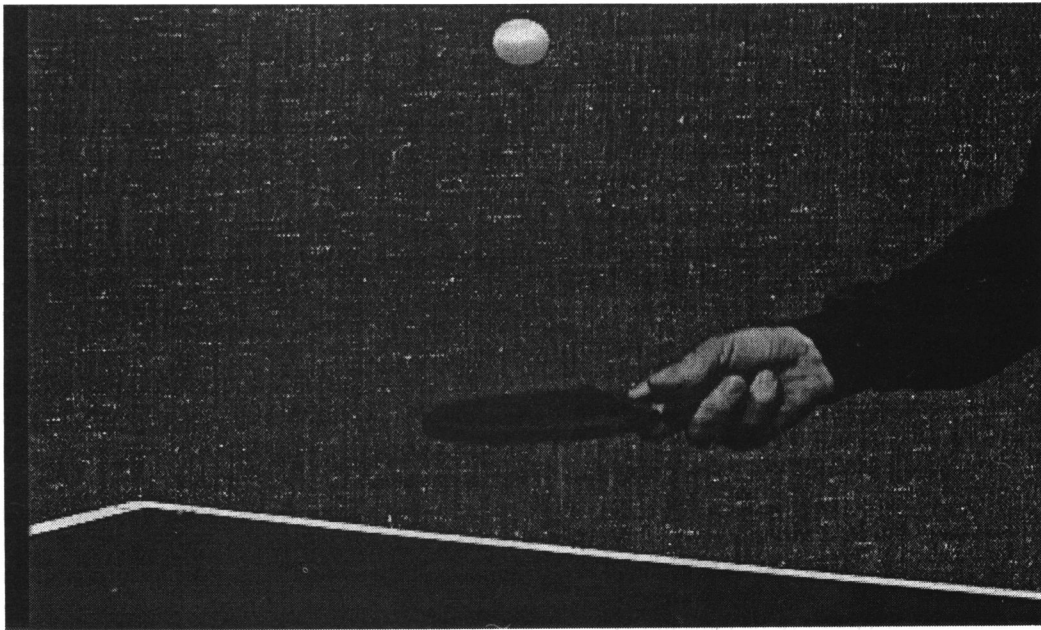


Figura 7: Resultados obtidos para a seqüência «Table Tennis» com o algoritmo apresentado. De cima para baixo: imagem interpolada, campo de vetores na direção x e campo de vetores na direção y.

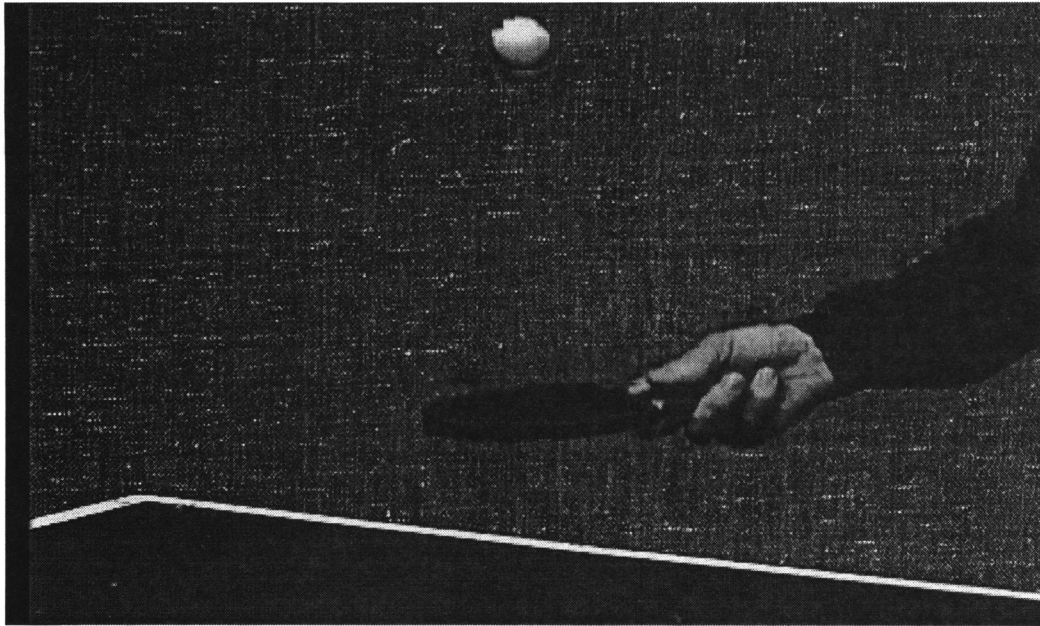


Figura 8: Seqüência «Table Tennis»: Imagem interpolada com o algoritmo block-matching

Referências

- L. A. Barros, Architecture Intégrée pour le Ré-échantillonnage d'Images Animées., Rapport Technique de l'Ecole Nationale Supérieure des Télécommunications, 1993.
- L. A. Barros, Real-time Estimation of object trajectories in image sequences, Proc. of the SIBGRAP'94, Curitiba, Brasil.
- H. C. Bergmann, Analysis of different displacement estimation algorithms for digital television signals, in Image Sequence Processing and Dynamic Scene Analysis, pp. 215-233, Ed. T.S. Huang, 1993.
- P. Bouthémy, Modèles et méthodes pour l'analyse du mouvement dans une séquence d'images, Technique et Science Informatiques, pp. 527-545, 1988.
- C. Cafforio and F. Rocca, Methods for measuring small displacements in television images, IEEE Trans. Inform. Theory, Vol. IT-22, pp. 573-579, Sept. 1976.
- B. Choquet, Estimation de mouvement et segmentation spatio-temporelle en séquence d'images. Contribution à la télévision haute définition: utilisation de l'information de mouvement dans un système HMAC, Thèse de Doctorat de l'Ecole Nationale Supérieure des Télécommunications, Paris, 1988.
- O. Colavin, A. Artieri, J-F. Naviner, and R. Pacalet, A dedicated circuit for real time motion estimation, Proceedings of the EUROASIC'91, pp. 2453-2456, 1991.
- E. Dubois and S. Sabri, Noise reduction in image sequences using motion-compensated temporal filtering," IEEE Trans. Commun., vol. COM-32, pp. 826-831, July 1984.
- W. Enkelmann, Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences, Comput. Vision, Graphics Image Proces., vol. 43, pp. 155-177, 1988.
- T. Komarek and P. Pirsch, Array architectures for block-matching algorithms, IEEE Trans. on Circuits and Systems, pp. 1301-1308, 1989.

- J. O Limb and J. A. Murphy, Measuring the speed of moving objects from television pictures, *IEEE Trans. Comm.*, Vol. COM-23, pp. 474-478, 1975.
- MPEG-4 Requirements Ad-Hoc Group, MPEG-4 Requirements Document - Revised First Draft, ISO/IEC JTC1/SC29/WG11 MPEG94, Jul. 1994.
- H. G. Musmann, P. Pirsch, and H-J Grallert, Advances in picture coding, *Proceedings of the IEEE*, pp. 523-548, 1985.
- A. N. Netravali and J. D. Robbins, Motion compensated television coding: part I, *The Bell System Technical Journal*, pp. 631-670, 1978.
- Recommendation H.262, ISO/IEC 13818, Generic Coding of Moving Pictures and Associated Audio, Draft International Standard of MPEG-2, 1991.
- H. Sanson, Motion affine models identification and application to television image coding, *Proceedings of SPIE Visual Communications and Image Processing'91*, pp. 570-581, 1991.
- R. Thoma and H. Bierling, Motion compensating interpolation considering covered and uncovered background, *Signal Processing: Image Communications*, Vol. 1, pp. 191-212, 1989.
- S. Wu and J. Kittler, 2D motion parameter estimation. a survey, *8th Scandinavian Conference on Image Analysis*, Oulum Finland, pp. 1043-1050, June, 1989.