

Compressão de Imagens por Blocos Unidimensionais
Lineares

Claudia Alvarenga

Luiz Velho

Globo Computação Gráfica
Rua J. Carlos, 101
Rio de Janeiro, RJ

ABSTRACT

Este trabalho apresenta uma técnica para a compressão de imagens digitais, que explora a coerência das linhas da imagem. O método utiliza uma abordagem geométrica do problema que pode ser interpretada como uma aproximação linear da função de intensidade da imagem. Esta aproximação é adaptativa e parametrizada, permitindo o melhor compromisso entre fidelidade da imagem e taxa de compressão. No limite, é possível uma codificação da imagem sem perda de informação. O algoritmo desenvolvido é bastante eficiente, e particularmente adequado à imagens geradas por computador.

1. Introdução.

A finalidade das técnicas de compressão de imagens digitais é reduzir o número total de bits empregados na sua representação [Jain 89]. Os algoritmos utilizados exploram, de uma forma ou de outra, a redundância intrínseca da informação contida nas imagens, considerando a correlação existente entre seus elementos [Netravalli 77].

Em geral, os métodos de compressão existentes podem ser classificados em duas categorias principais: estatísticos, e de transformação. Os primeiros procuram gerar um código mais compacto em função dos padrões mais prováveis dos dados [Huffman 52]. Os últimos procuram transformar a matriz de imagem de tal forma que esta seja mapeada em um número menor de elementos [Wintz 72].

O processo de compressão envolve basicamente as operações de mapeamento, quantização e codificação [Gonzalez 87]. O mapeamento transforma os elementos da imagem para um domínio que permita uma representação mais compacta. A quantização implica no truncamento do resultado da transformação para um conjunto discreto de valores. A codificação consiste na construção de uma representação física dos valores quantizados. As relações entre essas tres classes de operações determinam a categoria do método de compressão.

A compressão pode ou nao envolver uma perda de informação. Um método que preserva informação é chamado de reversível quando existe uma transformação inversa que reconstrói sem erro a imagem original. Métodos que nao atendem a este requisito são chamados de irreversíveis [Mitchell 79].

As técnicas de compressão de imagem são normalmente desenvolvidas de forma a privilegiar um determinado tipo de aplicação. Dessa forma, é possível atingir uma solução adaptada a problemas específicos. Alguns fatores a se considerar, além das características particulares dos tipos de imagem, incluem: fidelidade, rapidez de conversão, taxa de compressão, e etc.

Grande parte das técnicas de compressão pressupõem imagens digitalizadas de cenas reais. Essas imagens possuem características diferentes das imagens sintéticas, geradas por computador. Mais especificamente, as imagens sintéticas são produzidas digitalmente nao contendo ruído e outros componentes espúrios encontrados nas imagens captadas de cenas reais.

A técnica introduzida neste trabalho é particularmente adequada à compressão de imagens sintéticas, com aplicações diretas em sistemas de animação por computador e de produção de imagens de alta resolução. Essencialmente, o método consiste na segmentação da imagem em uma sequência de intervalos unidimensionais, ou "spans", que definem uma aproximação linear por partes da função de intensidade.

Este método pode ser considerado como uma extensão do "run-length encoding" [Paz 76], utilizando diferenças de intensidades. O "run-length encoding" é uma técnica de compressão bastante popular para armazenamento e compressão de imagens devido a facilidade de codificação e decodificação. Entretanto ela so é eficiente se a imagem tem grandes áreas de intensidade constante, o que nao ocorre em imagens sintéticas com tratamento de "anti-aliasing". Além de resolver este problema, a técnica apresentada oferece a possibilidade de parametrização da fidelidade da aproximação, permitindo taxas de compressão bastante elevadas.

2. Compressão por Aproximação Linear

Este método se baseia numa abordagem geometrica do problema de compressão de imagens. Dessa forma, a imagem sera interpretada como uma superficie tridimensional que passa por pontos cujas coordenadas são determinadas pela posição dos elementos da imagem e por sua intensidade. Sendo assim, a função de intensidade em cada ponto da imagem corresponde à elevação da superficie naquele ponto, formando uma superficie do tipo "Superficie de Monge", que é a generalização do grafico de uma função.

Para simplificar o problema iremos trabalhar numa dimensão inferior. Ou seja, a imagem sera decomposta nas linhas formadas por grupos de elementos paralelos a uma das dimensoes principais. Geometricamente, isso significa que vamos considerar as curvas resultantes da interseção da superficie com planos paralelos a um desses eixos principais.

A figura 1, exemplifica esses conceitos, mostrando a imagem de uma esfera, a superficie tridimensional correspondente, e o desenho da curva associada a uma linha que passa pelo seu centro.

A técnica proposta consiste fundamentalmente na determinação, em cada uma das linhas da imagem, dos intervalos que permitam a maior compressão da imagem usando uma aproximação linear por partes da função de intensidade.

Uma das vantagens dessa técnica esta na possibilidade de parametrização da taxa de compressão em função do grau de fidelidade desejado. No limite, a imagem pode ser codificada sem perda, considerando somente os intervalos nos quais a intensidade varia linearmente.

A solução deve minimizar o numero de intervalos para um determinado erro local entre a aproximação e a imagem original.

A configuração otima do grupo de intervalos que permite uma reconstrução linear por partes de uma função escalar unidimensional com o erro minimo, tem como geradora a "partição" definida por pares contiguos de "pontos interessantes". O conjunto dos pontos interessantes para funções continuas e diferenciaveis, é constituido pelos maximos e minimos locais, e pelos pontos de inflexão. Funções descontinuas são segmentadas nos pontos de descontinuidade, os quais serao tambem incluidos no conjunto dos pontos interessantes.

A partição inicial definida pelos pontos interessantes deve ser refinada até que o erro local esteja dentro da faixa de tolerancia definida pelo critério de fidelidade da

imagem.

Finalmente, os segmentos unidimensionais resultantes desse processo devem ser codificados de modo a representar a imagem comprimida, e permitir posterior reconstrução.

2.1. Mapeamento

O problema central, portanto, se reduz à identificação do conjunto de pontos interessantes, a partir dos quais a função de intensidade da imagem pode ser reconstruída dentro de uma tolerância pre-estabelecida.

Esse processo corresponde ao mapeamento da sequência de elementos das linhas da imagem (i_1), (i_2), ..., (i_N); na sequência de intervalos (vA_1, vB_1, l_1), (vA_2, vB_2, l_2), ..., (vA_M, vB_M, l_M), onde "i" denota intensidade da imagem, "v" o valor da função de intensidade nos extremos do intervalo e "l" a largura do intervalo.

Se considerarmos a função de intensidade como uma sequência de curvas contínuas e deriváveis em todos os pontos do seu intervalo de definição, a análise das primeiras e segundas derivadas em cada ponto nos permite definir o seu comportamento. Os máximos e mínimos locais definem a concavidade da curva e os pontos de inflexão definem a mudança de concavidade. A identificação desses pontos, os pontos interessantes, é feita através da mudança de sinal das primeiras e segundas derivadas. A estes acrescentamos os pontos de descontinuidades que correspondem às fronteiras de cada intervalo.

A figura 2 mostra o desenho de uma curva senoidal, de sua primeira e segunda derivadas, com os respectivos pontos interessantes. A figura 3 mostra a curva original sobreposta à aproximação linear correspondente à partição inicial.

2.2. Quantização

A quantização consiste na aproximação da curva de intensidade da imagem em cada intervalo da partição inicial por segmentos de reta secantes à curva, de modo que estes pertençam a uma determinada vizinhança tubular da curva.

Para isso, devemos considerar todos os pontos do intervalo e comparar os valores da aproximação linear com a função original, subdividindo sempre que o erro exceder a tolerância especificada.

Existem duas maneiras de realizar este processo: Incremental e Recursiva. A primeira tem a vantagem de ser mais

eficiente, mas a segunda proporciona uma subdivisão mais uniforme.

2.3. Codificação

Na etapa de codificação, os "spans" resultantes do processo de quantização devem ser representados por unidades de informação que possibilitem: 1) minimizar o número de bits usados para a imagem comprimida; 2) uma reconstrução eficiente da imagem original.

Basicamente, um "span" pode ser representado pela diferença relativa de intensidade e pelo tamanho do intervalo. O primeiro intervalo é um intervalo especial de tamanho 0 que indica a intensidade absoluta do elemento inicial da imagem.

Uma variável char (8 bits) é suficiente para armazenar a distância entre os pixels de um "span", já que permite armazenar distâncias de até 256 pixels. Mas numa imagem contendo texturas existem longas sequências de pixels que não permitem boa compressão.

Para evitar criar "spans" de apenas um pixel, é usado um bit desse char para especificar o tipo de informação que vem a seguir. Se o bit estiver ligado a informação contida no resto dos bits é a distância entre os pixels e o próximo char é o valor da diferença de intensidades. Se estiver desligado o restante do char contém o número de pixels a seguir, já que a variação não é constante. Desse modo são usados segmentos de no máximo 128 pixels de tamanho.

Alem disso, considerando que os valores de intensidade variam de 0 a 255, as diferenças podem variar entre -255 e 255, sendo necessários mais de oito bits para a sua representação. Por isso é usado mais um bit do char que especifica o tamanho do "span", indicando o sinal da diferença. Isso permite um tamanho máximo de "span" de 64 pixels.

2.4. Reconstrução

Na reconstrução da imagem original o processo de quantização utilizado pode introduzir um erro marginal na inversão da aproximação linear. Este erro, entretanto, não ultrapassa uma magnitude absoluta de 2 níveis de intensidade.

Uma análise completa desse problema, aplicada ao caso da rasterização de segmentos de reta pode ser encontrada em [Fiume 88].

3. Discretização da Imagem Original

Normalmente, imagens digitais originadas a partir de cenas reais apresentam uma série de problemas relativos ao processo de captação da imagem. Dentre eles destacamos a precisão limitada dos valores de intensidade causada pelo uso de códigos de 8 bits para sua representação. Embora este problema não seja intrínseco das imagens sintéticas, que em geral utilizam 12 bits para cada canal de intensidade, alguns sistemas de geração de imagem também representam a intensidade utilizando apenas 8 bits.

Quanto menor o número de bits utilizado para representar a intensidade da imagem, maior será o problema causado pela discretização. Este fato, introduz uma imprecisão na imagem original que se reflete, principalmente, como ruído nos valores da primeira e segunda diferenças.

Apesar desse ruído atrapalhar a identificação dos pontos interessantes, em especial dos pontos de inflexão, podemos observar os máximos e mínimos locais onde a primeira diferença muda de sinal e os pontos de inflexão onde a segunda diferença muda de sinal. Para isso são selecionados somente os pontos onde o valor absoluto das diferenças é maior que um delta especificado.

A figura 4 mostra uma esfera metálica branca com um reflexo, e o desenho de uma scanline central, com a primeira e segunda derivadas correspondentes.

4. Algoritmo de Compressão

A ideia essencial do algoritmo é localizar os pontos interessantes, verificando se as primeiras e segundas diferenças mudam de sinal. A partir desses pontos a curva pode ser reconstruída usando-se uma aproximação linear.

São localizados pontos adicionais de forma que todos os pontos selecionados definam segmentos onde a variação de valor é constante, dentro de uma tolerância preestabelecida.

O arquivo comprimido deve conter apenas o valor inicial e as diferenças a partir daí.

Segue a especificação do algoritmo em pseudo-código:

```

cria o span inicial correspondente ao primeiro elemento;

for (todos os elementos da imagem) {
    calcule a primeira diferenca neste ponto;
    calcule a segunda diferenca neste ponto;
    calcule o erro entre a aproximação e a curva;

    if (primeira diferenca mudou de sinal) {
        seleciona elemento;
    } else if (valor absoluto do erro > tolerancia) {
        seleciona elemento;
    } else if (segunda diferenca mudou de sinal e
        o seu valor absoluto > delta) {
        seleciona elemento;
    }
    if (elemento foi selecionado) {
        armazena a diferenca relativa de intensidade;
        armazena o numero de pixels do "span";
        inicializa o erro;
    }
}

```

5. Algoritmo de Descompressão

O processo de descompressão é bastante simples. A partir das diferenças relativas de intensidade entre uma sequência de elementos da imagem são calculados os valores dos pontos inicial e final de cada segmento.

Em seguida são gerados os valores intermediários através de um algoritmo incremental, semelhante aos de rasterização de retas. Foi utilizado o algoritmo de Bresenham modificado para este propósito.

Descrevemos abaixo o algoritmo em pseudo-código.

```

inicializa o valor de intensidade;

for (todos os "spans") {
    calcula incrementalmente
    a intensidade
    do elemento corrente;
}

```

6. Implementação

Os programas de compressão e descompressão foram implementados na linguagem C, no ambiente de programação do

sistema operacional UNIX. Os equipamentos utilizados incluem minicomputadores VAX 11/750 e Ridge 32C, além de frame buffers DeAnza e Raster Technologies.

Estes programas integram o sistema de animação tridimensional da Globograph.

7. Exemplos

As figuras 5, 6 e 7 mostram os resultados da compressão de tres imagens fotorrealistas geradas no sistema Globograph.

As subfiguras (a) são as imagens originais. As subfiguras (c) são imagens reconstruidas sem erro, com tolerancia residual minima de intensidade. As subfiguras (f) são imagens aproximadas com tolerancia de 10 unidades de intensidade.

O erro local entre as imagens (c) e (a) é mostrado nas subfiguras (d), e o erro local entre (f) e (a) é mostrado nas subfiguras (g). Esse erro é obtido escalando os valores minimo e maximo de maior variação nas imagens para 0 e 255.

As subfiguras (e) mostram os pontos interessantes das imagens(c) e as subfiguras (h) os pontos selecionados nas imagens (f). As subfiguras (b) mostram as partições iniciais das imagens.

8. Avaliação

As tabelas abaixo relacionam o fator de compressão, o erro global e o tempo de processamento para algumas imagens comprimidas por este método.

Pode-se notar que a taxa de compressão depende principalmente das características da imagem, dado o carater adaptativo do método. A diferenca entre o fator de compressão sem erro e a aproximação com tolerancia 5 nao é muito significativa.

Alem disso, o tempo de compressão se mostra praticamente constante para uma determinada complexidade de imagem. Em alguns casos, o tempo de descompressão é relativamente maior devido à dominancia da inicialização da parte incremental, quando existe um numero expressivo de segmentos pequenos.

FATOR DE COMPRESSAO
(bytes / pixel de 24 bits)

tolerancia	esfera	brahma	amx
1	1.22	2.00	2.07
5	1.04	1.86	2.00
10	1.01	1.82	1.97

ERRO GLOBAL
(RMS: root-mean-square)

tolerancia	esfera	brahma	amx
1	0.65	0.28	0.27
5	2.84	1.06	0.46
10	3.68	1.57	0.68

TEMPO DE PROCESSAMENTO DE COMPRESSAO
(segundos - Vax 11/750)

tolerancia	esfera	brahma	amx
1	26.9	45.6	48.6
5	25.6	45.8	47.2
10	25.7	46.0	47.4

TEMPO DE PROCESSAMENTO DE DESCOMPRESSAO
(segundos - Vax 11/750)

tolerancia	esfera	brahma	amx
1	18.3	66.1	47.9
5	17.6	61.8	46.7
10	18.0	61.2	48.2

9. Conclusões

O algoritmo apresentado se revelou uma técnica de compressão de imagens bastante flexível e eficiente.

Algumas modificações poderiam ser incluídas para aumentar ainda mais o fator de compressão, provavelmente implicando em maior tempo de processamento.

Um aspecto que pode ser otimizado é a codificação dos "spans". O uso de códigos mais sofisticados, baseados em técnicas estatísticas [Welch 84], permite usar menos bits para armazenar a mesma informação. Isto poderia ser utilizado também como pós-processamento, uma vez que se trata de técnica de uso geral.

Outra sugestão é introduzir técnicas reconstitutivas usadas em visão computacional [Geiger 89], para minimizar os problemas causados pela representação de 8 bits da imagem original.

Uma próxima etapa seria usar aproximações de mais alta ordem que permitem um mapeamento mais conciso para funções de intensidade com variação suave.

Finalmente, uma última extensão seria considerar o problema da reconstrução tridimensional da superfície correspondente a imagem, ou mesmo a correlação linha a linha dos pontos selecionados (coerência espacial).

10. Agradecimentos

Agradecemos a Globograph pelo apoio a esta pesquisa. Em particular, mencionamos Jose Dias, diretor da empresa; Rogerio Ponce e Sonia Barbosa da equipe de software; Carlinhos pela ajuda com as ilustrações; e Jonas Miranda por discussões férteis sobre o problema.

11. Refêrencias

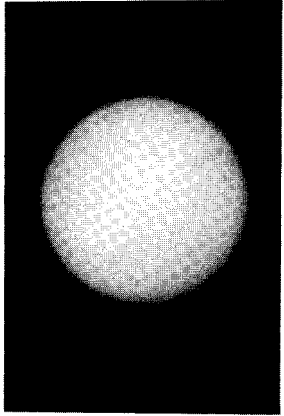
[Fiume 88]

E. Fiume, "Mathematical Concepts of Raster Graphics"
Springer Verlag, 1988.

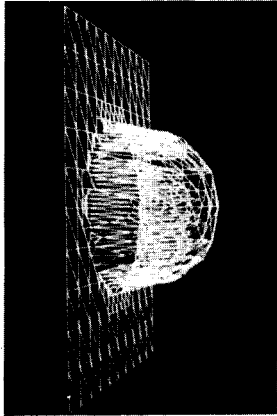
[Geiger 89]

D. Geiger and F. Girosi, "Parallel and Deterministic algorithms for MRFs: Surface Reconstruction and Integration",
AI Memo N. 1114, Massachusetts Institute of Technology,
June 1989.

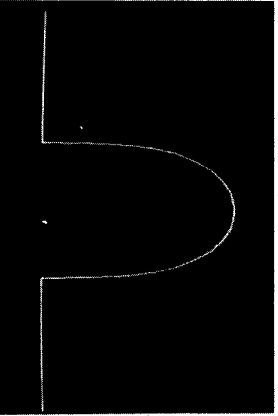
- [Gonzalez_87]
C. R. Gonzalez, "Digital Image Processing",
Addison-Wesley, 1987.
- [Huffman_52]
D. A. Huffman, "A Method for the Construction of
Minimum Redundancy Codes",
Proc. IRE, v. 40, n. 10, pp. 1098-1101,
1952
- [Jain_89]
A. K. Jain, "Fundamentals of Digital Image Processing",
Prentice Hall, 1989.
- [Mitchell_79]
E. J. Delp and O. R. Mitchell, "Image Compression using
Block Truncation Coding",
IEEE Trans. Commun., v. 27,
September 1979.
- [Netravali_77]
A. N. Netravalli, "On Quantizers for DPCM Coding of
Picture Signals",
IEEE Trans. Inform. Theory, IT-23, pp. 360-370,
May 1977.
- [Paz_76]
I. M. Paz, G. C. Collins and B. H. Batson, "A Tri-state
Delta Modulator for Run Length Encoding of Video",
Proc. National. Telecomm. Conf., Dallas, Texas, v. I,
pp 6.31-6.36,
November 1976.
- [Welch_84]
T. A. Welch, "A Technique for High Performance Data
Compression", IEEE Computer, pp. 8-19.
June 1984.
- [Wintz_72]
P. A. Wintz, "Transform Picture Coding",
Proc. IEEE, v. 60, n. 7, pp. 809-820,
July 1972.



A

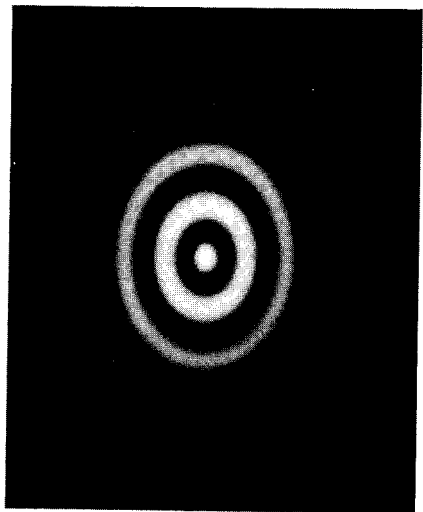


B

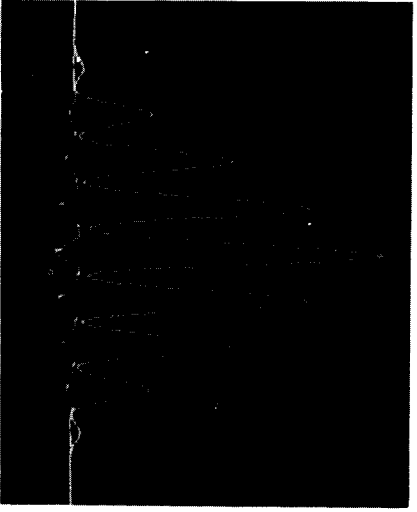


C

Figura 1



A



B

Figura 2

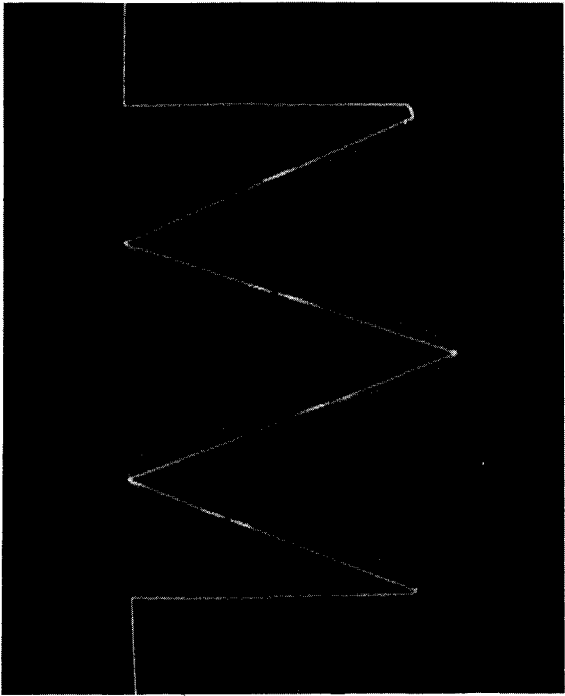


FIGURA 3

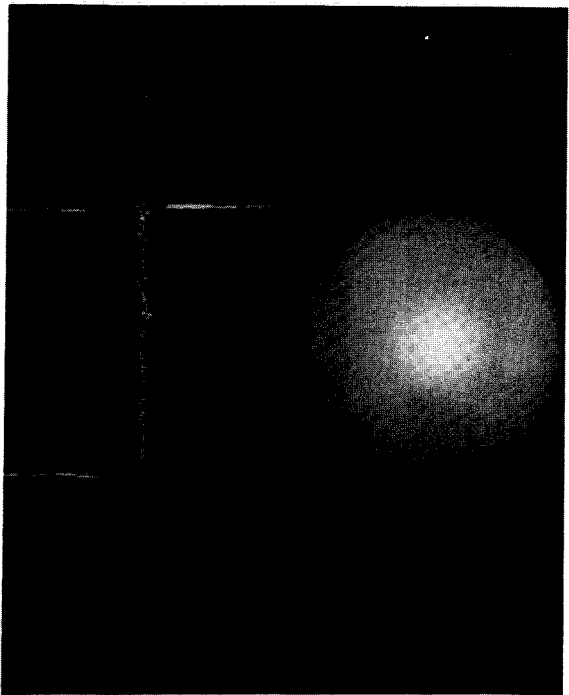
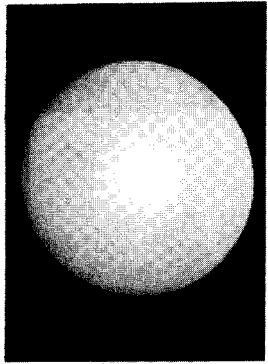
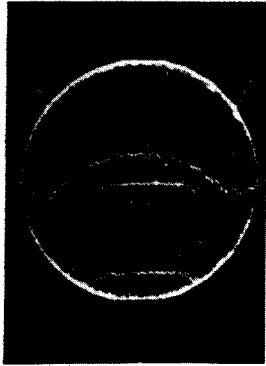


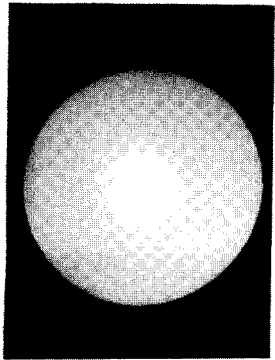
FIGURA 4



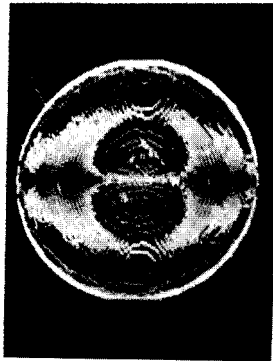
A



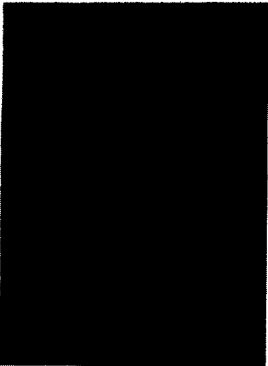
B



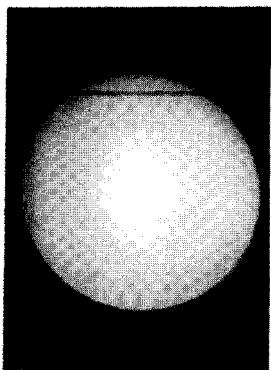
C



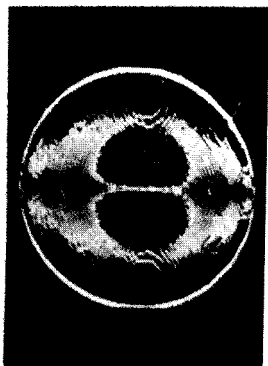
D



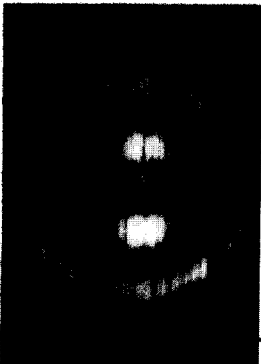
E



F



G



H

Figura 5

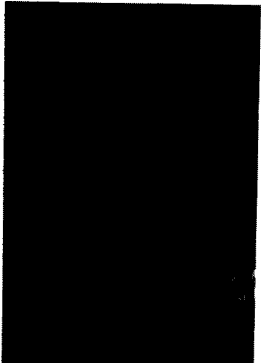
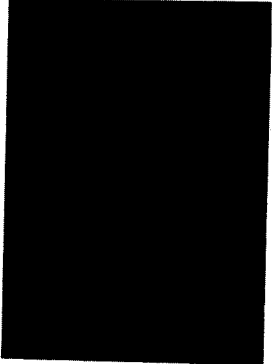
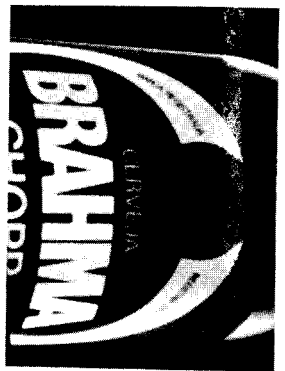
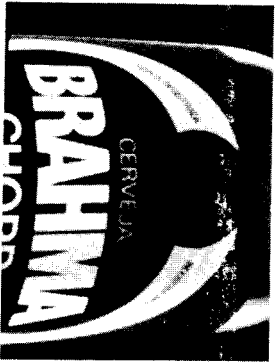
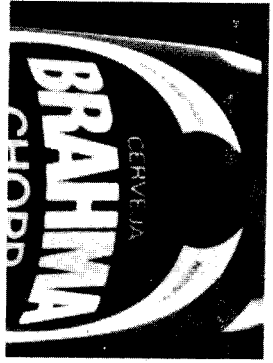
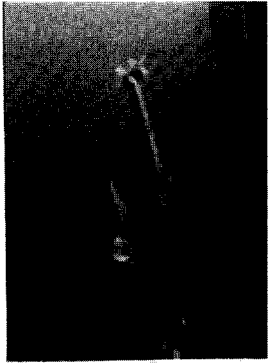
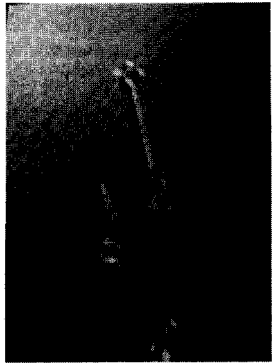


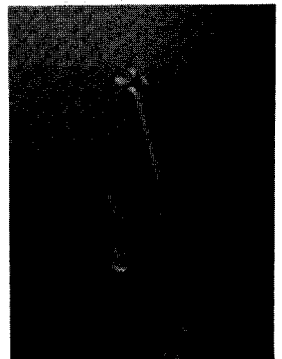
Figura 6



A



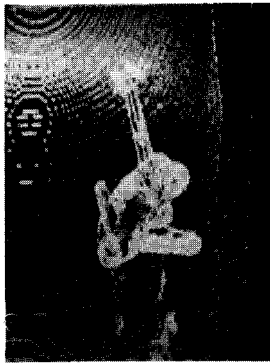
C



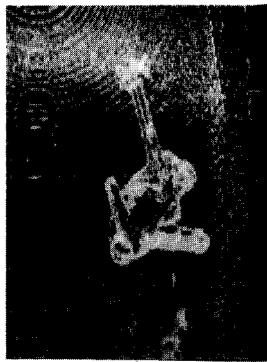
E



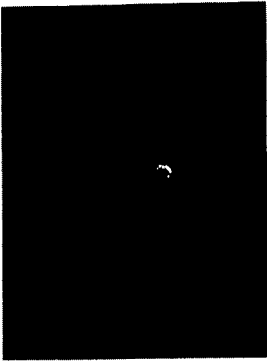
B



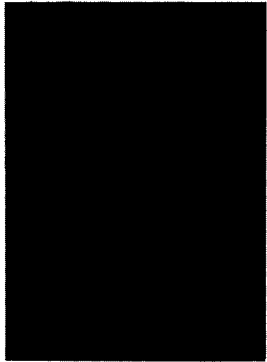
D



G



E



H

FIGURA 7