

Algoritmo Segmentador de Texturas

por

Alexandre Gonçalves Pinheiro

DISSERTAÇÃO SUBMETIDA À COORDENAÇÃO DO CURSO
DE PÓS-GRADUAÇÃO EM FÍSICA, COMO REQUISITO PARCIAL
PARA OBTENÇÃO DO GRAU DE

MESTRE

Departamento de Física

Universidade Federal do Ceará

Abril de 1997

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

P718a Pinheiro, Alexandre Gonçalves.

Algoritmo segmentador de texturas / Alexandre Gonçalves Pinheiro. – 1997.

61 f. : il.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Física, Fortaleza, 1997.

Orientação: Prof. Dr. Murilo Pereira de Almeida.

1. Algoritmos . 2. Imagens digitais. I. Título.

CDD 530

UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE FÍSICA

ALGORITMO PARA SEGMENTAR IMAGENS COM TEXTURAS

Alexandre Gonçalves Pinheiro

Dissertação apresentada ao Curso de Pós-Graduação em Física da Universidade Federal do Ceará como parte dos requisitos para obtenção do grau de Mestre em Física

Comissão Julgadora:

Dr. Murilo Pereira de Almeida - UFC

Dr. José Ricardo de Almeida Torreão - UFC

Dr. Paulo César Cortez - UFC

Fortaleza, Abril de 1997

Abstract

We present an algorithm to segment textured digital images. We use an approach to computer vision based in Markov Random Field (MRF) models. The model breaks the image into small windows (regions) that captures the pattern of the textures, and estimates their corresponding parameters. This is done by solving a system of linear equations built from the image gray levels. A two level MRF is used, one modelling the textures in the image itself, and the other corresponding to the label configuration assigned to the regions. A “cooling” schedule, based in the Simulated Annealing is employed to reach the label configuration that segments the image. A set of experiments is performed using real scenes textures.

Resumo

Neste Trabalho é desenvolvido um algoritmo para segmentar imagens digitais texturizadas. É utilizada uma abordagem de visão computacional baseada em modelo de Campo Aleatório de Markov (CAM). A imagem é dividida em blocos contendo texturas, cujos parâmetros, associados a um modelo de CAM, são estimados através de um sistema de equações lineares construído a partir das intensidades dos pixels. Um modelo de duas camadas de CAM é empregado, o primeiro modelando a textura e o segundo representando as configurações de rótulos dos blocos que dividem a imagem total. É empregado o esquema de "Annealing" simulado para obtenção da configuração de rótulos que segmenta a imagem. Um conjunto de experimentos é realizado com texturas de cenas reais.

Agradecimentos

Ao prof. Murilo Pereira de Almeida pela orientação desprendida com dedicação, compreensão e paciência. Ao prof. Josué Mendes Filho, pela colaboração e incentivo. Ao chefe do depto. de Física prof. José Carlos Parente de Oliveira pelo apoio concedido. Aos estudantes: Antonio Carlos, Gerson, Arnaldo, Leonaldo, Ivoneide, Bosco, Kleiton e Carlos Jacinto pelos anílios prestados. A todos os amigos do Laboratório de Física de Nuvens e Mesoescala (LFNM), pela ajuda fornecida. A todos os colegas do curso de Física da UFC. A todos os funcionários e professores do depto. de Física da UFC. À banca examinadora. A FUNCEME, na sua colaboração para com o laboratório (LFNM). Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), pelo apoio financeiro infalível.

Ao Brasil, e sobretudo A Deus.

Conteúdo

Abstract	iii
Resumo	iv
Agradecimentos	v
Lista de Figuras	viii
Lista de Tabelas	ix
1 Introdução	1
2 Campo Aleatório de Markov	6
2.1 Estimadores Variacionais para Campos Aleatórios de Markov.	9
3 Dois modelos e algoritmos para segmentar texturas	11
3.1 Segmentação de texturas pelos filtros dos mínimos quadrados.	11
3.2 Detecção de bordas por otimização vinculada.	15
3.2.1 A — Partição.	20
3.2.2 B — Interação rótulo-dado.	21

C — Medidas de disparidade.	21
3.2.4 D — Penalidades.	22
4 Algoritmo para segmentação de texturas modeladas por Campos Aleatórios de Markov	24
Introdução	24
4.2 Modelo de Partição	26
4.2.1 Interação Imagem-Rótulo.	27
4.3 "Annealing" Simulado	29
5 Experimentos	31
6 Conclusões	37
Bibliografia	58

Lista de Figuras

Imagem de RADAR	4
Cliques usados	9
Grade de pixels	17
Bloco associado a s	21
4.1 Cliques usados	28
Imagem 1	32
5.2 Segmentação da imagem 1 ($\sigma = 32, P = 5, \epsilon = 12.44, T = 3.0$)	33
5.3 Imagem 2	34
Segmentação da imagem 2 ($\sigma = 32, P = 5, \epsilon = 666.0, T = 3.0$)	34
Imagem 3	35
Segmentação da imagem 3 ($\sigma = 32, P = 5, \epsilon = 90.0, T = 3.0$)	35
5.7 Imagem 4	36
Segmentação da imagem 4 ($\sigma = 32, P = 5, \epsilon = 45.66, T = 3.0$)	36

Lista de Tabelas

6.1	Parâmetros da textura 1 da figura 5.1 com $\sigma = 32$	38
6.2	Parâmetros da textura 2 da figura 5.1 com $\sigma = 32$	39
6.3	Parâmetros da textura 3 da figura 5.1 com $\sigma = 32$	40
6.4	Parâmetros da textura 4 da figura 5.1 com $\sigma = 32$	
6.5	Parâmetros das 4 texturas da figura 5.1 com $\sigma = 128$	
6.6	Parâmetros da textura 1 da figura 5.3 com $\sigma = 32$	43
6.7	Parâmetros da textura 2 da figura 5.3 com $\sigma = 32$	44
6.8	Parâmetros da textura 3 da figura 5.3 com $\sigma = 32$	45
6.9	Parâmetros da textura 4 da figura 5.3 com $\sigma = 32$	
6.10	Parâmetros das 4 texturas da figura 5.3 com $\sigma = 96$	
6.11	Parâmetros da textura 1 da figura 5.5 com $\sigma = 32$	48
6.12	Parâmetros da textura 2 da figura 5.5 com $\sigma = 32$	49
6.13	Parâmetros da textura 3 da figura 5.5 com $\sigma = 32$	50
6.14	Parâmetros da textura 4 da figura 5.5 com $\sigma = 32$	51
6.15	Parâmetros das 4 texturas da figura 5.5 com $\sigma = 96$	52
6.16	Parâmetros da textura 1 da figura 5.7 com $\sigma = 32$	53

6.17 Parâmetros da textura 2 da figura 5.7 com $\sigma = 32$

6.18 Parâmetros da textura 3 da figura 5.7 com $\sigma = 32$

6.19 Parâmetros da textura 4 da figura 5.7 com $\sigma = 32$.

6.20 Parâmetros das 4 texturas da figura 5.7 com $\sigma = 96$

Capítulo 1

Introdução

Visão computacional é o ramo da ciência computacional que estuda os métodos e modelos para representar, analisar e interpretar imagens digitalizadas. Este ramo só se tornou possível devido ao desenvolvimento de três áreas tecnológicas relacionadas com: (a) dispositivos sensores ópticos; (b) computadores digitais e (c) software aplicados a processamento de imagens.

De um modo geral, a informação (imagem) é captada pelo sensor que a converte em um conjunto de valores ou intensidades correspondendo cada um a uma pequena região da cena observada. A imagem resultante tem valores de intensidades que permanecem constantes sobre os componentes elementares da imagem (pixels); ademais estas intensidades são condicionadas a tomar valores num conjunto finito (o mais comum sendo o subconjunto dos inteiros, com valores entre 0 e 255). Em algumas aplicações medem-se outras formas de luz, como o infra-vermelho, cuja intensidade está relacionada com a temperatura do corpo que a emite. Imagens de infra-vermelho captadas por sensores colocados em satélites meteorológicos têm grande valor para os meteorologistas para a determinação dos campos de

temperatura da atmosfera.

Outra área com vastas aplicações de visão computacional é a medicina, na qual a informação pode ou não ser observada em forma de luz visível. Exemplos de outros tipos de fontes seriam os raios X, ondas de ultrassom, emissão radioativa de substâncias injetadas no paciente sob exame, etc.

Na área industrial, visão computacional é utilizada para inspeção automática de materiais e componentes, robótica, soldas, etc.

Uma vez captada através do sensor apropriado a imagem digitalizada é então manipulada por algoritmos especialmente desenvolvidos para tratá-la e interpretá-la. Estes algoritmos demandam, em geral, grande quantidade de operações matemáticas, sendo portanto fundamental a utilização de computadores digitais com grande velocidade de processamento. Além disso, o computador deve possuir grande capacidade de armazenamento de dados para poder guardar essas imagens: por exemplo, uma única imagem multi-espectral do Landsat pode conter 30Mbytes de dados. Isto mostra a dependência de visão computacional com relação ao progresso na área de desenvolvimento de "hardware".

A grande variedade de aplicações de visão computacional gerou uma quantidade também grande de algoritmos para o tratamento de imagens. A maioria destes algoritmos se baseia em idéias puramente heurísticas, particulares para cada problema. No entanto emergiu uma corrente de idéias baseadas em Campos Aleatórios de Markov (CAM) para a modelagem e tratamento de imagens digitais. Esta abordagem permite a unificação dos princípios teóricos para a modelagem e processamento de imagens formando a base para uma formulação coerente dos diversos problemas da área [5, 11, 18, 22].

Os Campos Aleatórios de Markov (CAM) têm desempenhado importante papel na des-

criação de sistemas de mecânica estatística [25]. Recentemente eles foram introduzidos em processamento de imagens e visão computacional [18]. Eles são também usados em redes neurais e inferência da percepção [1] e reconhecimento de voz [24]. O esquema apresentado por Geman & Geman [18] tem sido testado com sucesso numa variedade de aplicações de processamento de imagens [14, 15], tomografia computadorizada [16, 17], etc.

As diversas tarefas de visão computacional podem ser classificadas em dois grupos gerais: (a) tarefas de baixo nível e (b) tarefas de alto nível. As tarefas de baixo nível incluem reconstrução de imagem, remoção de ruídos e/ou borramentos, detecção de fronteiras, segmentação, estimação de orientação tridimensional e profundidade, análise de movimento, etc. Estas tarefas constituem o que se chama de análise de imagem. Outra tarefa de baixo nível é a síntese de imagens. As tarefas de alto nível são as que lidam com a interpretação semântica da informação contida na imagem. O reconhecimento é a interpretação das características bi ou tridimensionais baseada em imagens bidimensionais.

Um aspecto importante nas imagens digitais é a textura de suas regiões. Uma textura pode ser interpretada como uma ordenação de padrões ou sub-padrões, como um tabuleiro de xadrez, por exemplo. Tais texturas, que apresentam um primitivo bem definido com uma também bem definida regra de colocação são denominadas *Determinísticas*. Existe ainda o tipo de textura que simplesmente obedece a regras estatísticas; estas são as texturas *Estocásticas*.

Dois elementos principais da textura são sua *Granulação* e sua *Direção*. A granulação está ligada à flutuação dos valores da intensidade em pixels vizinhos, enquanto que a direção está ligada à existência de regiões com direções prioritárias de alinhamento.

Vários algoritmos para a análise de texturas foram propostos ao longo dos anos e muitos

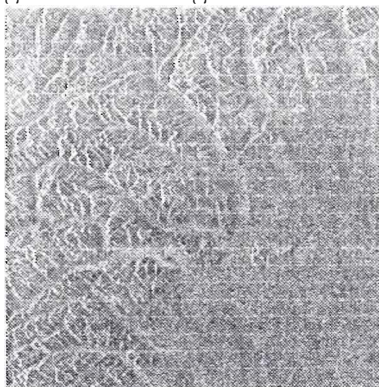
destes (os de antes de 1983) são discutidos em Van Gool et al. [27].

Os algoritmos para a análise de textura podem ser classificados em dois grupos: (a) estruturais e (b) estatísticos. Os algoritmos estruturais são usados para texturas determinísticas [10]. Os algoritmos estatísticos são utilizados para analisar texturas estocásticas. Dentre os algoritmos estatísticos existem aqueles que são baseados em modelos e dos quais ressaltamos os que utilizam modelos de Campos Aleatórios de Markov (CAM).

Do ponto de vista de representar, gerar e analisar texturas com modelos de CAM um problema básico é o da estimação de parâmetros a partir dos dados, que podem ser tanto completos ou incompletos (com ruído). Este problema é eficientemente resolvido pela utilização de estimadores variacionais [3, 4]. A grande demanda computacional dos estimadores clássicos (máxima verossimilhança e pseudo máxima verossimilhança) era o grande obstáculo ao emprego desses modelos na prática.

A figura 1.1 mostra uma típica imagem obtida por radar SAR (synthetic aperture radar) em satélite. Observa-se que a imagem tem oscilações nos níveis de cinza e que os padrões de texturas são aleatórios.

Figura 1.1: Imagem de RADAR



A presente dissertação tem por objetivo desenvolver um algoritmo para segmentar ima-

gens texturizadas. O algoritmo aqui proposto é baseado no trabalho de Geman et al. [15], incorporando algumas alterações no tocante à medida que diferencia duas texturas em imagens finitas. Para isto, adota-se uma abordagem de Campo Aleatório de Markov (CAM) para modelar as texturas e um algoritmo para estimar os parâmetros a elas associados, ambos proposto por Almeida [3].

A presente abordagem tem a vantagem de substituir a medida de disparidade apresentada por Geman et al. [15], que necessita de várias estatísticas espaciais para caracterizar a textura, por um único vetor de parâmetros que incorpora informações sobre os níveis de cinza e suas distribuições espaciais na textura.

No Capítulo 2, definem-se os CAM (Campos Aleatórios de Markov), que são usados para modelar texturas, e os estimadores variacionais, que são usados para estimar os parâmetros associados às mesmas. No Capítulo 3 são apresentados dois métodos para segmentação de textura, um probabilístico, que usa dois níveis de CAM [15] e outro determinístico que usa a técnica de filtro de mínimo quadrado [10]. Os Capítulos 4 e 5 são sobre o modelo proposto nesta dissertação e o consequente algoritmo, e ainda contém os resultados numéricos dos experimentos, incluindo as aplicações e conclusões.

Capítulo 2

Campo Aleatório de Markov

Primeiramente sejam estabelecidas algumas notações. Seja Z o conjunto de todos os inteiros, d um inteiro finito e Λ um sub-conjunto finito da grade Z^d .

Seja denotado por um Campo Aleatório (CA), um processo estocástico X_t , com $t \in \Lambda$, $\Lambda \subseteq Z^d$, i.e., um CA é um processo aleatório, cujo conjunto de indexação é um conjunto discreto multi-dimensional. Considera-se que este processo aleatório toma valores no espaço Ω_o . Pode-se considerar que $\{X_t : t \in \Lambda\}$ é uma função aleatória em algum espaço de probabilidades, tomando valores no espaço de configurações $\Omega_o^\Lambda = \{x_\Lambda : \Lambda \rightarrow \Omega_o\}$ onde $x_\Lambda = \{x_t : t \in \Lambda\}$ é uma das configurações.

Aqui letras maiúsculas X, Y, \dots representam variáveis aleatórias e as minúsculas x, y, \dots as suas particulares realizações.

Um campo Aleatório é dito *Campo Aleatório de Markov* quando, para cada t , a probabilidade condicional de X_t dada toda a configuração restante é igual a probabilidade condicional de X_t dado os valores de X na vizinhança de t , i.e.,

$$P(dX_t | \{X_i\}_{i \neq t}) = P(dX_t | \{X_i\}_{i \in N_t}) \quad (2.1)$$

onde N_t é o conjunto de sítios vizinhos a t .

Os trabalhos de Dobrushin [11] e Ruelle e Lanford [22] estabeleceram as idéias básicas para o desenvolvimento de CAM. Besag [5] apresentou uma prova do teorema de Hammersley-Clifford, que estabelece a equivalência entre CAM e a distribuição de Gibbs. Uma versão generalizada deste teorema é provada por Georgii [19].

Condições suficientes para esta equivalência são satisfeitas pelo modelo de textura aqui empregado [3, 4], que é especificado por distribuições de Gibbs, escritas em termos de potenciais definidos num sistema de vizinhança específica.

Os Campos Aleatórios de Markov são generalizações de Cadeias de Markov quando tomamos o conjunto de índices com dimensão maior que um. Existem problemas não triviais nestas generalizações e o primeiro se refere a definição dos vizinhos de um certo sítio. Em uma dimensão, dado um tempo presente t , seus vizinhos são prontamente determinados por $t - 1$ (passado imediato) e $t + 1$ (futuro imediato). Em altas dimensões, a idéia de passado e futuro já não se aplica e os vizinhos de um sítio devem ser especificados explicitamente construindo-se um sistema de vizinhança $N = \{N_i : i \in A\}$, onde para cada $i \in A$, N_i é um subconjunto de A definindo os pixels vizinhos de i . O par (A, N) constitui um gráfico. O sistema de vizinhança é dito simétrico quando $i \in N_j$ se e somente se $j \in N_i$. Um subconjunto $c \subset A$ é dito um clique quando quaisquer dois pixels de c são vizinhos um do outro [18].

A maneira prática de se construir CAM é através da especificação de sua probabilidade condicional (Dobrushin [11]) na forma de distribuição de Gibbs dada por:

$$\Pi_\Lambda(X_\Lambda|X_{\Lambda^c}) = \frac{\exp(-H_\Lambda(X_\Lambda|X_{\Lambda^c}))}{Z_{\Lambda, X_{\Lambda^c}}} \quad (2.2)$$

onde a constante de normalização $Z_{\Lambda, X_{\Lambda^c}}$ é a função partição, e $H_\Lambda(X_\Lambda|X_{\Lambda^c})$ é a função energia (Hamiltoniano). Ficou provado por Lanford [22] que quando H_Λ satisfaz as condições de regularidade, a família de probabilidades condicionais Π_Λ convergem fracamente a distribuições de probabilidade à medida que $\Lambda \rightarrow Z^d$.

Nos casos aqui estudados não se está interessado em tal limite, pois a análise deve ser concentrada em configurações de volumes finitos e suas respectivas distribuições de probabilidades. A abrangência da teoria de CAM sobre todo o Z^d é mais geral do que o necessário para a representação de imagens digitais.

Usa-se um sistema de vizinhança homogênea (exceto para as bordas) composta de N diferentes tipos de cliques (figura 2.1), cada um formado por dois pixels. Para cada clique do tipo α existe uma energia associada $U_\Lambda^{(\alpha)}$

$$U_\Lambda^{(\alpha)}(X_\Lambda) = \sum_{\substack{c \subseteq \Lambda \\ c \in C_\alpha}} \Phi_c(X_\Lambda) \quad (2.3)$$

onde C_α é o conjunto de todos os cliques do tipo α e o Φ_c é a função potencial do clique c . O Hamiltoniano específico usado no modelo de textura do presente trabalho é dado por:

$$H_\Lambda(x_\Lambda) = \sum_{\alpha=1}^N \theta^{(\alpha)} U_\Lambda^{(\alpha)}(X_\Lambda) \quad (2.4)$$

onde $\theta^{(\alpha)}$ é o peso da energia $U_\Lambda^{(\alpha)}$ e os $U_\Lambda^{(\alpha)}$'s são dados por:

$$U_\Lambda^{(1)}(x_\Lambda) = \sum_{i \in \Lambda} x_i^4, \quad (2.5)$$

$$U_\Lambda^{(2)}(x_\Lambda) = -\frac{1}{2} \sum_{i \in \Lambda} x_i^2, \quad (2.6)$$

$$U_A^{(3)}(x_\Lambda) = - \sum_{i \in \Lambda} x_i, \quad (2.7)$$

$$U_A^{(4)}(x_\Lambda) = \frac{1}{2} \sum_{\{i, j\} \in \Lambda} (X_i - X_{i, f_1})^2, \quad (2.8)$$

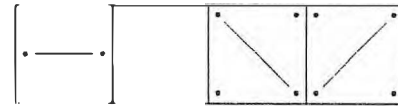
$$U_A^{(5)}(x_\Lambda) = \frac{1}{2} \sum_{\{i, j\} \in \Lambda} (X_i - X_{i, f_2})^2, \quad (2.9)$$

$$U_A^{(6)}(x_\Lambda) = \frac{1}{2} \sum_{\{i, j\} \in \Lambda} (X_i - X_{i, f_3})^2$$

$$U_A^{(7)}(x_\Lambda) = \frac{1}{2} \sum_{\{i, j\} \in \Lambda} (X_i - X_{i, f_4})^2.$$

com o parâmetro $\vec{\theta} = (\lambda, A, h, \beta_1, \beta_2, \beta_3, \beta_4)^T$, e os vetores $f_1 = [1, 0]$, $f_2 = [0, 1]$, $f_3 = [1, 1]$ e $f_4 = [1, -1]$.

Figura 2.1: Cliques usados



2.1 Estimadores Variacionais para Campos Aleatórios de Markov.

Considera-se que a imagem observada x_Λ , é uma configuração de volume finito de uma realização do CAM que tem Hamiltonianos de volumes finitos da forma da equação (2.4). Deseja-se estimar os valores dos parâmetros $\theta^{(\alpha)}$, $\alpha = 1, \dots, N$ associados ao CAM usando a imagem observada.

Os detalhes do algoritmo dos estimadores variacionais são apresentados em Almeida [3, 4]. Estes estimadores são deduzidos para CAM com “spins” contínuos mas produzem também bons resultados para CAM com “spins” discretos. A idéia básica está embutida na escolha de

N estatísticas vetoriais $W^{(\alpha)}(x_\Lambda) = (W_1^{(\alpha)}(x_\Lambda), \dots, W_n^{(\alpha)}(x_\Lambda))$, $\alpha = 1, \dots, N$, onde $n = |\Lambda|$,

tais que

$$\int_{\mathcal{R}^n} \nabla \cdot (\vec{W}^{(\alpha)}(x_\Lambda)) \Pi_\Lambda(dx_\Lambda) = 0, \quad \alpha = 1, \dots, N. \quad (2.11)$$

Pelo teorema da divergência, (2.11) acima é uma identidade desde que os $\vec{W}^{(\alpha)}$'s sejam escolhidos de modo que a integral de superfície no infinito seja nula. Usando esta condição,

(2.11) pode ser escrita como

$$\sum_{j=1}^N \theta^{(j)} \int_{\mathcal{R}^n} \{ \vec{W}^{(\alpha)} \cdot \nabla U^{(j)}(x_\Lambda) \} \Pi_\Lambda(dx_\Lambda) = \int_{\mathcal{R}^n} \{ \nabla \cdot \vec{W}^{(\alpha)} \} \Pi_\Lambda(dx_\Lambda), \quad (2.12)$$

para $\alpha = 1, \dots, N$. Substituindo as integrais acima em (2.12) pelos seus valores empíricos obtém-se o sistema linear de equações algébricas para os parâmetros $\theta^{(\alpha)}$, $\alpha = 1, \dots, N$.

Em particular, tomando-se

$$\vec{W}^{(\alpha)}(x_\Lambda) = \nabla U_\Lambda^{(\alpha)}(x_\Lambda), \quad \alpha = 1, \dots, N, \quad (2.13)$$

chega-se ao sistema

$$A_{\alpha\gamma}(x_\Lambda) \hat{\theta}^{(\alpha)}(x_\Lambda) = v_\gamma(x_\Lambda),$$

onde $\gamma = 1, \dots, N$ e

$$A_{\alpha\gamma}(x_\Lambda) = \sum_{i \in \Lambda} \left(\frac{\partial U_\Lambda^{(\alpha)}}{\partial X_i}(x_\Lambda) \frac{\partial U_\Lambda^{(\gamma)}}{\partial X_i}(x_\Lambda) \right), \quad (2.15)$$

e

$$v_\gamma(x_\Lambda) = \sum_{i \in \Lambda} \frac{\partial^2 U_\Lambda^{(\alpha)}}{\partial X_i^2}(x_\Lambda). \quad (2.16)$$

Capítulo 3

Dois modelos e algoritmos para segmentar texturas

Neste capítulo são apresentados como exemplos duas técnicas de segmentação de textura, sendo uma determinística e outra estocástica. O objetivo do capítulo é mostrar que existem várias abordagens possíveis para o problema de segmentação e também apresentar o modelo estocástico do Genan [15] que serve de base para o algoritmo que é implementado nesta dissertação.

3.1 Segmentação de texturas pelos filtros dos mínimos quadrados.

O trabalho de Hexin Chen e Yisong Dai [10] apresenta uma aplicação da segmentação de texturas, usando imagens testes como guia.

O principal enfoque, nas técnicas de classificação, ou segmentação de imagens com tex-

turas, é desenvolver meios de se detectar a ocorrência de uma regularidade (textura) em uma imagem ou estrutura. Aqui serão vistos os filtros dos mínimos quadrados (FMQ), um método para análise de texturas que está relacionado com a derivação de características texturais, onde os filtros específicos são derivados da matriz de auto-correlação da textura em conjunto com o vetor de correlação cruzada entre a imagem original e cada textura de fundo.

Seja a imagem original (texturizada) $y(i, j)$, $0 \leq i \leq N - 1$, $0 \leq j \leq M - 1$, composta de r tipos de texturas, $f_1(i_m, j_m)$, $f_2(i_m, j_m)$, \dots , $f_r(i_m, j_m)$, onde as variáveis (i, j) pertencem ao conjunto $\Omega \subset Z^2$ e (i_m, j_m) pertencem a Ω_m , ($m = 0, 1, \dots, r$), de modo que $\Omega = \cup_{m=0}^r \Omega_m$ $\forall p, m, p \neq m, \Omega_p \cap \Omega_m = \emptyset$.

O objetivo é achar um conjunto de r filtros $h_m(k, l)$ ($1 \leq k, l \leq n$), com janela de tamanho $n \times n$ ($n \ll N, n \ll M$), cada um dos filtros correspondendo a $y_m(i, j)$ (imagem total composta pela m -ésima textura).

Para se determinar estes filtros é necessário montar uma função, conforme abaixo:

$$\Psi = \sum_{i,j} \left(\sum_{k=1}^n \sum_{l=1}^n h_m(k, l) * y(i+k-1, j+l-1) - y_m(i, j) \right)^2 \quad \forall m$$

onde $*$ denota a operação de convolução. A função Ψ é minimizada com respeito aos valores do filtro: $h_m(k, l)$, a solução da equação (3.1) constitui o conjunto de FMQ 's $\{h_m(k, l)\}$ que é dado por:

$$\{h_m(k, l)\} = \begin{vmatrix} h_m(1, 1) & h_m(1, n) \\ h_m(n, 1) & h_m(n, n) \end{vmatrix} \quad (3.2)$$

Seja H_m o vetor filtro e y o vetor que representa a imagem original dentro da janela,

$$H_m = [h_m[1, 1], \dots, h_m[1, n], h_m[2, 1], \dots, h_m[n, n]]^T, \quad (3.3)$$

$$y = [y[i, j], \dots, y[i, j + n - 1], y[i + 1, j], \dots, y[i + n - 1, j + n - 1]]^T, \quad (3.4)$$

onde T representa a transposta. Então a equação (3.1) fica:

$$\Psi = \sum_{i,j} (H_m \cdot y^T - y_m(i, j)) \quad \forall m, \quad (3.5)$$

que pode ser simplificada para

$$\Psi = \sum_{i,j} \phi_m(i, j) \quad \forall m. \quad (3.6)$$

Minimizar a equação (3.1) com respeito ao filtro $h_m(k, l)$ significa resolver as equações:

$$\sum_{i,j} \frac{\partial \phi_m(i, j)}{\partial h_m(1, 1)} = 0 \quad (3.7)$$

$$\sum_{i,j} \frac{\partial \phi_m(i, j)}{\partial h_m(n, n)} = 0.$$

Então, a solução é $H_m = A^{-1} \cdot B_m^T$, onde A é a matriz de auto-correlação da imagem original $y(i, j)$ dada por:

$$A = \sum_{i,j} \begin{vmatrix} y^2(i, j) & y(i, j)y(i, j + 1) & y(i, j)y(i + n - 1, j + n - 1) \\ y(i, j)y(i, j + 1) & & \\ & & y^2(i + n - 1, j + n - 1) \end{vmatrix} \quad (3.8)$$

O vetor B corresponde ao vetor de correlação cruzada entre a imagem original $y(i, j)$, e a

textura conhecida $y_m(i, j)$:

$$B^T = \sum_{i,j} \begin{vmatrix} y(i, j)y_m(i, j) \\ y(i, j + 1)y_m(i, j) \\ \\ y(i, j + n - 1)y_m(i, j) \\ y(i + 1, j)y_m(i, j) \\ \\ y(i + n - 1, j + n - 1)y_m(i, j) \end{vmatrix}$$

H_m é o FMQ na forma vetorial $n^2 \times 1$, correspondente a $y_m(i, j)$, satisfazendo a equação (3.7).

Obviamente, o algoritmo simplesmente precisa das amostras $y_m(i, j)$, $i = 1, 2, \dots, r$, e da imagem original $y(i, j)$ ($1 \leq i \leq N - 1$, $1 \leq j \leq M - 1$). Calculando-se a correlação de $y(i, j)$ com cada $y_m(i, j)$, o FMQ é obtido [2, 8, 9].

No método dos FMQ's somente um filtro por textura é estimado para capturar as características únicas de cada textura, independente da área e da fronteira.

Deve-se salientar que o conhecimento a priori de $y_m(i, j)$ não envolve informação sobre as formas geométricas das texturas presentes em $y(i, j)$.

Os FMQ's podem ser visto como um conjunto de Kernels de convolução ortogonal que determina a presença de características texturais globais nas áreas texturizadas. De acordo com técnicas anteriores de classificação de texturas [8, 9, 23], as saídas desses filtros são usadas para classificar pixels ou posições como pertencentes a diferentes regiões texturizadas.

Para a imagem texturizada $y(i, j)$ de N por M pixels, e os filtros das equação (3.2) a

janela de resposta $R_m(i, j)$ é definida como:

$$R_m(i, j) = \sum_{k=1}^n \sum_{l=1}^n y(i+k, j+l) t_m(k, l) \quad \forall i, j \quad (3.10)$$

onde $i = 0, 2, \dots, N-1$; $j = 0, 2, \dots, M-1$.

A resposta retificada ou “atividade” é definida como:

$$A_i(i, j) = (R_m(i, j) - \langle R_m \rangle)^2 \quad \forall i, j \quad (3.11)$$

onde $\langle R_m \rangle$ corresponde a resposta média sobre toda a imagem.

Aqui os perfis da “atividade” são computados a partir da média sobre as regiões correspondendo ao tamanho dos elementos micro-padrão da textura.

A segmentação é conseguida, projetando-se os filtros em cada pixel (por correlação cruzada).

Veja que a técnica dos FMCQ's precisa de imagens testes, ou seja, suponha que um robô precise identificar e separar 3 tipos de texturas de tecido para o corte. Então, filma-se 3 pedaços dos tecidos e fornece-se ao programa as imagens destes. Logo ficará fácil identificar e separar os 3 padrões, usando a técnica abordada nesta seção, pois, o robô usará as imagens testes para identificar cada textura.

A desvantagem de tal método é que ele necessita das texturas testes, as quais devem ser idênticas àquelas presentes na imagem a segmentar.

3.2 Detecção de bordas por otimização vinculada.

Será visto agora um modelo proposto por Geman et.al. [15] para a segmentação de texturas contidas numa imagem. Ver-se-á aqui um modelo de representação de imagens denominado

de partição. Este técnica é um caso especial de um modelo mais geral de rótulos, que usa a forma de um “funcional de energia ” envolvendo duas componentes, uma que expressa as interações entre os rótulos, e outra que depende da interação entre os dados (pixels) da imagem e dos rótulos. A segmentação atribui a melhor configuração de rótulos para as texturas envolvidas. Esta configuração representa aquela de menor energia do sistema.

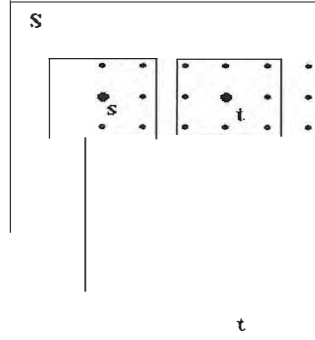
O termo de interação entre os rótulos incorpora um medida de disparidade entre certas características espaciais de pares de blocos de pixels, usando a medida não paramétrica de Kolmogorov-Smirnov. Grandes disparidades significam regiões de texturas diferentes. O número de parâmetros no modelo é reduzido pelas configurações de rótulos proibidas, tais como: pequenas regiões com altas variações de valores de pixels, bordas inacabadas na imagem, etc. Estes estados proibidos, são colocados como restrições no cálculo da configuração otimizada.

A segmentação consiste em atribuir a elementos de uma mesma textura, rótulos iguais tornando-se assim fácil a identificação das diferentes texturas na imagem. A segmentação pode ser complicada por uma ausência de informação sobre o número de texturas na imagem ou o tamanho das mesmas, assim como as formas geométricas presentes.

Aqui não existe modelo para texturas individuais e nenhuma capacidade para síntese das mesmas. As operações aqui são para pares de sítios vizinhos contidos em S (grade de pixels), e estes pares de sítios estão associados a pares de blocos de pixels, como na figura 3.2 abaixo.

A interação entre x (configuração de rótulos) e y (configuração de pixels) é definida em

Figura 3.1: Grade de pixels



termos de uma função energia

$$U = \sum_{\langle s,t \rangle} \psi_{s,t}(x) \Phi_{s,t}(y) \quad (3.12)$$

O somatório se estende a todos os vizinhos $\langle s, t \rangle$, $s, t \in S$; $\Phi_{s,t}(y)$ é a medida da disparidade entre os dois blocos de pixels, associados com os sítios de rótulos $s, t \in S$. $\psi_{s,t}(x)$ é uma função que depende somente dos rótulos x_s e x_t .

Toma-se simplesmente $\psi_{s,t}(x) = \delta_{\{x_s=x_t\}}$, em que δ é a função delta de Dirac. Desta maneira, nos estados de baixas energias, grandes disparidades ($\Phi > 0$) serão traduzidas como regiões diferentes ($x_s \neq x_t$), enquanto que em pequenas disparidades ($\Phi < 0$) ter-se-á regiões similares em texturas ($x_s = x_t$).

A interação entre rótulos e dados está baseada em várias medidas de disparidades em se comparando os blocos de dados da imagem. Estas medidas podem ser derivadas diretamente da imagem e de sua transformação(ões) da forma

$$t'_t = |t_t - \sum \alpha_j t_{tj}|, \quad (3.13)$$

onde $\sum \alpha = 1$ e $\{t_j\}$ são aqueles pixels vizinhos a t chamados de "direcionais residuais".

A disparidade é medida pela distância de Kolmogorov-Smirnov, uma ferramenta muito comum em estatística não paramétrica que possui certas propriedades de invariância [15].

Então a forma geral de Φ , assumindo inclusive as transformações na imagem, é a seguinte:

$$\Phi_{s,l}(y) = \phi\left(\max_{1 \leq i \leq m} \rho(y_s^{(i)}, y_l^{(i)})\right) \quad (3.14)$$

Onde ϕ é monotonicamente crescente, ρ denota a distância baseada na estatística de Kolmogorov-Smirnov, e $y_s^{(i)}, y_l^{(i)}$ são os dados associados aos blocos $\langle s, l \rangle$ para a i -ésima transformação [15]. a outra componente no modelo é a função penalidade $V(x)$, que contabiliza o número de "taboo patterns" (estados proibidos) em x , de tal forma que se $V(x) > 0$ tem-se uma proibição. As partições são penalizadas, quando existem muitas transições de pixels ou regiões pequenas demais. Então dada a imagem texturizada, o objetivo é estimar $x^* = x^*(l)$, com a solução vinculada a equação abaixo:

$$x^* = \arg \min_{x: V(x)=0} U(x).$$

razão para a otimização vinculada é que as expectativas sobre certos tipos de rótulos, são quase precisas e rígidas. No modelo aqui proposto, não existem parâmetros multiplicativos, tais como "suavizações" e/ou "pesos" como descritos nos trabalhos [9, 12, 14, 18, 16].

Os pontos de cortes para se fazer o "thresholding" de cada textura, devem ser selecionados das medidas de disparidade, isso pode ser feito "às cegas", mas a eficiência do algoritmo fica melhor se os dados forem bem direcionados após alguns testes na imagem.

A busca de x^* , a melhor rotulação, é baseada numa versão de relaxação estocástica, a qual é incorporada fortes vínculos e a teoria pode ser consultada no trabalho do Geman [13]. Neste trabalho são empregados dois algoritmos, os quais são aproximações precisas para se obter convergência.

É introduzido um parâmetro de controle t correspondendo a "temperatura" a ser baixada, e outro λ , que corresponde a um multiplicador de Lagrange para o vínculo $V = 0$ conforme

abaixo:

$$U_k = t_k^{-1}[U + \lambda_k V(k)] \quad (3.16)$$

Como y (imagem) é fixa, $t_k \searrow 0$, e $\lambda_k \nearrow \infty$. Gera-se então uma sequência de estados x_k^* , $k = 1, 2, \dots$, pelo algoritmo tipo Monte Carlo, da medida da energia de Gibbs U_k .

Se t_k é fixo, então sobre suaves crescimentos de λ_k , a sequência x_k converge para uma amostragem da distribuição de Gibbs, com energia U , mas restrita a $V = 0$, ou seja, vinculada a atribuir probabilidade zero as configurações proibidas, e a isso chama-se de “relaxação estocástica vinculada”. Se porém $t_k \searrow 0$, e $\lambda_k \nearrow \infty$, ambos a taxas suaves a sequência então converge para uma solução da equação (3.15), e este processo é denominado de “simulação de anelamento vinculado” (“constrained simulated annealing”).

Para se facilitar a colocação do modelo numa forma mais geral, seja $x = (x^L, x^P)$, onde x^L é o vetor de rótulos, e x^P é o vetor dos pixels.

A distribuição a priori Π é uma probabilidade para $x : 0 \leq \Pi(x) \leq 1$, $\sum_x \Pi(x) = 1$, onde \sum_x é o somatório sobre todas as configurações de x . Adotando a representação de Gibbs no caso sem vínculo ter-se-á que:

$$\begin{aligned} \Pi(x) &= \frac{1}{Z} \exp\{-U(x)\} \\ Z &= \sum_x \exp\{-U(x)\} \end{aligned} \quad (3.17)$$

onde função real U é a energia e determina Π .

$U(x^L, x^P)$ promove as atribuições de rótulos distintos entre regiões que demonstram diferentes padrões de textura. As interações entre rótulos é para inibir regiões de altas transições, regiões pequenas e outras “configurações inesperadas”. Nisso atribui-se altas energias ou probabilidade zero a essas tais “configurações inesperadas”.

3.2.1 A — Partição.

Agora seja a rede de pixels (imagem) $\{(i, j) : 1 \leq i, j \leq N\}$ denotada por S_I e seja S_L a rede de rótulos. Para cada experimento, uma resolução σ é escolhida, tal que determina uma sub-rede $S_L^{(\sigma)} \subset S_L$, um σ maior pode capturar mais detalhes da textura, mas pode perder detalhes da cena. Seja

$$S_L^{(\sigma)} = \{(i\sigma + 1, j\sigma + 1) : 0 \leq i, j \leq \frac{N}{\sigma} - 1\} \quad (3.18)$$

e os dados (imagem), consiste de níveis de cinza $y_s, s \in S_I$, tal que:

$$\Omega_I = \{\{y_s\} : s \in S_I, 0 \leq y_s \leq 255\} \quad (3.19)$$

O espaço de configuração para o particionamento de x , é determinado por σ , e por um número máximo de regiões permitidas P :

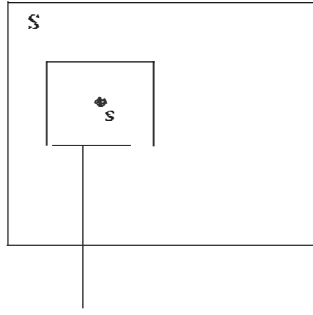
$$\Omega_I^{(\sigma, P)} = \{\{x_s\} : s \in S_L^{(\sigma)}, 0 \leq x_s \leq P - 1\} \quad (3.20)$$

Os rótulos são genéricos e x define uma partição identificando sítios com um dado rótulo $(0, 1, \dots, P - 1)$ como pertencendo a uma mesma região. Somente a sub-rede $S_L^{(\sigma)}$ é rotulada e um número máximo de rótulos é fixado a princípio, número este que deve refletir o número de regiões distintas.

A cada sítio de rótulo $s \in S_L$, está associado um bloco quadrado (figura 3.2.1) $B_s \in S_I$ de pixels centrados em s (lembre-se que S_L é uma cópia de S_I , e às vezes usa-se s ambigüamente para referenciar um sítio em S_L e em S_I).

x_s rotula os pixels em $B_s : \{\{y_r\} : r \in B_s\}$, como se verá o particionamento é baseado nas estatísticas espaciais destas subimagens (sobrepostas). O tamanho de B_s é importante. Com texturas, quanto maior B_s , mais informação do padrão pegar-se-á.

Figura 3.2: Bloco associado a s



3.2.2 B — Interação rótulo-dado.

Será estabelecido um sistema de vizinhança na rede de rótulos $S_L^{(\sigma)}$. Cada $s \in S_L^{(\sigma)}$ está associado a um conjunto de vizinhos $N_s \subset S_L^{(\sigma)}$. O sistema é simétrico tal que $s \in N_r \Leftrightarrow r \in N_s$. Seja $\langle s, l \rangle_\sigma$ um par de vizinhos ($s, l \in S_L^{(\sigma)}$ e $s \in N_l$). Será introduzido uma medida de disparidade $\Phi_{s,l} = \Phi_{s,l}(l)$ para cada par vizinho $\langle s, l \rangle_\sigma$. $\Phi_{s,l}$ mede a similaridade nos dois blocos associados aos pares $s, l \in S_L^{(\sigma)}$ e poderá ser $\Phi_{s,l} = -1$ (blocos iguais), ou $\Phi_{s,l} = +1$ (blocos diferentes) e a energia de interação fica:

$$U = \sum_{\langle s, l \rangle_\sigma} \delta_{\{r_s=r_l\}} \Phi_{s,l}(l) \quad (3.21)$$

onde $\delta_{\{x_s=x_l\}} = 1$ se $x_s = x_l$ e vice-versa.

A função disparidade não é exata, e durante a segmentação ocorrerá de texturas semelhantes serem classificadas como dissimilares ($\Phi = +1$) e vice-versa.

3.2.3 C — Medidas de disparidade.

Algumas vezes, só o histograma é suficiente para se fazer uma boa partição, como uma imagem SAR (synthetic aperture radar), de água e gelo. Sabendo que B_s , $s \in S_L^{(\sigma)}$, é um bloco quadrado de pixel (21×21 por exemplo) centrado em s . E seja $y(B_s) = \{y_r : r \in B_s\}$.

Dado dois vizinhos $s, l \in S_L^{(\sigma)}$, define-se $\Phi_{s,l}$ usando a estatística de Kolmogorov-Smirnov [15] e um limiar c .

$$\Phi_{s,l} = 2\delta_{\{d(y(B_s), y(B_l)) > c\}}(y) - 1 \quad (3.22)$$

$d(y(B_s), y(B_l))$ é a distância entre os blocos, $\Phi_{s,l}$ é $+1$ ou -1 , dependendo se a estatística de Kolmogorov-Smirnov [15] esteja acima ou abaixo do limiar.

Pode acontecer que texturas diferentes, tenham aproximadamente o mesmo histograma. Nestes casos, deve-se aplicar transformações na imagem, envolvendo várias estatísticas espaciais, como conseqüência duas texturas distintas vão diferir em pelo menos uma destas transformações. A equação (3.22) ficaria após incluir as transformações, da seguinte forma:

$$\max_{1 \leq i \leq m} [2\delta_{\{d(y^{(i)}(B_s), y^{(i)}(D_r)) > c_i\}}(y^{(i)}) - 1], \quad (3.23)$$

onde $y^{(i)}$ é i -ésima transformação da imagem, e $y^{(i)}(B_s) = \{y_s^{(i)}, s \in D_r\} r = s, l$. Os cortes ou limiares c_1, \dots, c_m são escolhidos de tal forma a limitar a taxa de “falsos alarmes”.

3.2.4 D — Penalidades.

Note que $V(x)$ conta o número total de penalidades associadas com $x \in \Omega_L^{(\sigma, P)}$. E que existem aqui dois tipos configurações proibidas que se usarão como penalidades: uma é a ocorrência de regiões muito pequenas e a outra são as regiões estreitas que se obtêm fixando σ e $s \in S_L^{(\sigma)}$. E definindo-se E_s como sendo um bloco de 5×5 (tamanho arbitrário) $\in S_L^{(\sigma)}$ e centrado em s . Então, diz-se que uma configuração é pequena em $s \in S_L^{(\sigma)}$ se menos de nove (número também arbitrário) rótulos em $\{x_t : t \in E_s\}$ concorda com x_s . O número total de

penalidades, para essas pequenas regiões é dado por:

$$\sum_{s \subset S_l^{(\sigma)}} \delta \left\{ \left(\sum_t \in E_s \delta_{\{x_t = x_s\}} \right) < 9 \right\} \quad (3.24)$$

Para as regiões estreitas, que geralmente têm um sítio de largura (vertical ou horizontal), faz-se o seguinte: seja τ_h uma translação horizontal de um sítio em $S_l^{(\sigma)}$ e do mesmo modo τ_v no caso vertical. As penalidades surgem quando $\{x_{s-\tau_h} \neq x_s, x_{s+\tau_h} \neq x_s\}$ ou $\{x_{s-\tau_v} \neq x_s, x_{s+\tau_v} \neq x_s\}$. O número de penalidades da “região magra” é:

$$\sum_{s \subset S_l^{(\sigma)}} \delta_{\{x_{s-\tau_h} \neq x_s, x_{s+\tau_h} \neq x_s\}} + \delta_{\{x_{s-\tau_v} \neq x_s, x_{s+\tau_v} \neq x_s\}}.$$

$V(x)$ será a soma da equação (3.24) com equação (3.25).

Pode-se então resumir todo o processo, da seguinte forma:

Tem-se:

1. imagem de nível de cinza $y = \{y_{i,j}\}$;
2. resolução $\sigma = 1, 2, \dots$, e um número máximo de rótulos P ;
3. medida de disparidade $\Phi_{s,t}(y)$ para cada par $\langle s, t \rangle_\sigma$ na sub-rede $S_l^{(\sigma)}$;
4. coleção de padrões de penalidades.

A partição $x^* = x^*(y)$ é então qualquer solução $x \in \Omega_l^{(\sigma, P)}$ da otimização vinculada a

$$x^*(y) = \arg \min_{x: V(x)=0} \sum_{\langle s, t \rangle} \delta_{\{x_s = x_t\}} \Phi_{s,t}(y), \quad (3.26)$$

onde $V(x)$ é o número de penalidades em x .

Capítulo 4

Algoritmo para segmentação de texturas modeladas por Campos Aleatórios de Markov

4.1 Introdução

Muitos problemas de análise de imagens envolve um processo de representação dos dados observados em termos de rótulos. A determinação destes rótulos é em alguns problemas o próprio objetivo da análise e em outras vezes serve como etapa intermediária para obter informações adicionais.

Neste trabalho lida-se com imagens formadas por regiões com texturas estocásticas. O objetivo é desenvolver um algoritmo que incorpore informação sobre textura para segmentar (particionar) a imagem em regiões que sejam homogêneas com relação às texturas que as compoem.

O algoritmo aqui proposto é uma adaptação do esquema de particionamento de imagens apresentado em [15] (Seção 3.2). No presente trabalho, implementa-se uma nova abordagem para medir as disparidades (ou diferenças de informação textural) entre as regiões. Este esquema segue uma estrutura geral para resolução de problemas de rotulação através da construção de uma função de energia que envolve duas componentes: uma sendo a interação entre os rótulos e as intensidades da imagem dada e a outra sendo composta por interações entre os rótulos em si e restrições que incorporem conhecimento a priori de como deve ser a configuração de rótulos. A configuração solução será aquela que minimizar tal energia. Como em geral não consegue-se abordar este problema de minimização com técnicas determinísticas, devido ao grande número de variáveis, adota-se uma abordagem de minimização estocástica mais especificamente o algoritmo de “annealing” simulado usando o amostrador de Gibbs. Para a amostragem pode-se ainda usar qualquer esquema do tipo Monte-Carlo.

O algoritmo não é para classificação e sim para particionar a imagem, mas pode ser adaptado para classificar texturas. Isto não é feito no presente trabalho e fica como sugestão para continuação da pesquisa.

A partição da imagem é genérica e rotula blocos de pixels com tamanho ou resolução variável, porém previamente fixada no início de cada particionamento. A resolução no presente trabalho depende das texturas envolvidas na imagem e do modelo de textura em termos de CAM usado. Se os blocos forem muito pequenos, o algoritmo para estimação dos parâmetros poderá apresentar sistemas de equações lineares com matriz singular.

Apresenta-se na próxima sub-seção um resumo do algoritmo de partição proposto em [15] (Seção 3.2) . As alterações aqui implementadas serão comentadas no ponto apropriado, ressaltando-se a diferença entre as duas abordagens.

4.2 Modelo de Partição

Adota-se a notação usada em [15] (Seção 3.2) para a formulação do problema e descrição do algoritmo de partição.

Seja a imagem y dada na grade de pixels

$$S_I = \{(i, j) : 0 \leq i, j \leq N - 1\}. \quad (4.1)$$

Seja a grade para os rótulos, com resolução σ , $S_L^{(\sigma)}$, dada por

$$S_L^{(\sigma)} = \{(i\sigma, j\sigma) : 0 \leq i, j \leq \frac{N}{\sigma} - 1\} \quad (4.2)$$

Note que quanto maior for o valor de σ maiores serão os blocos e menor será a resolução da partição. Observe que a grade de rótulos $S_L^{(\sigma)}$ é uma sub-grade da grade de imagens S_I .

A imagem observada y está no espaço de matrizes

$$\Omega_I = \{y = \{y_s\} : s \in S_I, 0 \leq y_s \leq 255\}. \quad (4.3)$$

o espaço de configurações para as partições x é determinado pela resolução σ e pelo número máximo de rótulos P :

$$\Omega_L^{(\sigma, P)} = \{\{x_s\} : s \in S_L^{(\sigma)}, 1 \leq x_s \leq P\}.$$

Somente a sub-grade $S_L^{(\sigma)}$ é rotulada e o número máximo de rótulos P é especificado inicialmente.

Cada sítio de rótulo $s \in S_L^{(\sigma)}$ está associado a um bloco de pixels da imagem, B_s , de tamanho $\sigma \times \sigma$ tendo s como seu sítio superior esquerdo.

A partição vista no trabalho [15] se baseia em estatísticas sobre transformações espaciais das sub-imagens. Controla-se uma função de energia U , que depende da configuração

de rótulos, e que deve ser minimizada pela configuração solução, conforme apresentado no Capítulo 3, Seção 3.2. A medida de disparidade ($\phi_{s,t}(y)$), ou distância estatística entre os blocos B_s e B_t da imagem y para efeito de decisão de associá-los a uma mesma textura, é tomada em como uma função da estatística de Kolmogorov-Smirnov (discutida por Geman [15]) medida sobre as transformações espaciais das sub-imagens.

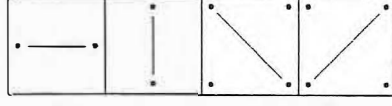
No presente trabalho, cada sub-imagem será representada por um vetor de parâmetros associado a um modelo fixo de CAM dado no Capítulo 2, equação (2.4), com os $U_{\lambda}^{(\alpha)}$'s dados pelas equações (2.5) a (2.10). Logo a informação da textura de cada bloco B_s estará contida no seu respectivo vetor de parâmetros $\vec{\theta}_s$. A abordagem por vetor de parâmetros de CAM leva em consideração a disposição espacial das intensidades que formam a textura, pois estes parâmetros medem a intensidade de interação entre pixels vizinhos segundo várias direções. A disparidade $\phi_{s,t}(y)$ no presente trabalho é função da distância Euclidiana $S(\vec{\theta}_s - \vec{\theta}_t)$, entre os vetores de parâmetros θ_s e θ_t .

4.2.1 Interação Imagem-Rótulo.

Estabelece-se um sistema de vizinhança sobre a grade de rótulos $S_L^{(\sigma)}$: cada $s \in S_L^{(\sigma)}$ tem associado um conjunto de vizinhos $N_s \subset S_L^{(\sigma)}$. O sistema é simétrico no sentido de que $s \in N_r \Leftrightarrow r \in N_s$. Denomina-se clique ao par de sítios s, r tal que $s \in N_r$. Há sistemas de vizinhanças que contêm cliques formados por mais de dois sítios, mas aqui usam-se apenas cliques formados por pares de sítios. Mais especificamente os cliques adotados são os mostrados na figura 4.2.1

Seja $\langle s, t \rangle_{\sigma}$ um clique. Introdúz-se a medida de disparidade $\phi_{s,t} = \phi_{s,t}(y)$ para cada par de cliques $\langle s, t \rangle_{\sigma}$. A função $\phi_{s,t}$ mede a similaridade entre as texturas nos blocos

Figura 4.1: Cliques usados



indexados por s e t respectivamente. A energia de interação é dada por:

$$U(x, y) = \sum_{\langle s, t \rangle \in \sigma} \delta_{\{x_s = x_t\}} \phi_{s, t}(y),$$

onde $\delta_{\{x_s = x_t\}} = 1$ se $x_s = x_t$ e 0 caso contrário. Nos baixos níveis de energia, rótulos similares ($x_s = x_t$) são associados a baixas medidas de disparidade e rótulos distintos ($x_s \neq x_t$) são associados a altas disparidades.

A probabilidade de uma configuração de rótulos x dada uma imagem y $\Pi(x|y)$ é dada por

$$\Pi(x|y) = \frac{1}{Z} \exp(-U(x, y)) \quad (4.6)$$

onde

$$Z = \sum_{x \in \Omega_{I_s, I_t}^{\{\sigma, \rho\}}} \exp(-U(x, y)). \quad (4.7)$$

No atual trabalho, a energia é tomada como

$$U(x, y) = \sum_{\langle s, t \rangle \in \sigma} (S(\vec{\theta}_s - \vec{\theta}_t) - \epsilon \delta_{\{x_s \neq x_t\}})^2, \quad (4.8)$$

$$S(\vec{\theta}) = \left[\sum_{\alpha=1}^P (\theta^{(\alpha)} p(\alpha))^2 \right]^{1/2}, \quad (4.9)$$

sendo $\vec{\theta}_s$ e $\vec{\theta}_t$ os vetores de parâmetros de textura estimados respectivamente dos blocos s e t e $p(\alpha) > 0$ $\alpha = 1, \dots, P$, os pesos de cada parâmetro no cálculo da distância S , que é uma norma em R^P . Observe que se $x_s = x_t$, a energia recebe a parcela $(S(\vec{\theta}_s - \vec{\theta}_t))^2$ que é

a distancia normalizada entre os parâmetros. No caso de $x_s \neq x_r$ a energia recebe o valor de $(\epsilon - S(\vec{\theta}_s - \vec{\theta}_r))^2$. Na implementação computacional do algoritmo os parâmetros para cada bloco são avaliados uma única vez, no início dos cálculos, e armazenados para uso nas iterações da relaxação estocástica. Isto implica que a imagem original é usada somente para gerar uma configuração fixa de parâmetros $\{\vec{\theta}_s : s \in S_L^{(\sigma)}\}$.

4.3 "Annealing" Simulado

É usado no algoritmo um esquema de "annealing" simulado [21]. Este processo é introduzido de tal forma que tende a favorecer a configuração de menor energia do sistema.

Para tal equilíbrio ser atingido é preciso que a cada número de iterações a temperatura seja baixada, satisfazendo uma regra prescrita de acordo com o trabalho de Geman e Geman [18] e que é dada por:

$$T(k) \geq \frac{c}{\log(1+k)} \quad (4.10)$$

onde k é a k -ésima iteração e c uma constante que independe de k .

Neste trabalho é usada uma temperatura inicial $T(k) = 100$ para se permitir um espalhamento dos rótulos. Depois disto propõe-se o abaixamento da temperatura prescrito pela equação (4.10). A medida que o tempo passa, faz-se a análise de cada pixel atualizado (em conjunto com a vizinhança) e se avalia a energia de cada configuração com as demais, fazendo-se uso do amostrador de Gibbs [18]. Faz-se uma decisão probabilística para decidir qual rótulo será escolhido.

O procedimento segue até que se atinja o número de iterações pré-definidas.

Observações:

1. O modelo para segmentar imagens baseado em texturas, apresentado nesta dissertação, pode ser ampliado para fazer a classificação automática das texturas presentes na imagem digital. Para isto, deve-se incluir um procedimento para agrupamento (“clustering”) dos blocos de pixels com tamanhos dados pela resolução σ , a partir da informação contida no vetor de parâmetros a eles associados. Uma vez feito o agrupamento, a classificação estaria completa assim como a segmentação.
2. Outro ponto a ser considerado com cautela é o tamanho dos blocos elementares de pixels sobre os quais são feitas as estimativas dos parâmetros. O algoritmo usado aqui para estimar os parâmetros é o Estimador Variacional proposto por Almeida [3]. Almeida [3] prova que este algoritmo é consistente e assintoticamente normal quando se faz a amostra (tamanho da janela) tender a infinito. No entanto, até o presente momento não foi feito nenhum estudo sobre “grandes desvios” dos parâmetros estimados e portanto não se conhece teoricamente o comportamento do algoritmo para amostras de tamanho finito. É fato que para imagens com poucos valores distintos de intensidades, é necessário que o tamanho da amostra seja grande, pois, caso contrário, o algoritmo de estimação variacional produz um sistema linear com matriz singular.

Na verdade, é facilmente provado que se uma matriz $A(n \times n)$ tem a forma

$$A = \sum_{k=1}^m \vec{X}^{(k)} \otimes \vec{X}^{(k)},$$

onde $\vec{X}^{(k)}, k = 1, \dots, m$ são vetores de R^n e \otimes representa o produto tensorial entre dois vetores, então A será não singular se e somente se $m \geq n$ e entre os vetores $\vec{X}^{(k)}, k = 1, \dots, m$ existirem n vetores linearmente independentes.

Capítulo 5

Experimentos

Neste capítulo são apresentados os resultados de alguns experimentos com imagens formadas por colagem de texturas. As imagens apresentadas são consideradas como matrizes contendo os níveis de cinza. As texturas são modeladas por CAM de acordo com o modelo proposto no Capítulo 3. O objetivo é observar o desempenho do algoritmo segmentador em imagens testes realísticas.

Como mostrado no Capítulo 4 considera-se que as imagens pertencem a um espaço de configurações Ω_I e as partições pertencem ao espaço de rótulos $\Omega_I^{(\sigma, P)}$. A probabilidade condicional de uma configuração de rótulos x dada uma imagem y , $\Pi(x|y)$, é dada por

$$\Pi(x|y) = \frac{1}{Z} \exp(-U(x, y)) \quad (5.1)$$

onde

$$Z = \sum_{x \in \Omega_I^{(\sigma, P)}} \exp(-U(x, y)). \quad (5.2)$$

O modelo aqui proposto usa a energia

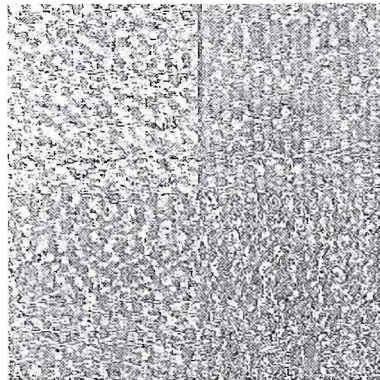
$$U(x, y) = \sum_{\langle s, t \rangle \in \sigma} (S(\vec{\theta}_s - \vec{\theta}_t) - \epsilon(1 - \delta_{\{x_s=x_t\}}))^2 \quad (5.3)$$

$$S(\vec{\theta}) = \left[\sum_{\alpha=1}^P (\theta^{(\alpha)} p(\alpha))^2 \right]^{1/2} \quad (5.4)$$

sendo $\vec{\theta}_s$ e $\vec{\theta}_l$ os vetores de parâmetros de textura estimados respectivamente dos blocos s e l . Nos experimentos que seguem são tomados os valores $T = 3.0$ para a temperatura final de equilíbrio e ϵ variável para cada experimento. Este coeficiente (ϵ) está relacionado com o valor limiar de $S(\vec{\theta}_s - \vec{\theta}_l)$ que separa pares de blocos com texturas similares dos pares com texturas diferentes. Quando $x_s = x_r$ a função energia é dada por $(S(\vec{\theta}_s - \vec{\theta}_l))^2$, que é a distância normalizada entre os parâmetros. No caso de $x_s \neq x_r$ a energia é dada por $(\epsilon - S(\vec{\theta}_s - \vec{\theta}_l))^2$.

A primeira imagem utilizada, figura 5.1, é composta de quatro texturas retiradas do livro de texturas de Brodatz [7], cada uma de tamanho de 128×128 pixels. Colam-se esses quatro blocos de 128×128 pixels formando uma imagem de 256×256 pixels.

Figura 5.1: Imagem 1



Antes de se processar a segmentação da imagem da figura 5.1, são definidos os parâmetros a serem inseridos no programa, oriundos do modelo apresentado no Capítulo 4. São eles:

1. a janela de resolução σ ;
2. a quantidade de texturas diferentes ou rótulos P ;

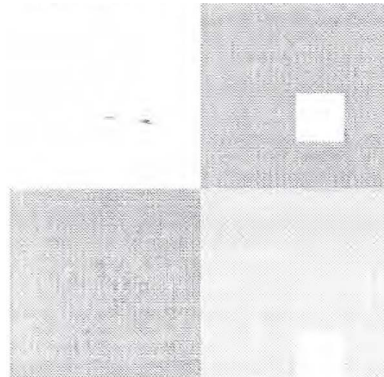
3. o número de iterações que o algoritmo deve perfazer antes de parar
4. a temperatura inicial c para o esquema do “annealing” simulado.
5. o limiar ϵ .

O resultado da segmentação da figura 5.1 é apresentado na figura 5.2, onde estão indicados os valores usados dos parâmetros.

Para verificar que dos blocos de 32×32 pixels podem-se obter parâmetros estimados, compatíveis com as texturas contidas nos blocos maiores, com 128×128 pixels, analisam-se os valores estimados com estas respectivas resoluções. Os resultados com $\sigma = 32$ estão nas tabelas 6.1 a 6.4, que referem-se aos quadrantes 1 a 4 da figura 5.1. Na tabela 6.5 estão os parâmetros obtidos com $\sigma = 128$ para os mesmos quadrantes.

O resultado da segmentação da figura 5.1 mostrado ns figura 5.2.

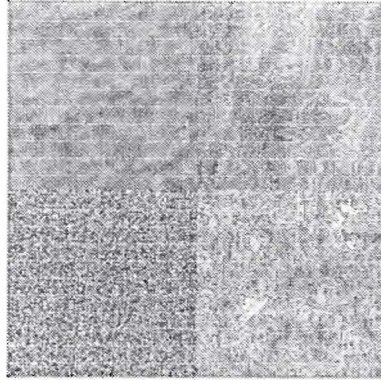
Figura 5.2: Segmentação da imagem ($\sigma = 32, P = 5, \epsilon = 12.44, T = 3.0$)



Outro conjunto de quatro texturas (192×192) obtidas no “site” <http://www.meat.com> da internet é mostrado na figura 5.3.

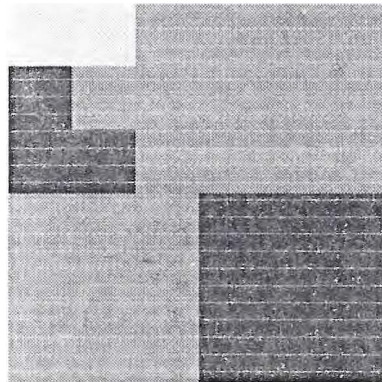
Para a segmentar imagem de entrada mostrada na figura 5.3 usam-se os parâmetros mostrados na figura 5.4.

Figura 5.3: Imagem 2



Também se estimam os parâmetros para cada textura (1-4) da figura 5.3 com janela $\sigma = 32$; os respectivos parâmetros estão nas tabelas 6.6 a 6.9. Na tabela 6.10 estão os parâmetros obtidos usando uma janela com resolução $\sigma = 128$. Como saída obtém-se a seguinte segmentação:

Figura 5.4: Segmentação da imagem 2 ($\sigma = 32$, $P = 5$, $\epsilon = 666.0$, $T = 3.0$)

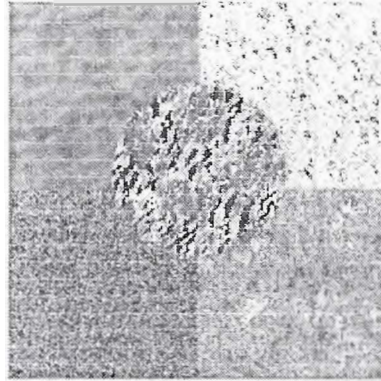


A terceira imagem usada está mostrada na figura 5.5. Esta tem também dimensões de 192×192 pixels e é composta de 5 texturas.

Obtém-se a segmentação dada pela figura 5.6 usando no algoritmo os parâmetros indicados na respectiva figura.

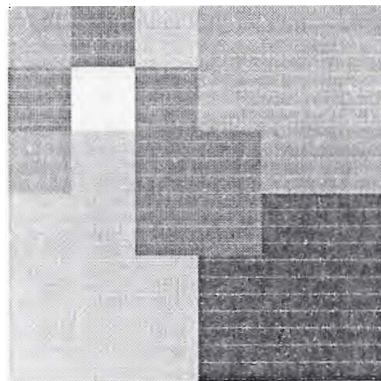
Nas tabelas 6.11 a 6.14 têm-se os parâmetros de cada textura (1-4) presente na figura 5.5 obtidos com resolução $\sigma = 32$. Para uma resolução $\sigma = 128$ têm-se os parâmetros mostrados

Figura 5.5: Imagem 3



na tabela 6.15. A segmentação da imagem 3 está na figura 5.6.

Figura 5.6: Segmentação da imagem 3 ($\sigma = 32$, $P = 5$, $\epsilon = 90.0$, $T = 3.0$)



Outra imagem usada é mostrada na figura 5.7.

Os parâmetros da imagem 4 são dados no rótulo da figura 5.8 que mostra a segmentação obtida.

Figura 5.7: Imagem 4

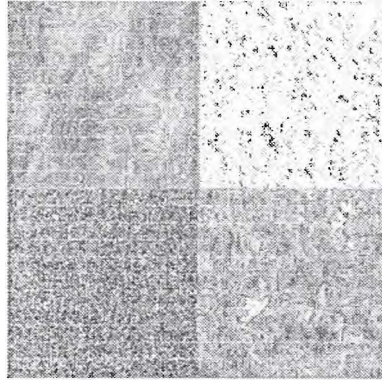
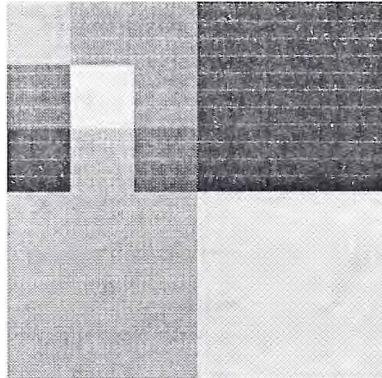


Figura 5.8: Segmentação da imagem 4 ($\sigma = 32$, $P = 5$, $\epsilon = 45.66$, $T = 3.0$)



Capítulo 6

Conclusões

Os resultados conseguidos no Capítulo 5 são promissores para a utilização de Campos Aleatórios de Markov em dois níveis na segmentação de imagens com texturas. Em todos os exemplos apresentados foi conseguido uma boa segmentação. No entanto mais testes devem ser feitos para ajuste fino dos parâmetros do algoritmo, e até mesmo usando funções de disparidade alternativas.

O ponto importante a ser salientado é que uma abordagem por modelo de Campos Aleatório de Markov para textura, mesmo que não totalmente adequado às texturas presentes na imagem analisada, pode fornecer uma medida (vetor de parâmetros do modelo) vantajosa para representar texturas, no contexto de um algoritmo de segmentação, substituindo um conjunto de estatísticas espaciais que contenham informações texturais como usado em [15]. Ademais o número de operações computacionais do modelo de CAM é bem menor que com as estatísticas, visto que o maior número delas está no cálculo dos parâmetros, o que é realizado uma única vez para cada bloco de resolução σ , no início do algoritmo, as iterações lidando somente com os valores inicialmente obtidos.

Tabela 6.1: Parâmetros da textura 1 da figura 5.1 com $\sigma = 32$

ι	J	λ	γ	A	h	β_1	β_2	β_3	β_4
0	0	7.617	0.076	25.113	0.936	-0.123	0.299	0.421	-0.182
1	0	7.197	0.214	23.802	1.361	-0.268	0.237	0.429	0.010
2	0	7.093	0.031	22.597	1.135	-0.423	0.454	0.029	-0.122
3	0	6.340	-0.008	19.893	1.036	-0.336	0.751	0.073	-0.159
0	1	7.621	0.273	25.864	1.735	-0.183	0.689	-0.115	0.268
1	1	7.527	0.049	24.547	1.245	-0.841	0.687	0.195	0.191
2		6.272	-0.095	19.710	1.070	0.203	0.332	0.180	-0.237
3		6.361	0.058	21.164	0.671	-0.283	0.778	0.378	0.129
0	2	6.314	-0.089	20.214	0.856	0.061	0.379	0.167	-0.010
1	2	5.852	0.031	19.286	0.691	-0.361	0.717	0.416	0.229
2	2	5.378	0.030	16.360	0.973	-0.070	0.670	-0.009	-0.047
3	2	5.462	-0.070	16.433	0.743	0.032	0.152	0.228	-0.178
0	3	6.831	0.099	23.786	0.645	0.945	1.042	0.023	-0.370
	3	6.332	-0.051	20.656	0.932	-0.356	0.786	0.588	-0.059
2	3	6.144	0.025	19.641	0.884	-0.068	0.308	0.196	-0.093
3	3	6.228	-0.120	20.381	0.448	-0.471	1.002	0.342	0.082

Tabela 6.2: Parâmetros da textura 2 da figura 5.1 com $\sigma = 32$

j	λ	γ	A	h	β_1	β_2	β_3	β_4	
4	0	2.866	-0.224	6.954	0.055	-0.025	-0.206	0.105	0.547
5	0	3.150	-0.379	7.495	-0.139	-0.656	0.227	0.255	0.568
6	0	2.560	-0.266	6.130	0.048	-0.086	0.033	0.518	0.465
7	0	2.968	-0.155	6.485	-0.026	-0.428	0.052	0.182	0.288
4	1	3.092	-0.422	6.106	0.116	-0.548	0.218	0.046	0.076
5	1	3.372	-0.253	8.232	0.145	-0.336	0.368	0.116	0.106
6	1	3.224	-0.049	7.653	0.201	-0.481	0.308	0.095	0.159
7	1	3.098	-0.291	6.759	-0.053	-0.373	0.149	0.053	0.141
4	2	3.587	-0.204	9.313	0.215	-0.376	0.060	0.159	0.313
5	2	4.628	-0.198	12.380	0.155	-0.523	-0.018	0.118	-0.014
6	2	2.331	-0.330	2.754	-0.302	-1.141	0.391	0.291	0.131
7	2	2.419	-0.329	4.100	-0.176	-0.112	0.231	-0.064	-0.108
4	3	3.835	-0.166	9.856	0.202	-0.386	0.416	-0.218	0.097
5	3	3.989	-0.183	10.861	0.211	-0.520	0.247	0.354	0.144
6	3	3.733	-0.138	9.531	0.243	-0.759	0.558	0.211	0.130
7	3	3.632	-0.309	8.584	0.040	-1.140	0.522	0.184	0.466

Tabela 6.3: Parâmetros da textura 3 da figura 5.1 com $\sigma = 32$

ι	J	λ	γ	A	h	β_1	β_2	β_3	β_4
0	4	3.251	-0.199	8.684	0.278	-0.088	0.712	0.272	-0.089
	4	3.648	-0.361	9.878	0.265	0.013	0.380	0.371	-0.170
2	4	3.236	-0.334	7.993	0.124	0.015	0.692	0.125	-0.189
3	4	3.643	-0.387	9.182	0.300	-0.092	0.373	0.145	-0.164
0	5	3.236	-0.039	9.452	0.421	0.246	0.436	0.625	0.069
	5	3.420	-0.229	9.204	0.241	0.455	0.364	0.329	-0.294
2	5	2.989	-0.247	7.594	0.243	0.158	0.457	0.435	-0.208
3	5	2.791	-0.285	6.810	0.157	0.382	0.671	0.016	-0.216
0	6	4.821	-0.087	13.908	0.473	-0.127	0.964	-0.094	-0.366
1	6	4.075	-0.031	11.625	0.497	-0.444	0.803	0.367	-0.062
2	6	4.093	-0.301	9.135	0.079	-0.467	0.642	0.036	-0.611
3	6	4.346	-0.149	11.413	0.395	-0.291	0.276	0.221	-0.312
0	7	3.544	-0.234	8.461	0.418	-0.189	0.552	0.267	-0.483
	7	4.139	-0.146	10.448	0.508	-0.550	0.947	0.156	-0.317
2	7	3.128	-0.326	6.452	0.196	-0.454	1.171	-0.088	-0.284
3	7	3.809	-0.296	9.414	0.267	-0.365	0.528	0.071	-0.109

Tabela 6.4: Parâmetros da textura 4 da figura 5.1 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
4	4	5.395	-0.026	16.638	0.893	-0.048	-0.220	0.120	0.235
5	4	5.165	-0.160	14.084	0.150	0.186	-0.723	0.154	-0.212
6	4	6.375	-0.012	20.056	0.631	0.104	-0.380	0.091	0.055
7	4	5.045	-0.042	15.377	0.436	0.063	-0.359	0.448	0.101
4	5	6.241	-0.014	19.819	0.685	0.152	-0.385	0.235	0.134
5	5	5.494	-0.100	16.761	0.385	0.335	-0.599	0.027	0.245
6	5	5.417	-0.119	15.112	0.561	-0.133	-0.741	0.286	-0.012
7	5	5.344	-0.124	15.133	0.362	-0.185	-0.410	0.167	-0.048
4	6	5.440	-0.195	15.440	0.127	-0.237	-0.364	0.264	-0.219
5	6	5.463	-0.201	16.046	0.147	0.071	-0.493	0.211	0.012
6	6	4.889	-0.230	13.201	0.024	-0.184	-0.295	-0.019	-0.143
7	6	5.548	-0.152	16.480	0.106	-0.044	-0.479	0.162	0.154
4	7	6.376	-0.164	19.346	-0.139	-0.370	-0.359	0.147	0.118
5	7	6.537	-0.286	18.474	-0.300	-0.284	-0.567	-0.239	-0.076
6	7	3.452	-0.323	7.283	-0.246	-0.349	-0.296	-0.162	-0.078
7	7	5.801	-0.079	15.123	0.358	-0.294	-0.181	-0.547	-0.403

Tabela 6.5: Parâmetros das 4 texturas da figura 5.1 com $\sigma = 128$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
0	0	6.365	0.016	20.551	0.932	-0.161	0.567	0.207	-0.051
	0	3.173	-0.247	7.291	0.037	-0.499	0.207	0.122	0.200
0		3.550	-0.226	9.079	0.299	-0.158	0.659	0.176	-0.258
		5.436	-0.124	15.734	0.314	-0.110	-0.398	0.085	-0.018

Tabela 6.6: Parâmetros da textura 1 da figura 5.3 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
0	0	328.454	664.815	-940.843	-136.713	144.499	150.080	-50.691	-35.906
1	0	252.409	536.498	-778.275	-113.297	-9.797	160.792	44.126	30.905
2	0	-25.134	-23.951	30.714	8.916	11.381	176.670	23.877	33.112
0	1	125.072	224.669	-242.307	-20.146	219.411	46.489	-6.518	25.231
1	1	20.497	45.755	-9.133	16.687	94.394	97.965	53.662	74.662
2	1	46.869	89.282	-47.635	15.859	106.196	21.648	80.222	134.448
0	2	128.329	290.452	-422.662	-56.443	177.791	-29.212	33.900	61.108
1	2	123.120	241.294	-274.298	-17.812	130.374	24.115	48.760	141.608
2	2	-26.385	-63.170	159.251	43.137	72.121	63.721	51.010	93.383

Tabela 6.7: Parâmetros da textura 2 da figura 5.3 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
3	0	12.289	19.266	-15.818	0.001	21.757	28.929	-7.438	-7.247
4	0	18.470	-4.340	14.200	-0.159	8.447	15.393	-1.199	6.254
5	0	12.455	9.917	1.297	1.096	22.299	22.531	-5.320	-4.747
3	1	33.913	55.468	-57.952	-5.004	19.733	30.114	-11.478	-6.117
4	1	27.543	31.315	-7.816	2.718	10.486	24.331	-2.643	2.182
5	1	27.495	6.108	10.245	0.852	17.303	9.764	-1.363	9.945
3	2	37.182	49.654	-32.643	0.344	20.957	25.292	-6.623	-0.125
4	2	7.622	11.810	-10.934	-0.678	20.540	22.497	-6.695	-3.432
5	2	60.720	19.955	18.113	1.753	13.832	11.026	-0.744	1.005

Tabela 6.8: Parâmetros da textura 3 da figura 5.3 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
0	3	5.822	3.398	1.949	-1.692	0.465	0.869	-0.464	-0.297
1	3	7.909	6.972	0.401	-0.967	0.797	0.020	-0.521	0.428
2	3	5.598	3.239	1.534	-1.839	0.303	0.100	0.051	-0.225
0	4	5.079	1.879	4.426	-1.280	0.096	0.510	-0.156	-0.208
1	4	7.351	5.020	0.311	-2.684	-0.179	-0.082	-0.838	-0.312
2	4	5.029	2.098	2.788	-1.601	-0.085	0.413	-0.122	-0.050
0	5	6.450	4.541	0.482	-1.902	-0.126	0.098	-0.026	-0.327
1	5	5.638	2.628	5.298	-0.718	0.475	0.219	-0.521	0.552
2	5	7.139	5.521	1.675	-0.954	0.604	0.598	-0.191	0.069

Tabela 6.9: Parâmetros da textura 4 da figura 5.3 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
3	3	7.076	6.679	-3.483	0.251	7.301	4.584	-0.026	-1.467
4	3	7.242	5.694	-2.499	0.071	7.378	7.549	-1.603	-2.064
5	3	5.146	-0.625	4.641	-0.026	5.136	3.378	0.794	-0.794
3	4	2.521	1.026	0.684	0.195	8.069	5.792	-1.840	-0.370
4	4	2.002	0.995	-0.964	-0.105	5.712	6.084	-1.622	1.269
5	4	8.099	13.168	-12.334	-0.322	12.628	11.952	-5.034	-1.944
3	5	1.468	0.188	-1.721	-0.357	6.516	3.562	-0.242	0.198
4	5	1.986	0.565	1.136	0.057	8.623	2.331	2.841	0.968
5	5	10.560	9.570	0.036	1.062	12.708	6.759	-0.992	-2.394

Tabela 6.10: Parâmetros das 4 texturas da figura 5.3 com $\sigma = 96$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
0	0	53.127	108.606	-129.049	-11.264	101.110	97.758	-0.779	23.604
1	0	6.359	4.075	2.783	0.660	15.766	19.198	-4.007	0.596
0	1	5.797	3.260	2.856	-1.377	0.294	0.374	-0.316	0.004
1	1	1.863	0.468	-0.436	-0.138	7.471	5.268	-0.612	-0.579

Tabela 6.11: Parâmetros da textura 1 da figura 5.5 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
0	0	328.454	664.815	-940.843	-136.713	144.499	150.080	-50.691	-35.906
1	0	252.409	536.498	-778.275	-113.297	-9.797	160.792	44.126	30.905
2	0	-25.134	-23.951	30.714	8.916	11.381	176.670	23.877	33.112
0	1	125.072	224.669	-242.307	-20.146	219.411	46.489	-6.518	25.231
1	1	-122.583	-241.545	363.433	59.680	32.430	28.805	63.462	68.629
2	1	2.142	-1.331	4.020	-1.036	2.587	0.190	0.063	0.370
0	2	128.329	290.452	-422.662	-56.443	177.791	-29.212	33.900	61.108
1	2	-1.195	-6.135	-0.603	-5.067	5.926	0.593	-2.646	3.005
2	2	6.619	0.750	13.638	-0.677	0.206	-0.454	-0.149	0.764

Tabela 6.12: Parâmetros da textura 2 da figura 5.5 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
3	0	10.476	2.491	22.748	2.982	0.248	-0.202	0.910	-0.609
4	0	9.949	2.999	20.665	3.978	-0.783	-0.154	0.900	0.347
5	0	10.146	3.280	23.023	3.101	-0.448	0.110	1.524	0.004
3	1	7.678	1.850	17.076	0.879	0.177	0.390	0.207	0.699
4	1	9.411	3.527	21.552	3.071	-0.424	0.167	0.756	0.353
5	1	10.018	3.853	22.123	4.336	-0.667	-0.191	0.542	0.415
3	2	6.397	1.357	11.936	-0.040	0.428	-0.126	-0.006	0.733
4	2	7.910	2.162	17.289	1.655	0.594	0.302	1.177	0.477
5	2	9.364	3.523	20.893	3.893	-0.521	-0.251	1.085	0.145

Tabela 6.13: Parâmetros da textura 3 da figura 5.5 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
0	3	5.822	3.398	1.949	-1.692	0.465	0.869	-0.464	-0.297
1	3	4.360	0.552	6.242	-1.018	0.915	-0.132	-0.587	0.778
2	3	6.965	0.744	13.206	-0.592	-0.091	0.041	-0.513	0.798
0	4	5.079	1.879	4.426	-1.280	0.096	0.510	-0.156	-0.208
1	4	7.351	5.020	0.311	-2.684	-0.179	-0.082	-0.838	-0.312
2	4	4.302	1.122	3.945	-1.269	0.144	0.489	-0.073	0.135
0	5	6.450	4.541	0.482	-1.902	-0.126	0.098	-0.026	-0.327
1	5	5.638	2.628	5.298	-0.718	0.475	0.219	-0.521	0.552
2	5	7.139	5.521	1.675	-0.954	0.604	0.598	-0.191	0.069

Tabela 6.14: Parâmetros da textura 4 da figura 5.5 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
3	3	6.025	0.669	13.428	-0.540	0.673	0.069	-0.003	0.620
4	3	-1.463	-1.002	-3.468	-0.654	2.919	1.261	-0.960	1.377
5	3	5.124	-0.624	4.571	-0.026	5.082	3.377	0.809	-0.801
3	4	0.256	-0.324	-2.042	-0.285	6.784	4.786	-2.173	0.328
4	4	2.002	0.995	-0.964	-0.105	5.712	6.084	-1.622	1.269
5	4	8.099	13.168	-12.334	-0.322	12.628	11.952	-5.034	-1.944
3	5	1.468	0.188	-1.721	-0.357	6.516	3.562	-0.242	0.198
4	5	1.986	0.565	1.136	0.057	8.623	2.331	2.841	0.968
5	5	10.560	9.570	0.036	1.062	12.708	6.759	-0.992	-2.394

Tabela 6.15: Parâmetros das 4 texturas da figura 5.5 com $\sigma = 96$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
0	0	-3.504	-8.239	0.575	-4.598	6.871	1.776	-0.419	3.417
1	0	9.097	2.668	20.825	2.185	0.022	0.199	0.940	0.483
0	1	4.748	0.943	7.566	-0.347	0.487	0.438	-0.255	0.528
1	1	-2.122	-1.642	-3.187	-0.724	4.077	1.933	-0.260	1.196

Tabela 6.16: Parâmetros da textura 1 da figura 5.7 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
0	0	9.671	16.280	-14.361	-0.067	13.049	20.142	-2.254	-2.308
1	0	19.470	-3.029	13.464	-0.116	7.766	14.635	-0.943	5.542
2	0	8.450	7.368	-1.154	0.189	16.625	15.367	-1.415	-1.067
0	1	34.269	56.661	-59.274	-4.995	13.735	21.759	-6.915	-1.737
1	1	26.908	30.706	-7.661	2.699	7.407	19.681	-0.897	4.940
2	1	24.278	5.099	9.502	0.760	14.234	8.445	0.266	9.760
0	2	36.157	47.397	-29.360	0.746	14.910	16.855	-2.327	4.081
1	2	4.840	7.497	-7.293	-0.551	14.139	16.590	-2.210	-0.367
2	2	57.021	18.701	16.782	1.586	12.043	10.114	-0.367	0.822

Tabela 6.17: Parâmetros da textura 2 da figura 5.7 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
3	0	10.476	2.491	22.748	2.982	0.248	-0.202	0.910	-0.609
4	0	9.949	2.999	20.665	3.978	-0.783	-0.154	0.900	0.347
5	0	10.146	3.280	23.023	3.101	-0.448	0.110	1.524	0.004
3	1	11.164	2.271	20.630	3.827	-0.508	0.106	0.094	0.236
4	1	9.898	3.639	21.237	3.966	-0.663	0.002	0.352	0.239
5	1	10.018	3.853	22.123	4.336	-0.667	-0.191	0.542	0.415
3	2	10.294	3.762	21.723	4.727	0.072	-0.521	0.077	-0.131
4	2	9.523	3.522	21.768	3.271	-0.151	-0.598	0.875	0.934
5	2	9.364	3.523	20.893	3.893	-0.521	-0.251	1.085	0.145

Tabela 6.18: Parâmetros da textura 3 da figura 5.7 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
0	3	5.822	3.398	1.949	-1.692	0.465	0.869	-0.464	-0.297
1	3	7.909	6.972	0.401	-0.967	0.797	0.020	-0.521	0.428
2	3	5.598	3.239	1.534	-1.839	0.303	0.100	0.051	-0.225
0	4	5.079	1.879	4.426	-1.280	0.096	0.510	-0.156	-0.208
1	4	7.351	5.020	0.311	-2.684	-0.179	-0.082	-0.838	-0.312
2	4	5.029	2.098	2.788	-1.601	-0.085	0.413	-0.122	-0.050
0	5	6.450	4.541	0.482	-1.902	-0.126	0.098	-0.026	-0.327
1	5	5.638	2.628	5.298	-0.718	0.475	0.219	-0.521	0.552
2	5	7.139	5.521	1.675	-0.954	0.604	0.598	-0.191	0.069

Tabela 6.19: Parâmetros da textura 4 da figura 5.7 com $\sigma = 32$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
3	3	7.077	6.649	-3.455	0.239	7.289	4.598	-0.019	-1.476
4	3	7.168	5.645	-2.540	0.057	7.496	7.611	-1.708	-2.104
5	3	5.124	-0.624	4.571	-0.026	5.082	3.377	0.809	-0.801
3	4	2.521	1.026	0.684	0.195	8.069	5.792	-1.840	-0.370
4	4	2.002	0.995	-0.964	-0.105	5.712	6.084	-1.622	1.269
5	4	8.099	13.168	-12.334	-0.322	12.628	11.952	-5.034	-1.944
3	5	1.468	0.188	-1.721	-0.357	6.516	3.562	-0.242	0.198
4	5	1.986	0.565	1.136	0.057	8.623	2.331	2.841	0.968
5	5	10.560	9.570	0.036	1.062	12.708	6.759	-0.992	-2.394

Tabela 6.20: Parâmetros das 4 texturas da figura 5.7 com $\sigma = 96$

i	j	λ	γ	A	h	β_1	β_2	β_3	β_4
0	0	5.635	4.005	1.731	0.523	12.158	15.202	-1.823	2.378
1	0	10.041	3.407	21.945	3.798	-0.317	-0.213	0.721	0.246
0	1	5.797	3.260	2.856	-1.377	0.294	0.374	-0.316	0.004
1	1	1.863	0.466	-0.438	1.119	7.470	5.273	-0.614	-0.585

Bibliografia

- [1] Ackley, D. H., G. E. Hinton & T.J. Sejnowski, "A Learning Algorithm for Boltzman Machines," *Cognitive Sciences*, Vol. 9, pp. 147-169, 1985.
- [2] Ade, F., "Characterization of textures by "eigenfilter"," *Signal Process.*, Vol 5, pp. 451-457, 1983.
- [3] Almeida, M.P., "em Statistical Inference for MRF with Unbounded Continuous Spins and Applications to Texture Representation", Ph.D. Thesis, Division of Applied Mathematics, Brown University, 1989.
- [4] Almeida, M.P. & B. Gidas, "A Variational Method for Estimating The Parameters of MRF from Complete or Incomplete Data, *Annals of Applied Probability*, 1983.
- [5] Besag, J., "On the statistical analysis of dirty pictures, (with discussion)", *J. Roy. Statist. Soc., series B*, Vol 48, pp. 259-302, 1986.
- [6] Blake, A., "The least principle disturbance and weak constraints," *Pattern Recog. Lett.*, Vol. 1, pp. 393-399, 1983.
- [7] Brodatz, P., "Textures," Dover Publ., 1966.

- [8] Caelli, T. M., "Three processing characteristics of virtual texture segmentation," *Spatial Vision*, Vol. 1, No. 1, 1985, pp 19-30.
- [9] Caelli, T. M., "An adaptive computation model for texture segmentation," *IEEE Trans. Systems, Man Cybernet*, 1989.
- [10] Cheng, H., & Y. Dai, "Texture segmentation by least-square filter," *Signal Proc.* Vol 19, pp. 199-204, 1990.
- [11] Dobrushin, R. L., "The Description of a Random Field by Means of Conditional Probabilities and Conditions of its Regularity," *Theory of Prob. and its Applications*, Vol. XIII, No 2, pp. 197-224, 1968.
- [12] Geman, D., "A stochastic model for boundary detection," *Image Vision Comput.*, May
- [13] Geman, D. & S. Geman, "Relaxation and annealing with constraints," Division Appl. Math., Brown Univ., Complex Systems Tech Rep. 35, 1987.
- [14] Geman, D., S. Geman, C. Graffigne, "Locating texture and object boundaries," In *Pattern Recognition Theory and Application*, P.A. Devijver and J. Kittler, Eds. Heidelberg: Springer-Verlag, 1987.
- [15] Geman, D., S. Geman, C. Graffigne & P. Dong, "Boundary Detection by Constrained Optimization," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 12 pp. 192-202, 1991.
- [16] Geman, S. & D. F. McClure, "Statistical methods for tomographic image reconstruction," In *Bull. ISI (Proc. 46th Session Int. Statistical Institute)*, vol. 52, 1987.

- [17] Geman, S. & D.E. McClure, "Bayesian Image Analysis: An Application to Single Photon Emission Tomography," In *Proc. Amer. Statist. Assoc., Statistical Computing Section*. Amer. Statist. Assoc., 1985.
- [18] Geman, S. & D. Geman, "Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 6, pp. 721-741, 1984.
- [19] Georgii, H., "Gibbs Measures and Phase Transitions," Walter de Gruyter, 1988.
- [20] Gonzalez, C. R. & R. E. Woods, "Digital Image Processing," Addison-Wesley Publ. Com., Inc., 1992
- [21] Kirkpatrick, S., C.D. Gelatt Jr. & M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, 220 (4598), pp. 671-680, 1983.
- [22] Lanford III, O. E., & D. Ruelle, "Observables at Infinity and States with Short Range Correlations in Statistical Mechanics," *Commun. Math. Phys.*, Vol 13, pp. 194-215, 1969.
- [23] Laws, K. I., "Rapid texture identification: image processing for missile guidance," *SPIE*, Vol. 238, 1980, pp 376-380.
- [24] Lee, K. F., "Large-Vocabulary Speaker-Independent Continuous Speech Recognition," the SPHINX System, Ph.D. Thesis, Computer Science Dept., Carnegie-Melon University, 1988.

- [25] Ruelle, D., "Thermodynamic Formalism," Addison-Wesley Publishing Company, Reading, Mass., 1978.
- [26] Torreão, J. R. A., "Métodos Estocásticos em Computação Visual," VIII Escola de Computação, Gramado, RS, 1992.
- [27] Van Gool, L., P. Dewaele & A. Oosterlinck, "Survey Texture Analysis Anno 1983", *Computer Vision, Graphics and Image Processing*, Vol. 29, 336-357, 1985.