

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**  
**FACULDADE DE ENGENHARIA ELÉTRICA**  
**PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**ALGORITMOS GENÉTICOS DA MÍNIMA ENTROPIA**  
**E DOS MODELOS DE TEXTURA MODIFICADOS**  
**APLICADOS À SÍNTESE DE TEXTURA**

**ROBSON VIEITAS RAMOS**

**JULHO**

**2005**

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA ELÉTRICA  
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALGORITMOS GENÉTICOS DA MÍNIMA ENTROPIA  
E DOS MODELOS DE TEXTURA MODIFICADOS  
APLICADOS À SÍNTESE DE TEXTURA

Dissertação apresentada por Robson Vieitas Ramos  
à Universidade Federal de Uberlândia para obtenção do  
título de Mestre em Engenharia Elétrica aprovada em  
22/07/2005 pela Banca Examinadora:

Professora Edna Lúcia Flôres, Dra (orientadora)  
Professora Kátia Lopes Silva, Dra (UNIMINAS)  
Professor Gilberto Arantes Carrijo, Ph.D. (UFU)  
Professor Keiji Yamanaka, Ph.D. (UFU)

ALGORITMOS GENÉTICOS DA MÍNIMA ENTROPIA  
E DOS MODELOS DE TEXTURA MODIFICADOS  
APLICADOS À SÍNTESE DE TEXTURA

ROBSON VIEITAS RAMOS

Dissertação apresentada por Robson Vieitas Ramos à Universidade Federal de Uberlândia  
como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

---

Professora Edna Lúcia Flôres, Dra. Eng  
Orientadora

---

Professor Darizon Alves Andrade, Ph.D.  
Coordenador do Curso de Pós Graduação

**À Deus**

**Aos meus pais Vinícius e Janete.**

**Aos meus irmãos Alexandre e Marcus.**

**À minha amada Carla**

## **Agradecimentos**

À professora e orientadora Edna Lúcia Flôres pela orientação e incentivo.

Ao amigo Giuliano Ferreira Dela Coleta pela sua contribuição na fase final dessa dissertação.

À querida Carla Elena Dias Martins por toda ajuda, paciência, dedicação, incentivo, carinho e amor que foram fundamentais para a realização deste trabalho.

Ramos, Robson V., Algoritmos Genéticos da Mínima Entropia e dos Modelos de Textura Modificados Aplicados À Síntese de Textura, Uberlândia, FEELT – UFU, 2005, 128 p.

## RESUMO

Atualmente a síntese de textura é um dos grandes desafios. Os métodos mais recentes restringem os tipos de imagens de texturas a serem modeladas devido à dificuldade de obter um método genérico de síntese para uma grande variedade desses tipos de imagens. A partir de um método eficiente de síntese de texturas, usando apenas uma amostra da textura original e realizando a síntese à medida que seja preciso utilizá-la, não é necessário armazenar grandes imagens, pois elas ocupam muito espaço. Este trabalho propôs e implementou modificações e otimizações no método de síntese de texturas utilizando algoritmos genéticos da mínima entropia (AGME) e modelos de textura (AGMT) desenvolvidos por Ashlock e Davidson. O principal objetivo desta dissertação foi obter um método de síntese de texturas que a partir de uma amostra de textura sintetiza uma outra amostra bastante similar à amostra fornecida. Os algoritmos implementados neste trabalho foram os algoritmos genéticos da mínima entropia modificado (AGMEM) e dos modelos de textura modificado (AGMTM). O AGMEM conseguiu diminuir em mais de 70% o número de gerações necessárias para encontrar um  $IRRW\ SDWWHUQ$  adequado reduzindo o tempo total de processamento de 0,9% a 6,73% em relação aos algoritmos AGME e AGMT. As imagens sintetizadas pelos modelos estocásticos obtidos pelo AGMTM apresentaram, para a maioria das amostras utilizadas neste trabalho, um erro menor do que o erro das imagens sintetizadas pelos modelos do AGMT em relação aos modelos das texturas das amostras originais.

3 D0DYUDV FKDYH WH[ WXUD VtQWHVH GH WH[ WXUD DOJ RULWP R J HQpWLF R P RGH0R GH O DUNRY SDFLDP HQWH

RUGHQDGRV QmR SDUDP pWULFRV IRRW SDWWHUQ

Ramos, Robson V., Genetic Algorithms of the Minimum Entropy and Texture Models Modified applied at the Texture Synthesis, Uberlândia, FEELT – UFU, 2005, 128 p.

### ABSTRACT

Nowadays, texture synthesis is one of the greatest challenges. The most recent methods restrict the types of texture images to be modeled due to the difficulty to get a generic method of synthesis for a great variety of these types of images. From an efficient method of textures synthesis, using only one sample of the original texture and performing the synthesis when needed, there is no need to store big images, which would take much space. This work has proposed and implemented modifications and improvements on the method of synthesis of textures using genetic algorithms of minimum entropy (AGME) and models of texture (AGMT) developed by Ashlock and Davidson. The main objective of this work was to obtain a method of textures synthesis that could, from one sample of texture, synthesize another, similar enough to the supplied sample. The algorithms implemented in this work have been the genetic algorithms of minimum modified entropy (AGMEM) and models of modified texture (AGMTM). The AGMEM decreased in more than 70% the number of needed generations to find one adequate foot pattern, reducing the total processing time from 0,9% to 6.73%, compared to the AGME and AGMT algorithms. The images synthesized by the stochastic models obtained by the AGMTM have presented, for most samples used in this work, a smaller error than the error of the images synthesized by the models of the AGMT, compared to the models of original textures samples.

. H\ Z RUGV WH[ WXUH WH[ WXUH V\QWKHVLV J HQHWLF DOJ RULWK P QRQSDUDP HWULF SDUWLD00\ RUGHUHG

O DUNRY P RGH0V IRRW SDWWHUQ

**ALGORITMOS GENÉTICOS DA MÍNIMA ENTROPIA  
E DOS MODELOS DE TEXTURAS MODIFICADOS  
APLICADOS À SÍNTESE DE TEXTURA**

**SUMÁRIO**

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Introdução	1
1.2	Textura	3
1.3	Síntese de Textura	8
1.4	Levantamento Bibliográfico dos Trabalhos Desenvolvidos em Síntese de Textura	9
1.5	Proposta deste Trabalho	10
1.6	Estrutura dessa Dissertação	11
1.7	Considerações Finais deste Capítulo	12
<b>2</b>	<b>Fundamentos em Processamento de Imagens Digitais</b>	<b>13</b>
2.1	Introdução	13
2.2	A Digitalização de uma Imagem	13
2.3	Vizinhança de um Pixel	15
2.4	Medidas de Distância	16
2.5	Representação de uma Imagem Digital	18
2.6	O Histograma de Imagens Digitais	20
2.7	Considerações Finais deste Capítulo	23

<b>3</b>	<b>Conceitos Básicos de Algoritmos Genéticos</b>	<b>24</b>
3.1	Introdução	24
3.2	Diferenças dos Algoritmos Genéticos em Relação aos Métodos Comuns de Busca	26
3.3	Introdução a Algoritmos Genéticos	27
3.3.1	Avaliação e seleção de AGs	29
3.3.2	Reprodução ou geração de descendentes pelo AG	31
3.3.2.1	& URVVRYHU	31
3.3.2.2	Mutação	33
3.3.3	Um exemplo simples de algoritmo genético	34
3.4	Operadores de Reordenamento em Algoritmos Genéticos	41
3.4.1	3 DUWLD00\ O DWFKHG & URVVRYHU (PMX)	42
3.4.2	Inversão	44
3.5	Algoritmos Genéticos com Valores Contínuos	44
3.5.1	Operadores de FURVVRYHU para valores contínuos	45
A)	Método de 5 DGF0LII	45
B)	Método de : ULJ KW ou método da extrapolação	46
C)	Método da combinação de FURVVRYHU e extrapolação	47
3.5.2	Operador de mutação para valores contínuos	48
3.6	Operador Genético de Elitismo	49
3.7	Algoritmo Guloso (* UHHG\ \$ OJ RULWKP )	49
3.8	Considerações Finais	50

<b>4 Algoritmos Genéticos Aplicados à Síntese de Textura</b>	<b>52</b>
4.1 Introdução	52
4.2 Definições Básicas	54
4.2.1 Entropia	55
4.2.2 Campos aleatórios de Markov (O DUNRY UDQGRP ILHOGV O 5)	57
4.2.2.1 Modelos de Markov parcialmente ordenados (SDUWLD00\ RUGHUHG O DUNRY P RGH0V)	59
4.3 Introdução ao Método Desenvolvido neste Trabalho	61
4.3.1 ) RRW SDWWHUQ	61
4.3.2 Histograma de eventos e entropia	62
4.3.3 A estrutura do POMM	64
4.3.4 O vetor de probabilidades dos modelos de textura utilizados neste trabalho	65
4.3.5 Gerando amostras a partir de vetores de modelos estocásticos de textura	66
4.4 Método de Síntese de Textura pelo Algoritmo Genético Tandem que utiliza Modelos de Markov Parcialmente Ordenados Não Paramétricos	67
4.4.1 Algoritmo genético da mínima entropia	67
4.4.2 Algoritmo genético dos modelos de textura	70
4.5 Método de Síntese de Textura pelo Algoritmo Genético Tandem que utiliza Modelos de Markov Parcialmente Ordenados Não Paramétricos Modificado	73
4.5.1 Algoritmo genético da mínima entropia modificado	74
4.5.2 Algoritmo genético dos modelos de textura modificado	78
4.6 Considerações Finais deste Capítulo	80

<b>5 Resultados Obtidos</b>	81
5.1 Introdução	81
5.2 Amostras Originais	81
5.3 Resultados Obtidos no AGMEM	83
5.4 Resultados Obtidos no AGMTM	92
5.5 Análise dos Resultados Obtidos do AGMEM e do AGMTM	104
5.5.1 Análise dos resultados obtidos do AGMEM	104
5.5.2 Análise dos resultados do AGMTM.	109
5.5.3 Tempo total de processamento dos algoritmos AGME e AGMT e dos algoritmos AGMEM e AGMTM	116
5.6 Conclusões	118
<b>6 Conclusões, Contribuições deste Trabalho e Sugestões para Futuros Trabalhos</b>	122
6.1 Conclusões	122
6.2 Contribuições deste Trabalho	124
6.3 Sugestões para Trabalhos Futuros	125
<b>Referências Bibliográficas</b>	126

## LISTA DE FIGURAS

1.1 – (a) e (b) texturas de tecidos; (c) textura de um muro de tijolos; (d) textura de pele de animal; (e) textura de madeira; (f) textura artificial	7
2.1– (a) Vizinhança $N_4(p)$ , (b) Vizinhança $N_D(p)$ , (c) Vizinhança $N_8(p)$	16
2.2– (a) Exemplo de imagem binária, (b)valores de níveis de cinza dos pixels na Figura 2.1(a)	18
2.3– Imagem da Lenna com diferentes escalas de níveis de cinza: (a) 256 níveis de cinza,(b) 16 níveis de cinza, (c) 8 níveis de cinza, (d) 2 níveis de cinza	20
2.4– (a) Imagem com alto contraste, (b) histograma da imagem da Figura 2.4(a), (c) imagem escura com baixo nível de contraste e (d) histograma da imagem da Figura 2.4 (c)	22
2.5– (a) e (c) Imagens com baixo contraste; (b) e (d) histogramas das imagens da Figuras 2.5(a) 2.5(c)	23
3.1 – Exemplos de indivíduos ou cromossomos: (a) vetor de números binários; (b) vetor de números inteiros; (c) vetor de números reais; (d) string de caracteres alfabéticos	28
3.2 – Exemplo de representação pelo método da roleta para o exemplo da Tabela 3.1	30
3.3 – Crossover entre os indivíduos A e C gerando dois novos indivíduos A' e C'	32
3.4 – Exemplo de mutação ocorrida no 4º gene do indivíduo A' gerando o novo indivíduo A''	34
3.5 – Gráfico da função $f(x)$ definida na Equação 3.1	35

3.6 – Gráfico da função $f(x)$ , população inicial e 1 <sup>a</sup> geração de indivíduos	38
3.7 – Gráfico da função $f(x)$ , 4 <sup>a</sup> e 5 <sup>a</sup> gerações	39
3.8 – Gráfico da função $f(x)$ , 9 <sup>a</sup> e 10 <sup>a</sup> gerações	40
3.9 – Exemplo de PMX entre os Indivíduos A e B de números inteiros	43
3.10 – Exemplo de inversão no cromossomo A''	44
3.11 – Exemplo de mutação para valores contínuos	48
4.1 – Exemplo de um $IRRW_{SDWWHUQ}$ (7,8)	61
4.2 – Construção do histograma de eventos	63
4.3 – Estrutura da POMM utilizada neste trabalho	64
4.4 – Estrutura do vetor do modelo estocástico de textura utilizado neste trabalho	65
4.5 – (a) representação de um $IRRW_{SDWWHUQ}$ pelos valores das distâncias ao centro da janela; (b) representação de um $IRRW_{SDWWHUQ}$ na forma espacial; (c) representação de um $IRRW_{SDWWHUQ}$ por um vetor usado no algoritmo como indivíduo	68
4.6 – Diagrama simplificado do AGME	70
4.7 – Diagrama simplificado do AGMT	73
4.8 – (a) representação de um $IRRW_{SDWWHUQ}$ por um vetor que representa cada indivíduo; (b) representação espacial de um $IRRW_{SDWWHUQ}$	75
4.9 – Diagrama simplificado do AGMEM	77
4.10 – Diagrama simplificado do AGMTM	79
5.1 – Cinco amostras de texturas utilizadas neste trabalho: (a) $SLSHV$ ; (b) $GLDJRQD0V$ ; (c) $IL0DPHQWV$ ; (d) $WKLFNGLDJRQD0V$ ; (e) $F0RXGV$	82
5.2 – (a) Gráfico da evolução da população do AGMEM para a amostra de textura $SLSHV$ ; (b) melhor $IRRW_{SDWWHUQ}$ encontrado para essa textura	85
5.3 – (a) Gráfico da evolução da população do AGMEM para amostra de textura	

GLDJ RQD0V; (b) melhor IRRW SDWWHUQ encontrado para essa textura	86
5.4 – (a) Gráfico da evolução da população do AGMEM para amostra de textura ILODP HQWV; (b) melhor IRRW SDWWHUQ encontrado para a textura ILODP HQWV	87
5.5 – (a) Gráfico da evolução da população do AGMEM para amostra de textura WKLFN GLDJ RQD0V; (b) melhor IRRW SDWWHUQ encontrado para essa textura	89
5.6 – (a) Gráfico da evolução da população do AGMEM para a amostra de textura FORXGV; (b) melhor IRRW SDWWHUQ encontrado para essa textura	90
5.7 – (a) Gráfico da evolução da população para a textura de SLSHV; (b) IRRW SDWWHUQ utilizado pelo AGMTM para essa textura	94
5.8 - (a) Amostra original da textura SLSHV; (b) amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do AGMTM	95
5.9 – (a) Gráfico da evolução da população para a textura GLDJ RQD0V; (b) IRRW SDWWHUQ utilizado pelo AGMTM para essa estrutura	96
5.10 - (a) Amostra original da textura GLDJ RQD0V; (b) amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do AGMTM	97
5.11 – (a) Gráfico da evolução da população da textura ILODP HQWV; (b) IRRW SDWWHUQ utilizado pelo AGMTM para essa estrutura	98
5.12 - (a) Amostra original da textura ILODP HQWV; (b) amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do AGMTM	99
5.13 – (a) Gráfico da evolução da população da textura WKLFN GLDJ RQD0V; (b) IRRW SDWWHUQ utilizado pelo AGMTM para essa textura	100
5.14 - (a) Amostra original da textura WKLFN GLDJ RQD0V; (b) amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do AGMTM	101
5.15 – (a) Gráfico da evolução da população da textura FORXGV;	

(b) IRRW SDWHUQ utilizado pelo AGMTM para essa textura	102
5.16 - (a) Amostra original da textura FORXGV; (b) amostra dessa textura sintetizada usando o modelo estocástico obtido do AGMTM	103
5.17 – Valores da entropia dos IRRW SDWHUQV obtidos do AGME e do AGMEM	104
5.18 – Tempo total de processamento do AGME e do AGMEM para encontrar o IRRW SDWHUQ com a menor entropia	107
5.19 - Valor da diferença do melhor indivíduo e da média das diferenças da população inicial até as gerações 100 e 200	113
5.20 – Erros entre o modelo estocástico encontrado pelo AGMTM e o modelo da imagem sintetizada do AGMT, em relação ao modelo estocástico da amostra da textura da imagem original	114
5.21 – Tempo total de processamento dos algoritmos AGME e AGMT e dos algoritmos AGMEM e AGMTM	117

## LISTA DE TABELAS

Tabela 1.1 – Recursos computacionais utilizados neste trabalho	11
Tabela 3.1 – Valores de aptidão de alguns cromossomos em um AG	29
Tabela 3.2 – Valores relativos à evolução da população mostrada nas Figuras 3.6, 3.7 e 3.8	41
Tabela 5.1 – Valores utilizados no AGMEM e AGME	84
Tabela 5.2 – Valores utilizados no AGMT e AGMTM	93
Tabela 5.3 – Valores de entropia obtidos do AGME e AGMEM	105
Tabela 5.4 – Comparação do tempo total de processamento do AGME e AGMEM	107
Tabela 5.5 – Resultados obtidos nos testes realizados com o AGMTM	112
Tabela 5.6 – Erros entre o modelo estocástico encontrado pelo AGMTM e o modelo da imagem sintetizada do AGMT, em relação ao modelo estocástico da amostra da textura da imagem original	115
Tabela 5.7 – Comparação do tempo total de processamento entre os algoritmos AGMEM e AGMTM e os algoritmos AGME e AGMT	117

# CAPÍTULO I

## INTRODUÇÃO

### 1.1 – Introdução

As imagens são importantes fontes de informação sobre aspectos e características de objetos, lugares, pessoas e animais. Para que essas imagens possam ser utilizadas no computador é necessário que elas sejam digitalizadas.

As imagens digitais ajudam em muitas áreas de pesquisa. Na área de medicina e odontologia, as imagens são utilizadas para auxiliar no diagnóstico de doenças, cirurgia a longa distância, entre outras aplicações. Na área da geografia e da meteorologia as imagens de satélites digitalizadas, por exemplo, ajudam nas previsões do tempo e no reconhecimento de regiões. Filmes, jogos e realidade virtual usam constantemente imagens digitais tentando reproduzir em um computador de maneira consistente, para que elas sejam tão parecidas com as imagens reais.

A digitalização das imagens possibilita que as mesmas possam ser trabalhadas e melhoradas no computador realçando as características de maior interesse dependendo da aplicação que elas serão utilizadas. O processamento de imagens digitais possibilita ao usuário obter as informações desejadas da imagem selecionada. Várias técnicas podem ser utilizadas dependendo apenas do objetivo final. Essas técnicas são:

- síntese de textura;
- realce de imagem;

- segmentação de imagem;
- restauração de imagem;
- compressão de imagem, e;
- etc.

Um dos tipos de imagens digitais mais utilizadas em processamento de imagens são as imagens de alguns tipos de texturas. A textura é um termo que se refere às propriedades que representam a superfície de um objeto. Apesar de ser fácil a sua visualização, é muito difícil definir precisamente o que significa textura de um objeto ou de uma imagem [1]. Não existe uma definição formal de textura, porém ela pode ser definida de maneira mais simplificada como um conjunto de determinados padrões que podem ocorrer de forma repetitiva ou não [2]. Os padrões podem ser vários. Por exemplo, um tecido xadrez possui uma estrutura ou padrão que se repete continuamente em todo o tecido. Já uma tábua de madeira possui um conjunto de padrões não repetitivos. Existem várias definições de textura, mas nenhuma que possa ser usada e aceita por todos. Na visão computacional procura-se uma fórmula ou um modelo geral que seja capaz de descrever uma grande variedade de texturas em um único trabalho [3].

A textura é um atributo de tratamento um pouco mais apurado, são desenvolvidos vários trabalhos relativos a sua análise e síntese. Alguns desses trabalhos são motivados pelas descobertas na área científica em relação à visão humana, outros pela sua aplicação em determinadas áreas, principalmente em computação gráfica, realidade virtual e em análise de imagens de satélites.

A síntese de textura é um dos grandes desafios atuais. Os métodos de síntese de textura sempre restringem os tipos de texturas a serem modeladas devido à dificuldade de

obter um método genérico de síntese para uma grande variedade de texturas existentes. Um dos objetivos de síntese de textura é a partir de uma amostra da textura original tentar produzir uma amostra de textura que seja o mais próximo possível e possua as mesmas características da amostra original. A partir de um método eficiente de síntese de texturas não é preciso armazenar imagens de texturas que ocupam muito espaço. É necessária apenas uma amostra da textura original e reconstruí-la à medida que seja preciso utilizá-la. Existe um grande número de trabalhos abordando esse assunto e vários métodos propostos.

A motivação desta dissertação é trabalhar com síntese de texturas com o objetivo de reproduzir as amostras das texturas fornecidas de forma que a amostra da textura sintetizada ao final do processo seja o mais próximo possível e possua as mesmas características da amostra da textura original.

Este capítulo mostra as várias definições de textura e os aspectos relacionados à sua síntese. Apresenta um levantamento bibliográfico dos trabalhos desenvolvidos em síntese de textura, a proposta deste trabalho e a estrutura desta dissertação. Finalmente são realizadas considerações finais deste capítulo.

## **1.2 – Textura**

O que vem a ser textura? Esta é uma pergunta bem difícil de ser respondida. Quando visualizada é fácil reconhecê-la, porém é difícil defini-la. Embora exista um grande número de definições existentes sobre esse assunto nenhuma é universalmente aceita. Pode-se dizer que a textura é um conjunto de detalhes existentes na superfície de um objeto ou de uma imagem que se traduz em um grande número de informações a respeito da forma da superfície da imagem ou do objeto [4]. Porém devido ao grande número de definições é importante

buscar na literatura referente à visão computacional e processamento de imagens outros tipos de definições [1]. Essas definições são:

- a definição de Sklansky [1] de textura é “uma região, em uma imagem, possui uma textura constante se um conjunto de estatísticas locais ou outras propriedades locais da função da figura são constantes, de lenta variação, ou de variação aproximadamente periódica”;
- Tamura e outros [1] disseram que “pode-se considerar como textura o que constitui uma região macroscópica. Sua estrutura é praticamente atribuída a padrões repetitivos em que os elementos ou as primitivas são arranjados de acordo com uma regra de colocação”;
- Zucker, e Kant [1], citaram que “a noção de textura é aparentemente paradoxal. De um lado é geralmente utilizada para processamento prévio da informação visual, especialmente para propósitos de classificação prática. De outro lado ninguém conseguiu produzir uma definição de textura aceita comumente. A resolução desse paradoxo depende de um modelo mais rico, mais desenvolvido de processamento da informação visual e de um aspecto central que represente os sistemas em diferentes níveis de abstração. Esses níveis, provavelmente, incluirão intensidades reais de fundo e progredirão em direção aos descritores de bordas e orientação para a superfície, e talvez descritores volumétricos. Consideradas as estruturas de vários níveis, parece claro que elas deveriam ser incluídas na definição e na computação dos descritores de textura”;

- Wilson e Spann [1] disseram que “regiões de textura são padrões espacialmente estendidos baseados na maior ou na menor repetição precisa de alguma unidade celular (texton ou subpadrão)”;
- Jain [1] citou que “o termo textura geralmente refere-se à repetição de elementos básicos de textura chamados texels. O texel contém vários pixels, cuja colocação pode ser aleatória, quase-periódica ou, periódica. Texturas naturais são geralmente aleatórias, ao passo que texturas artificiais são freqüentemente determinísticas ou periódicas. A textura pode ser áspera, fina, suave, granulada, ondulada, irregular, regular, ou linear”;
- O Instituto de Engenheiros Elétricos e Eletrônicos *IEEE Standart Glossary of Image Processing and Pattern Recognition* [4] define “textura como um atributo que representa o arranjo espacial dos níveis de cinza dos pixels em uma região”;
- Gonzalez e Woods [5] disseram que “intuitivamente nota-se a textura como um descritor provedor de uma medida de propriedades tal como suavidade, asperidade, e regularidade”, e,;
- Jahne [1] definiu que em processamento de imagens texturas são modelos que caracterizam os objetos.

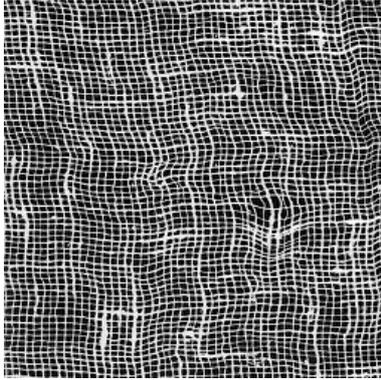
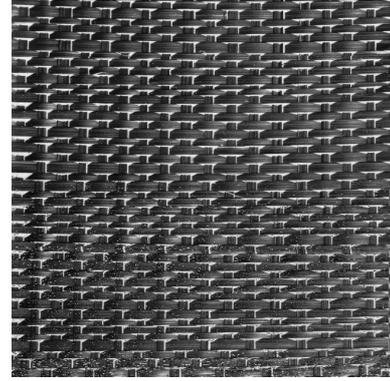
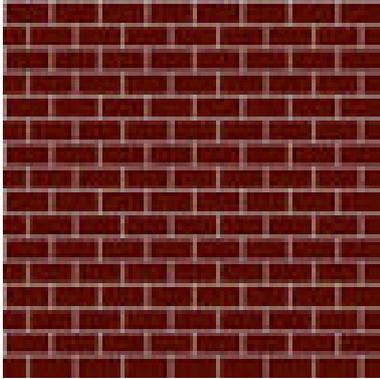
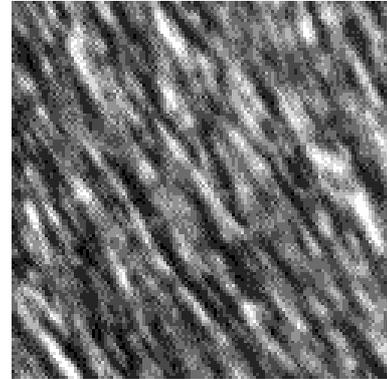
Apesar de não existir uma definição de textura aceita universalmente, pelas definições citadas acima pode-se concluir que:

- existe uma variação significativa na intensidade dos níveis de coloração entre os pixels próximos, ou seja, no limite da resolução da imagem, não existe homogeneidade [4], e;
- textura é uma propriedade homogênea a uma resolução espacial que é maior do que a resolução da imagem [4].

Das conclusões citadas acima, pode-se dizer que a textura depende da distância em que se observa o objeto ou a imagem. Por exemplo, se um tecido xadrez é observado a uma certa distância, a textura desse tecido pode ser considerada como sendo o xadrez ou em alta resolução, se for olhado bem próximo, a textura do tecido que o constitui. Um muro de tijolos se observado mais distante pode ser caracterizado em relação à textura pelos tijolos que o compõem ou em alta resolução, ou seja, bem próximo, pelos detalhes da forma de cada tijolo.

As texturas podem ser classificadas em artificiais e naturais [2]. As texturas artificiais são aquelas que possuem um arranjo simbólico sobre um fundo neutro, ou seja, são as texturas criadas. Já as texturas naturais são representadas pelos cenários encontrados tais como grama, areia, tecido, etc. Portanto a definição de textura a ser utilizada depende muito da sua aplicação.

A Figura 1.1 mostra alguns exemplos de texturas, sendo as Figuras 1.1(a), 1.1(b) e 1.1(c) texturas naturais e periódicas, as Figuras 1.1(d) e 1.1(e) naturais e não periódicas e a Figuras 1.1 (f) textura artificial e não periódica.

*a**b**c**d**e**f*

*Figura 1.1 – (a) e (b) texturas de tecidos; (c) textura de um muro de tijolos; (d) textura de pele de animal; (e) textura de madeira; (f) textura artificial.*

### 1.3 - Síntese de Textura

A síntese de textura consiste em gerar uma textura com base em alguma representação da textura ou pelo uso de uma amostra da textura original.

A síntese de textura a partir de uma textura original (imagem fonte) é de extrema importância em vários campos de pesquisa científica:

- no campo da computação gráfica o uso de imagens geradas por amostras de texturas traz um aspecto muito mais realístico as cenas que podem ser utilizadas em jogos, comerciais, aplicações militares, simuladores e até em cenas de realidade virtual. Além disso, é mais vantajoso armazenar apenas amostras da textura utilizada como fonte do que uma imagem completa que requer um grande espaço de armazenamento, e;
- na geração de texturas similares que são originadas da mesma fonte e que possuem características semelhantes, porém não são idênticas.

Os modelos de texturas mais utilizados para segmentar, classificar e sintetizar texturas são:

- Fractals [7];
- Markov Random Field (MRF) [1];
- Morfologia Matemática [7];

#### 1.4 – Levantamento Bibliográfico dos Trabalhos Desenvolvidos em Síntese de Textura

Os métodos de síntese de texturas propostos na literatura têm-se concentrado em grande parte na teoria de fractais [4]. Eles utilizam o conceito de movimento fractal browniano (*fbm*) para simular as variações dos valores relativos à textura [4]. De qualquer maneira, os métodos fractais utilizam sempre geradores pseudoaleatórios o que permite classificá-los como métodos de natureza estocástica. Entretanto, apesar de serem capazes de gerar amostras de texturas de boa similaridade, esses métodos não são capazes de reconstituir um determinado padrão de textura [8].

Ainda existem texturas baseadas em modelos evolutivos, que são métodos sistemáticos e complexos desenvolvidos por Haralick [9], que simulam o processo evolutivo da natureza. Esses modelos introduziram três operações básicas aplicadas à “células”, que são a distribuição de sementes, o crescimento da estrutura (ou do esqueleto) e o crescimento muscular.

CROSS e JAIN [7] propuseram um método de geração de texturas baseado na teoria estatística. O modelo de textura utilizado nesse método é baseado em campos aleatórios de Markov que é um procedimento matemático capaz de produzir e descrever uma imagem de textura. Esse método tem sido utilizado, principalmente, para a síntese de texturas baseadas em níveis de cinza, produzindo texturas utilizando-se cadeias de Markov. Porém, o processamento computacional desse método é elevado.

ZHU e outros [3] apresentaram um modelo de síntese de textura que combina a teoria de filtros e modelamento de campos aleatórios de Markov utilizando a teoria da máxima entropia. O princípio desse modelo baseia-se na tentativa de reprodução, o mais próxima possível das amostras das texturas fornecidas. Como citado no parágrafo anterior devido ao

uso de campos aleatórios de Markov o processamento computacional desse modelo é muito alto.

DAVIDSON e outros [10] apresentaram um modelo estocástico para síntese de textura. Esse modelo utiliza uma subclasse de campos aleatórios de Markov chamada de modelos de Markov parcialmente ordenados (*POMMs*). O *POMM* possui algumas simplificações em relação à teoria de Markov. Devido a essas simplificações, o processamento computacional diminui bastante em relação aos modelos baseados na teoria de Markov.

Um dos grandes problemas atuais para sintetizar uma textura a partir de uma amostra é o elevado tempo de processamento computacional.

### **1.5 – Proposta deste Trabalho**

Este trabalho propõe e implementa modificações e adaptações no método de síntese de textura baseado na teoria de *foot patterns*, *POMMs* e algoritmos genéticos desenvolvidos por Ashlock e Davidson [11]. O principal objetivo desta dissertação é obter um método de síntese que a partir de uma amostra de qualquer textura, construa uma textura a mais próxima possível e com as mesmas características da textura original, ou seja, com o menor erro em relação a essa textura. Outro objetivo deste trabalho é diminuir o tempo de processamento do algoritmo para sintetizar uma textura a partir apenas da sua amostra.

A Tabela 1.1 mostra os recursos computacionais utilizados neste trabalho.

*Tabela 1.1 – Recursos computacionais utilizados neste trabalho.*

CPU	AMD Athlon XP+2200, 1.8 GHz, 256 MB RAM DDR
Sistema Operacional	Windows XP Professional
Ferramenta de Programação	Matlab 6.5
Dispositivos de saída	Monitor LG 15’’ Tela plana Studioworks 560N

### **1.6 – Estrutura desta Dissertação**

Neste capítulo são mostradas as várias definições de textura e os aspectos relacionados à sua síntese. São apresentados um levantamento bibliográfico dos trabalhos desenvolvidos em síntese de texturas, a proposta deste trabalho e a estrutura desta dissertação.

O capítulo 2 descreve a representação, a amostragem e a quantização de uma imagem. São apresentadas a vizinhança de um pixel, as medidas de distância entre os pixels e a construção do histograma de uma imagem. Finalmente são realizadas considerações finais desse capítulo.

O capítulo 3 descreve os conceitos básicos de algoritmos genéticos. São apresentadas as representações, a formulação e a resolução de problemas utilizando essa técnica. Finalmente são realizadas considerações finais desse capítulo.

O capítulo 4 mostra o método proposto e implementado neste trabalho e suas definições específicas. São apresentadas as definições básicas de entropia, histograma de eventos, *foot patterns* e *POMMs*. Ele mostra o método desenvolvido por Ashlock e Davidson

[11] e suas definições. É apresentado um modelo de representação e síntese de texturas com dois níveis de cinza baseado no método de Ashlock e Davidson [11] usando apenas uma amostra da textura original. Finalmente são realizadas as considerações finais desse capítulo.

O capítulo 5 mostra os resultados obtidos ao utilizar o modelo proposto e implementado descrito no capítulo 4 deste trabalho. São mostrados os resultados da entropia dos *foot patterns* obtidos do algoritmo genético da mínima entropia modificado (AGMEM) e as imagens sintetizadas pelos modelos estocásticos encontrados pelo algoritmo genético dos modelos de texturas modificado (AGMTM). Finalmente são realizadas conclusões sobre os resultados obtidos nesses testes.

Finalmente, o capítulo 6 apresenta as conclusões e as contribuições deste trabalho e as sugestões para futuros trabalhos.

### **1.7 – Considerações Finais deste Capítulo**

Este capítulo mostrou as várias definições de textura e os aspectos relacionados à sua síntese. Foram apresentados um levantamento bibliográfico dos trabalhos desenvolvidos em síntese de textura e a proposta deste trabalho. Finalmente, foi mostrada a estrutura desta dissertação.

# CAPÍTULO II

## FUNDAMENTOS EM PROCESSAMENTO DE IMAGENS DIGITAIS

### 2.1 – Introdução

Como citado no capítulo 1 deste trabalho as imagens são importantes fontes de informação sobre aspectos e características de objetos, lugares, pessoas e animais. Para que essas imagens possam ser utilizadas no computador é necessário que elas sejam digitalizadas.

As imagens digitalizadas podem ser processadas e características delas que sejam importantes para as pesquisas e análises podem ser realçadas. Noções sobre a representação, distâncias entre pixels, vizinhança de pixels e histogramas são importantes na compreensão e na análise de qualquer problema que realiza o processamento de imagens.

Este capítulo descreve a representação, a amostragem e a quantização de uma imagem. Apresenta a vizinhança de um pixel, as medidas de distância entre os pixels e o histograma de uma imagem. Finalmente são realizadas considerações finais deste capítulo.

### 2.2 – A Digitalização de uma Imagem

Segundo Gonzalez e Woods [5] para um melhor processamento computacional a função  $f(x,y)$  de uma imagem deve ser digitalizada espacialmente e em amplitude. A digitalização espacial é chamada de amostragem da imagem e a digitalização de amplitude de quantização de níveis de cinza.

Considere uma imagem contínua  $f(x,y)$  aproximada por amostras igualmente espaçadas e representadas por uma matriz  $N \times M$ , como mostrado na Equação (2.1).

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \cdot & \cdot & \cdot & f(0,M-1) \\ f(1,0) & f(1,1) & \cdot & \cdot & \cdot & f(1,M-1) \\ \cdot & & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ f(N-1,0) & f(N-1,1) & \cdot & \cdot & \cdot & f(N-1,M-1) \end{bmatrix} \quad (2.1)$$

O lado direito da Equação (2.1) representa a imagem digital. Cada elemento dessa matriz é chamado de pixel, abreviatura de *picture elements* (elementos da figura) [5].

Uma imagem digitalizada possui  $N$  linhas por  $M$  colunas e cada pixel pode ser armazenado com um determinado número de níveis de cinza ou cor ( $G$ ). Geralmente para facilitar o processamento essas quantidades são potências inteiras de dois, como mostrado na Equação (2.2). Se isso não ocorrer devido ao tamanho da imagem, no processamento dela é realizado o enchimento com zeros.

$$N = 2^n, M = 2^k \text{ e } G = 2^m \quad (2.2)$$

Na Equação (2.2) os valores de  $n$ ,  $k$  e  $m$  são números inteiros e positivos. Considerando-se essa equação, o número de bits  $b$  necessário para armazenar uma imagem é obtido pela Equação (2.3).

$$b = N \times M \times m \quad (2.3)$$

Pode-se observar na Equação (2.2) que quanto menor o número de níveis de cinza  $G$  ou a dimensão  $N \times M$  de uma imagem, menos detalhes se tem dessa imagem. A importância ou não desse fato depende da aplicação e da necessidade de definição da imagem a ser analisada. Quanto mais detalhes e definição uma imagem possui, maior é o número de níveis de cinza, maior é a dimensão e, também, maior é a quantidade de bits necessária para armazená-la.

### 2.3 - Vizinhança de um Pixel

Considere uma imagem representada por  $f(x,y)$ , onde  $f(x,y)$  é a matriz de níveis de cinza dessa imagem. Cada pixel é indicado por  $p(x,y)$ . Se o pixel  $p(x,y)$  possui pixels vizinhos na direção horizontal e na direção vertical cujas coordenadas são  $(x+1,y)$ ,  $(x-1,y)$ ,  $(x,y+1)$  e  $(x,y-1)$ , como mostrado na Figura 2.1 (a). Esse conjunto de pixels é chamado vizinhança 4 de  $p$  ( $N_4(p)$ ). Cada um desses pixels está a uma distância igual a um de  $p(x,y)$ . Se o pixel  $p(x,y)$  possui 4 pixels vizinhos que estão na direção diagonal com coordenadas  $(x-1,y-1)$ ,  $(x-1,y+1)$ ,  $(x+1,y-1)$  e  $(x+1,y+1)$ , como mostrado na Figura 2.1 (b), o conjunto desses pixels é denominado  $N_D(p)$ . Se o pixel  $p$  possui pixels vizinhos  $N_4(p)$  e  $N_D(p)$ , como mostrado na Figura 2.1 (c), a vizinhança desse pixel é chamada vizinhança dos oito vizinhos de  $p$  ( $N_8(p)$ ).

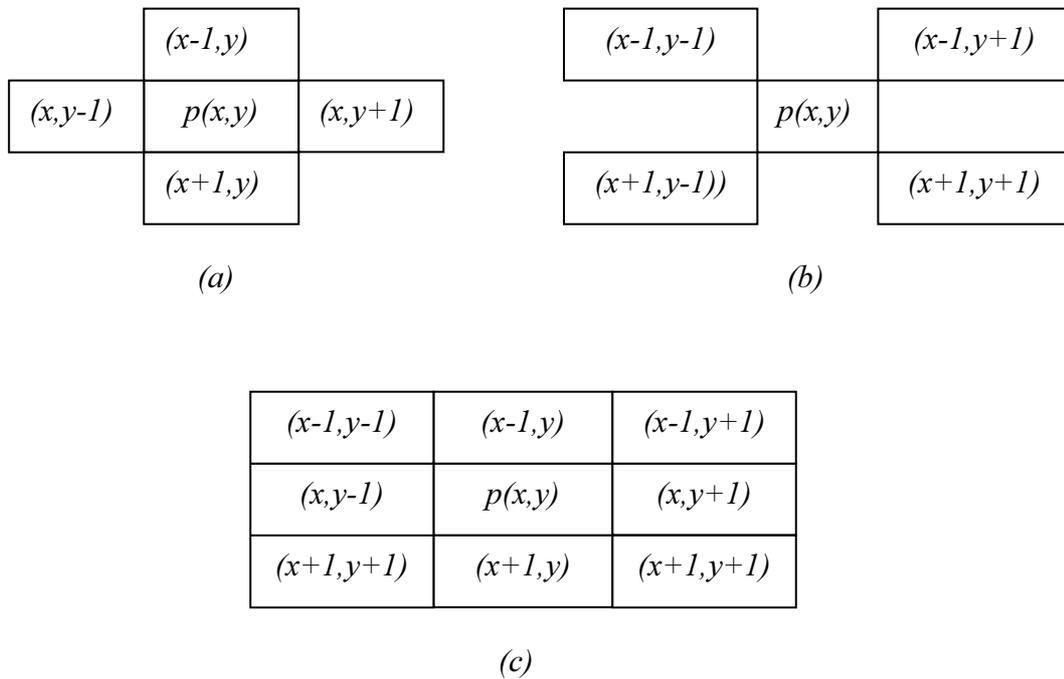


Figura 2.1 – (a) Vizinhança  $N_4(p)$ , (b) Vizinhança  $N_D(p)$ , (c) Vizinhança  $N_8(p)$ .

## 2.4 – Medidas de Distância

Considere os pixels  $q$  e  $z$  vizinhos do pixel  $p$ . Esses pixels possuem coordenadas  $(s, t)$ ,  $(u, v)$  e  $(x, y)$ , respectivamente.  $D$  é a distância ou métrica se [5]:

- $D(p, q) \geq 0$  (se  $D(p, q) = 0$  ( $p$  e  $q$  são coincidentes));
- A distância é a mesma se medida do pixel  $p$  ao pixel  $q$  ou do pixel  $q$  ao pixel  $p$ , ou seja,  $D(p, q) = D(q, p)$ , e;

- A distância do pixel  $p$  ao pixel  $t$  é sempre menor ou igual a soma das distâncias parciais de  $p$  a  $q$  mais a distância de  $q$  a  $z$ , ou seja,  $D(p,t) \leq D(p,q) + D(q,z)$ .

A distância Euclidiana entre os pixels  $p$  e  $q$  é calculada pela Equação (2.4).

$$D_e(p, q) = \sqrt{(x-s)^2 + (y-t)^2} \quad (2.4)$$

A distância  $D_4$  ou distância “city block” ou quarteirão entre os pixels  $p$  e  $q$  é obtida pela Equação (2.5).

$$D_4(p, q) = |x-s| + |y-t| \quad (2.5)$$

Quando  $D_4 = 1$  significa que os pixels  $p$  e  $q$  são pixels vizinhos  $N_4(p)$ .

A distância xadrez entre  $p$  e  $q$  é calculada pela Equação (2.6).

$$D_8 = \max(|x-s| + |y-t|) \quad (2.6)$$

Os pixels vizinhos com  $D_8 = 1$  são os oito pixels vizinhos de  $p$ .

## 2.5 – Representação de uma Imagem Digital

O termo imagem monocromática refere-se a uma função bidimensional  $f(x,y)$ , onde  $x$  e  $y$  são as coordenadas espaciais e o valor de  $f$  em qualquer ponto  $(x,y)$  é proporcional ao brilho (ou nível de cinza) [5].

Como citado anteriormente neste capítulo uma imagem digital geralmente é representada por uma matriz de valores. Cada valor da matriz corresponde a um pixel. O pixel é a menor subdivisão de uma imagem. O valor de cada pixel corresponde ao seu brilho ou a sua intensidade.

A Figura 2.2(a) mostra um exemplo de uma imagem binária, ou seja, uma imagem com apenas 2 níveis de cinza e a Figura 2.2(b) ilustra os valores dos pixels circundados em vermelho. Nessa ultima figura o valor um corresponde ao nível de cinza branco e o valor zero ao nível de cinza preto. Na Figura 2.2(a) considerou-se cada quadrado da imagem como um pixel.

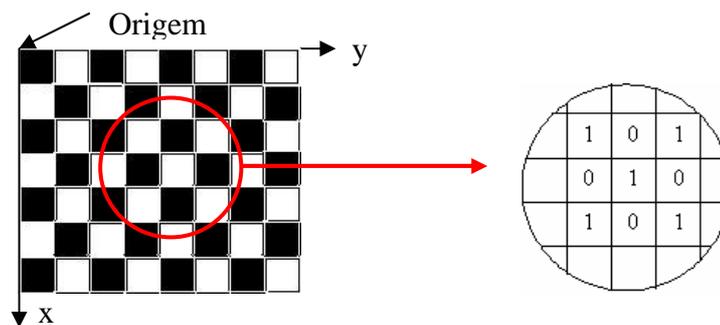


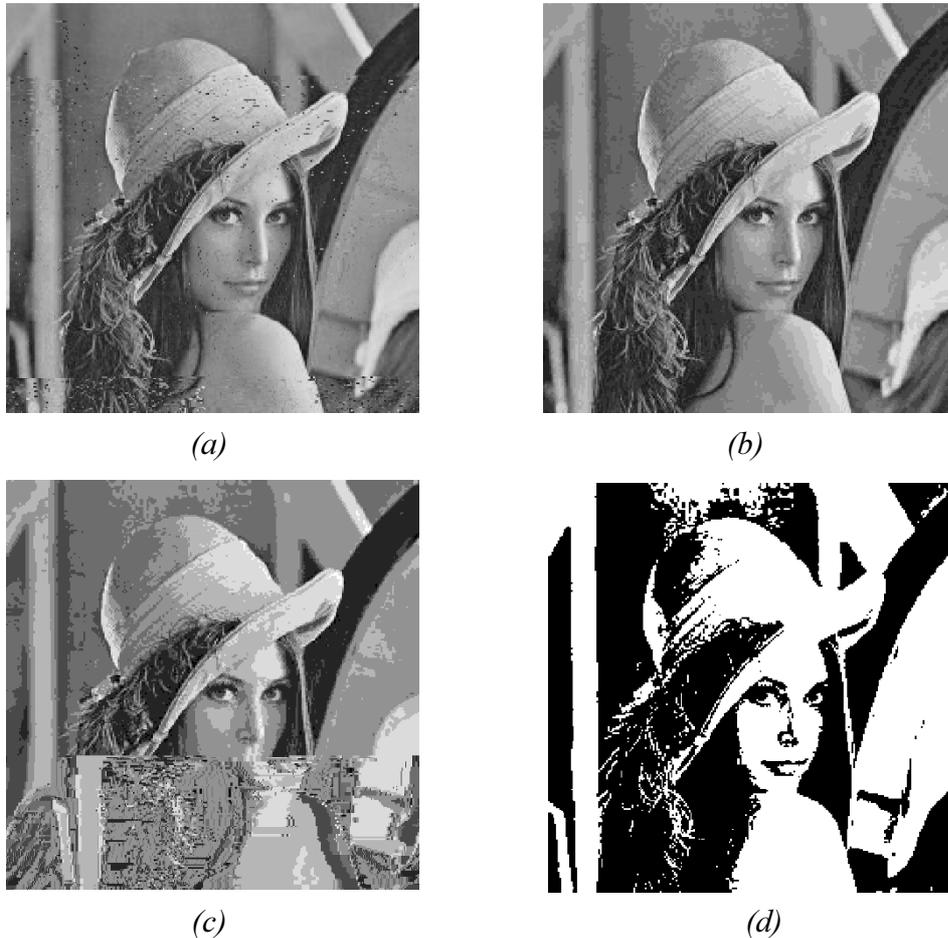
Figura 2.2 –(a) Exemplo de imagem binária; (b) valores de níveis de cinza dos pixels na

Figura 2.1(a).

O mais comum em trabalhos que utilizam imagens é que elas estejam com intensidades de níveis de cinza. Nessas imagens os valores de intensidade dos pixels variam de acordo com uma determinada escala de níveis de cinza. Geralmente as escalas de níveis de cinza possuem diferentes números de intensidades que são potências inteiras de dois (2, 4, 8, 16, 32, ...). Nessas escalas o valor zero corresponde à cor preta e o maior valor de intensidade à cor branca. Pode-se então ter uma imagem com níveis de cinza onde a intensidade de  $f(x, y)$  pode variar entre zero e um (imagem binária), zero e três (quatro níveis), zero e sete (oito níveis) e assim por diante, onde apenas os números inteiros são válidos como níveis de cinza.

A Figura 2.3 mostra a imagem da Lenna com diferentes escalas de intensidades de níveis de cinza.

As imagens coloridas possuem características parecidas com as imagens monocromáticas (ou níveis de cinza). A diferença dessas imagens está na quantidade de valores relacionados a cada pixel. Por exemplo, para o padrão de cor RGB (vermelho, verde e azul) cada pixel possui 3 valores de intensidade. Cada valor está relacionado a uma das cores. A combinação dessas três cores forma a maioria das cores existentes.



*Figura 2.3 – Imagem da Lenna com diferentes escalas de níveis de cinza: (a) 256 níveis de cinza, (b) 16 níveis de cinza, (c) 8 níveis de cinza, (d) 2 níveis de cinza.*

## **2.6 – O Histograma de Imagens Digitais**

O histograma é muito utilizado em análise e processamento de imagens digitais. Segundo Gonzalez e Wood [5] o histograma de uma imagem digital com níveis de cinza no intervalo de  $[0, L - 1]$ , no qual  $L$  é o número de níveis de cinza, é uma função discreta mostrada na Equação (2.7).

$$p(r_k) = n_k/n, \quad k = 0, 1, 2, \dots, L - 1. \quad (2.7)$$

onde:

$k$  – valores dos níveis de cinza;

$r_k$  -  $k$ -ésimo nível de cinza;

$n_k$  - número de pixels na imagem com o nível de cinza  $k$ ;

$n$  - total de pixels na imagem, e;

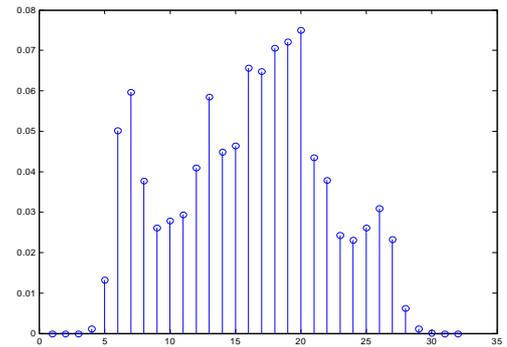
$p(r_k)$  - probabilidade de ocorrência do nível de cinza  $r_k$ .

O gráfico de  $p(r_k) \times r_k$  fornece uma descrição global da aparência de uma imagem.

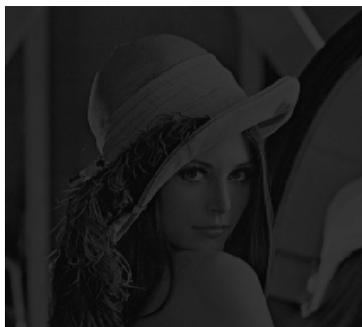
A Figura 2.4(a) mostra uma imagem 256 x 256 pixels com 32 níveis de cinza e a Figura 2.4(b) ilustra o histograma dessa imagem com os níveis de cinza variando de quatro a vinte e nove. Pode-se observar nessa figura o alto contraste dessa imagem. A Figura 2.4(c) mostra a mesma imagem da Figura 2.4(a), porém ela está mais escura e com pouca definição, o histograma dessa imagem é mostrado na Figura 2.4(d). Pode-se verificar nessa figura que os níveis de cinza variam apenas entre um e oito, produzindo uma imagem escura e com baixo contraste.



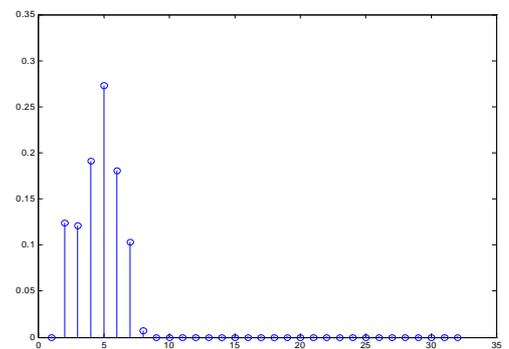
(a)



(b)



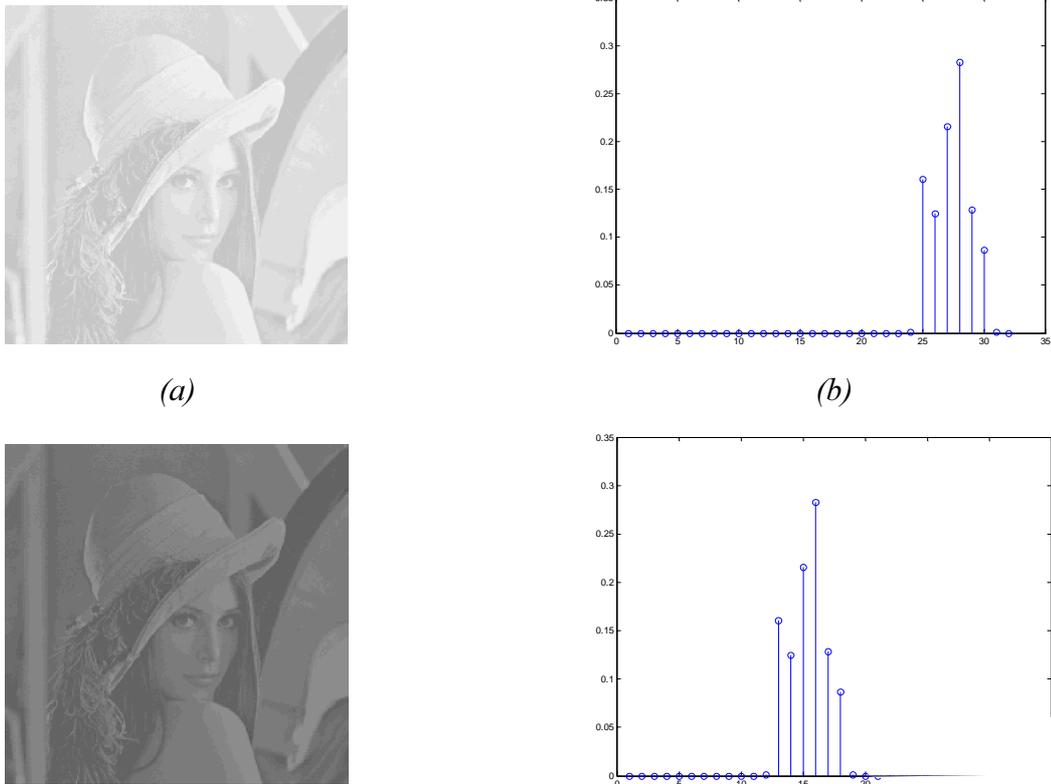
(c)



(d)

Figura 2.4 – (a) Imagem com alto contraste; (b) histograma da imagem da Figura 2.4(a); (c) imagem escura com baixo contraste; (d) histograma da imagem da Figura 2.4 (c).

As Figuras 2.5 (a) e 2.5 (c) mostram duas imagens com baixo contraste. A diferença entre essas imagens é o intervalo de variação de níveis de cinza. Na Figura 2.5 (b) pode-se verificar que no histograma os valores de níveis de cinza da imagem mostrada na Figura 2.5 (a) são superiores a vinte e cinco ilustrando que ela é uma imagem clara. Já no histograma mostrado na Figura 2.5 (d) os valores dos níveis de cinza estão mais centralizados. Isso significa que a imagem da Figura 2.5 (c) é mais escura do que a imagem da Figura 2.5 (a) e mais clara do que a da Figura 2.4 (c). Além do histograma fornecer a descrição global de uma imagem, pode fornecer uma informação extremamente útil para o realce do contraste da imagem.



*Figura 2.5 – (a) e (c) Imagens com baixo contraste; (b) e (d) histogramas das imagens das Figuras 2.5 (a) e 2.5(c).*

## 2.7 – Considerações Finais deste Capítulo

Este capítulo descreveu a representação, a amostragem e a quantização de uma imagem. Foi apresentada a vizinhança de um pixel, as medidas de distância entre os pixels e o histograma de uma imagem.

Neste capítulo foram apresentadas as técnicas básicas utilizadas em análise e em processamento de imagens.

Todas as técnicas e os métodos apresentados neste capítulo serão úteis quando elas forem aplicadas na análise e síntese de imagens digitais.

& \$ 3 È7 8 / 2 , , ,

& 2 1 & ( , 7 2 6 % È 6 , & 2 6 ' ( \$ / \* 2 5 , 7 0 2 6 \* ( 1 e 7 , & 2 6

± , QWURGX omR

Como citado por Goldberg [12] os algoritmos genéticos foram desenvolvidos por *John Holland*, seus colegas e seus estudantes da Universidade de Michigan. Os objetivos da pesquisa dele tem sido:

- ser um processo adaptativo abstrato de sistemas naturais e rigorosamente explicativo, e;
- desenvolvimento de *softwares* de sistemas artificiais que mantenham os importantes mecanismos do sistema natural.

Algoritmos genéticos são algoritmos de busca baseados em conceitos de seleção natural e genética humana. Eles combinam a sobrevivência das estruturas mais adequadas com uma troca aleatória de informações para formar um algoritmo de busca com algumas habilidades inovadoras da busca humana [12]. A cada nova geração são criadas novas estruturas usando partes mais adequadas das estruturas da geração anterior, buscando estruturas que mais se adequem ao objetivo proposto.

As características da biologia natural devem ser analisadas cuidadosamente. A forma de sobrevivência e adaptação dos seres ao ambiente e as condições em que eles são submetidos “selecionam” os seres mais adaptados para perpetuarem-se por mais gerações. Por

sua vez, os seres menos adaptados ao ambiente e as condições de vida vão desaparecendo aos poucos à medida que as gerações vão se sucedendo. A teoria de *Darwin* da Evolução das Espécies pela seleção natural cita:

“O conceito básico de seleção natural é que as condições ambientais (isto é, a natureza) determinam (ou selecionam) quão bem uma determinada característica de um organismo ajuda na sobrevivência e na reprodução desse organismo; organismos que não possuem essas características podem morrer antes de se reproduzirem ou serem menos prolíficos que os organismos que apresentam determinada característica. À medida que as condições ambientais não variem, ou permaneçam suficientemente similares, essas características continuam a serem adaptativas e elas tornar-se-ão mais comuns na população” [13].

Usando a teoria de *Darwin* citada no parágrafo anterior, o algoritmo genético é um método eficiente de busca principalmente em sistemas complexos que possuem um grande espaço de busca.

Um dos mais conhecidos problemas que pode ser solucionado eficientemente por algoritmos genéticos é o problema do “caixeiro viajante”. Existe um número de cidades e o caixeiro deve passar por cada uma dessas cidades uma só vez voltando no final à cidade que saiu percorrendo o menor percurso possível.

Este capítulo descreve os conceitos básicos de algoritmos genéticos. São apresentadas as representações, a formulação e a resolução de problemas utilizando essa técnica. Finalmente são realizadas considerações finais deste capítulo.

± ' LI HUHQoDV GRV \$ 0J RULWP RV \* HQpWLFrv HP 5 H0DomR DRV O pWRGRV & RP XQV GH %XVFD

Os algoritmos genéticos (AGs) diferem dos métodos mais comuns de busca em quatro pontos:

1. trabalham com uma série de parâmetros;
2. procuram várias soluções possíveis, ou seja, um conjunto de soluções não simplesmente uma única solução para o problema;
3. usam a informação de uma função objetivo ou função de aptidão e não outro conhecimento auxiliar. Essa função está relacionada com cada estrutura da população. Os valores retornados por essa função são usados na evolução da população, e;
4. utilizam regras de transição probabilística para direcionar a sua busca.

Esses quatro fatores juntos contribuem para obter um algoritmo genético robusto resultando em vantagens sobre as técnicas mais comuns utilizadas como, por exemplo:

- o espaço de busca é muito mais completo no AG do que nos métodos tradicionais, porque ele trabalha com uma série de parâmetros;
- recebe pouca influência das características locais dos problemas, e;

- ocorre uma significativa melhora porque ele utiliza a escolha aleatória como uma ferramenta para direcionar a procura por regiões do espaço de busca [12].

± , QWURGXomR D \$ UJ RULWP RV \* HQpWLFrv

A representação de um algoritmo genético simples não envolve métodos complexos ou complicados. Os termos utilizados nele são análogos aos da genética natural.

Basicamente os passos de um algoritmo genético são:

1. adequada codificação do problema;
2. geração da população inicial;
3. processo de avaliação da população que calcula o valor da aptidão de cada indivíduo da população;
4. processo de seleção que escolhe os indivíduos para a reprodução ou geração de descendentes;
5. reprodução e geração de descendentes onde são geradas novas soluções a partir das soluções já existentes pela aplicação de operadores genéticos, e;
6. com os descendentes retorna-se ao passo três e o processo é repetido até que o número de gerações desejadas seja alcançado.

Uma adequada codificação do problema depende das formas de representação das variáveis ou dos indivíduos. As formas de representação mais comuns de um indivíduo podem ser por uma *string* de caracteres alfabéticos, um vetor de números reais, um vetor de números inteiros ou um vetor binário onde, cada indivíduo tem uma dimensão fixa.

Em AGs a população inicial é criada aleatoriamente por uma distribuição uniforme e ela é o conjunto de possíveis soluções do problema. Cada uma dessas soluções é um indivíduo da população. Cada vetor ou *string* que representa um indivíduo é chamado de cromossomo. Cada elemento ou caractere do cromossomo é chamado de gene. O gene é a unidade básica do cromossomo. Portanto, um conjunto de genes forma um cromossomo que por sua vez representa um indivíduo. O conjunto de indivíduos é chamado de população. A Figura 3.1 mostra quatro exemplos de cromossomos de dimensão cinco, ou seja, com cinco genes cada um, com um dos genes de cada cromossomo em destaque.

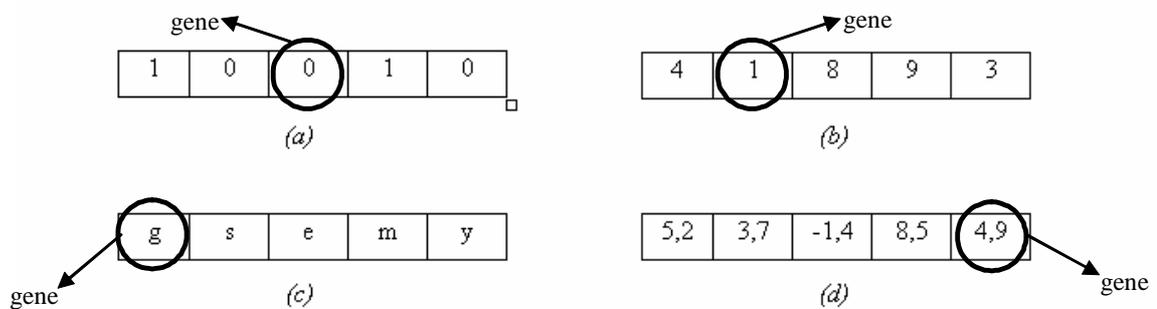


Figura 3.1 – Exemplos de indivíduos ou cromossomos:

- (a) vetor de números binários; (b) vetor de números inteiros;  
 (c) vetor de números reais; (d) string de caracteres alfabéticos.

± \$ YD0LDomR H VH0HomR GH \$ \* V

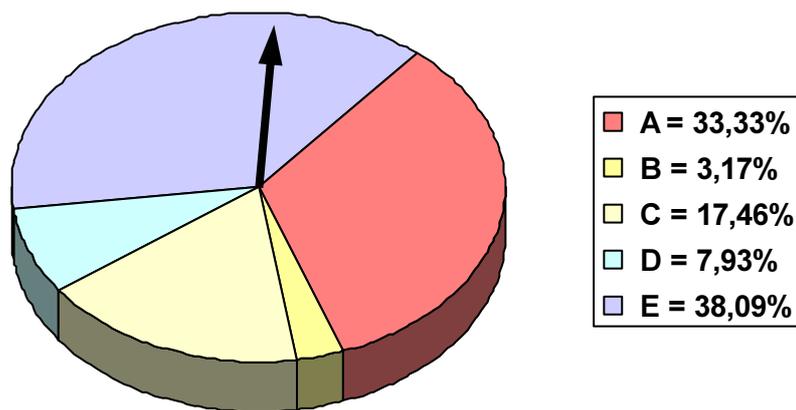
No algoritmo genético após a definição da população inicial ocorre a avaliação dos indivíduos. Nesta avaliação os indivíduos passam pela função de aptidão. Esta função é uma ferramenta importante nessa versão artificial de seleção natural. A função de aptidão retorna um valor que é relacionado ao que se busca, ou seja, a aptidão de cada cromossomo ou de cada indivíduo da população em relação ao objetivo da busca. Por exemplo, busca-se o máximo valor de uma string binária de dimensão cinco. O valor retornado pela função de aptidão é o valor em decimal correspondente a cada cadeia de caracteres binários. Os cromossomos que retornam os valores mais altos, de acordo com a função de aptidão, são os mais aptos e conseqüentemente possuem maior chance de perpetuarem suas características nas próximas gerações. A Tabela 3.1 ilustra um exemplo de alguns valores de aptidão para esse tipo de cromossomo. Nessa tabela as porcentagens mostradas referem-se ao quanto apto cada indivíduo é em relação à aptidão total da população.

*Tabela 3.1 – Valores de aptidão de alguns cromossomos em um AG.*

Indivíduos	<i>String</i>	Aptidão	% do total
A	10101	21	33,33
B	00010	2	3,17
C	01011	11	17,46
D	00101	5	7,93
E	11000	24	38,09
7 RWD0			

Pode-se observar na Tabela 3.1 que o indivíduo E é o que é mais apto dessa geração da população seguido depois pelos indivíduos A, C, D e B, respectivamente.

Com os valores de aptidão definidos, os indivíduos são selecionados. A seleção pode ser feita de várias formas. A mais simples é o uso do método da roleta. Com os valores de porcentagem atribuídos, como, por exemplo, ilustrado na Tabela 3.1, pode-se construir uma roleta como mostrado na Figura 3.2. Nesse método cada indivíduo com a sua respectiva porcentagem de aptidão é colocado na roleta para a seleção.



*Figura 3.2 – Exemplo de representação pelo método da roleta para o exemplo da Tabela 3.1.*

O número de indivíduos selecionados pelo método da roleta, geralmente, é o mesmo número total de indivíduos da população. Cada candidato à reprodução é escolhido com um giro da roleta, quando ela parar o indivíduo escolhido para a reprodução é aquele onde a seta está apontando. Quando o indivíduo é escolhido pela seleção uma cópia de seu cromossomo é feita e é essa cópia que vai para a reprodução. Assim um indivíduo pode ser escolhido mais de

uma vez, podendo-se ter indivíduos idênticos na reprodução. Depois de selecionar os indivíduos, a próxima etapa do AG é a reprodução.

± 5 HS URGX omR RX J HUDomR GH GHVFHQGHQWHV SHOR \$ \*

No AG são os operadores genéticos que modificam os dados e geram os novos descendentes. A cada geração ocorrem na população transformações que são realizadas por esses operadores. Os dois operadores mais utilizados para essas transformações são o *crossover* e a mutação.

#### ± *Crossover*

O *crossover* é o operador genético mais importante em AG. Ele é o responsável durante as gerações pela maior parte das mudanças nos indivíduos.

A operação de *crossover* é simples de ser realizada e sempre ocorre em pares. Primeiro, separa-se a população resultante da reprodução em pares. Então, uma posição  $p$  qualquer, denominada ponto de corte, é selecionada uniformemente entre 1 e o comprimento do cromossomo menos um ( $l - 1$ ). Então, ocorre uma troca de genes entre os cromossomos. Os genes da posição  $p + 1$  até  $l$  são trocados gerando dois novos indivíduos.

A Figura 3.3 mostra um exemplo de uma operação de *crossover*. Nessa figura um número aleatório  $p$  é escolhido entre 1 e 4. Supondo que  $p = 2$ , então são trocados os genes dos indivíduos A e C correspondentes às posições de 3 a 5.

1	0	1	0	1
---	---	---	---	---

0	1	0	1	1
---	---	---	---	---

1	0
---	---

1	0	1
---	---	---

0    1

Figura 3.3 – Crossover entre os indivíduos A e C gerando dois novos indivíduos A' e C'.

Na Figura 3.3 após ser realizado o *crossover*, dois novos indivíduos são gerados A' e C'. O indivíduo A' possui os dois primeiros genes iguais ao do seu ascendente A e os três últimos iguais ao seu ascendente C. O mesmo acontece com o indivíduo C', ou seja, seus dois

primeiros genes são iguais ao do ascendente C e os três últimos iguais ao do ascendente A. Os dois novos indivíduos A' e C' entram no lugar de seus pais na população.

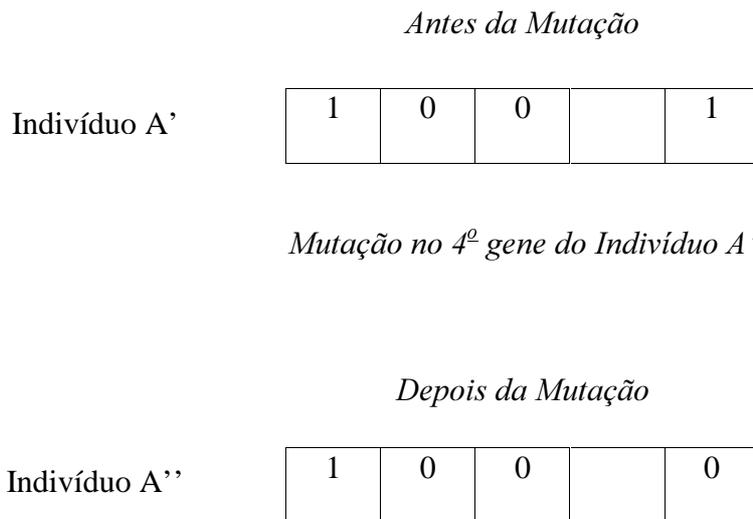
É importante deixar claro que nem sempre todos os indivíduos selecionados pela reprodução passam pelo *crossover*. Existe uma taxa que deve ser levada em conta antes de se realizar essa operação. Essa taxa é conhecida como taxa de *crossover*. Ela pode variar entre zero a um. Quando essa taxa é um significa que todos os pares de indivíduos passarão pelo *crossover* e quando ela é zero significa que nenhum par de indivíduos passará por esse operador. Para qualquer valor da taxa entre zero e um deve ser gerada uma variável com distribuição uniforme para cada par de indivíduos para ser comparada com a taxa de *crossover*. Se a variável é menor ou igual à taxa, ocorre *crossover* com aquele par, se a variável é maior, passa-se para o próximo par de indivíduos sem modificar o par anterior.

± O XWDomR

Mutação é o segundo operador genético mais comum em AG. Esse operador auxilia a encontrar valores (do espaço de busca) que, às vezes, não são conseguidos pelo *crossover*.

A mutação é um operador que deve ser usado com pequena frequência. Com o objetivo de se obter bons resultados a frequência de mutação recomendada em estudos de algoritmos genéticos é de uma mutação a cada mil *bits* [12]. Análogo ao que ocorre no *crossover*, também na mutação existe uma taxa denominada taxa de mutação que também pode variar entre zero e um. Essa taxa comporta-se basicamente da mesma forma que a taxa de *crossover*. A única diferença é que ela é usada indivíduo por indivíduo enquanto a taxa de *crossover* é utilizada a cada par de indivíduos.

A mutação ocorre em um indivíduo quando algum dos seus genes (em uma posição definida por uma variável aleatória uniformemente distribuída) tem o valor mudado ou invertido sem interferência de nenhum outro indivíduo. A Figura 3.4 mostra um exemplo de mutação com um dos indivíduos gerados pelo *crossover* da Figura 3.3, o Indivíduo A'.



*Figura 3.4 – Exemplo da mutação ocorrida no 4<sup>o</sup> gene do indivíduo A' gerando o novo indivíduo A''.*

Pode-se observar na Figura 3.4 que o valor que antes da mutação era um passou para zero. Isso ocorreu porque estão sendo usados nos cromossomos números binários, portanto existem somente dois valores zero e um. Quando são utilizados no cromossomo números reais ou inteiros, letras ou outros tipos de dados, são necessárias outras formas de mudanças.

O objetivo deste sub-item é mostrar a implementação e o funcionamento de um AG. Esse exemplo foi retirado do livro *A Survey of Genetic Algorithms* [14].

O problema é encontrar o valor de  $x$  que minimiza a função  $f(x)$  definida na Equação (3.1).

$$f(x) = -|x \cdot \text{sen}(\sqrt{|x|})| \quad (3.1)$$

O intervalo de interesse de  $x$  na Equação (3.1) é de  $[-512; 512]$ , portanto o seu valor deve estar compreendido entre  $-512$  e  $+512$ . Como a função  $f(x)$  é simétrica usa-se o intervalo de  $[0; 512]$ . O gráfico dessa função é mostrado na Figura 3.5.

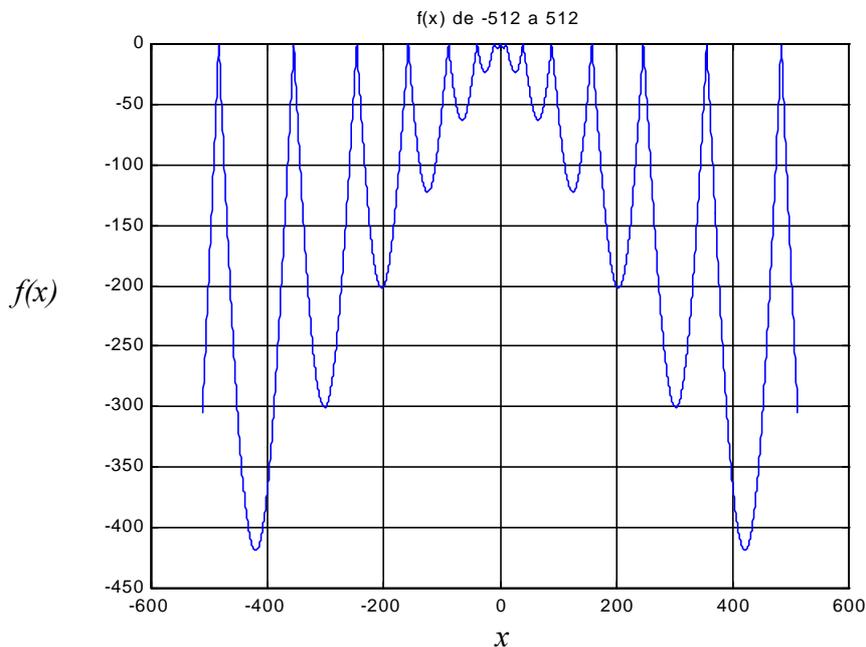


Figura 3.5 – Gráfico da função  $f(x)$  definida na Equação (3.1).

Na função  $f(x)$  definida na Equação (3.1), o vetor usado como cromossomo do indivíduo possui dez campos binários com um intervalo mínimo de 0,5 entre cada ponto. No

programa são assumidos valores de  $[0; 511,5]$  que totalizam 1024 pontos, onde cada ponto pode ser armazenado em um vetor de 10 *bits*.

Os dados de entrada no programa para encontrar o valor  $x$  que minimiza a função  $f(x)$  são:

- $f(x)$  é usada como função de aptidão;
- intervalo de variação, e;
- tamanho do cromossomo.

O primeiro passo do AG é gerar uma população inicial aleatória entre 0 e 512 com 10 indivíduos. Após esse passo é preciso selecionar pela função  $f(x)$  os indivíduos mais aptos dessa população. Isso é feito calculando 3 variáveis pelas Equações (3.2), (3.3) e (3.4).

$$s = \sum_{i=1}^n f_i(x) \quad (3.2)$$

$$p_i = \frac{f_i(x)}{s}, \text{ para } i = 1, 2, 3, \dots, n \quad (3.3)$$

$$c_i = \sum_{k=1}^i p_k, \text{ para } i = 1, 2, 3, \dots, n \quad (3.4)$$

onde:

$s$  – soma dos valores da aptidão de todos os indivíduos;

$n$  – número de indivíduos da população inicial;

$f_i(x)$  – valor da função de aptidão para o indivíduo  $i$ ;

$p_i$  – porcentagem de aptidão de cada indivíduo em relação à aptidão total da população, e;

$c_i$  – soma cumulativa de  $p_i$ .

Os indivíduos são escolhidos por uma variável aleatória  $\gamma$  de distribuição uniforme entre  $[0; 1]$ . Se  $c_{i-1} < \gamma \leq c_i$  é selecionado o indivíduo de índice  $i$ . Essa forma de seleção é a implementação do método da roleta. Esse procedimento é repetido até completar o número de indivíduos da população ( $n$ ). Após essa seleção, os indivíduos são separados em pares. Então, inicia-se o *crossover*. Esse processo caracteriza-se pelos seguintes passos:

1. com os pares escolhidos, é gerada uma variável aleatória uniforme entre  $[0; 1]$  com o número de vezes correspondente ao número de pares existentes. Se essa variável é menor ou igual à taxa de *crossover* que no caso desse exemplo é de 0,6, ocorre o *crossover*;
2. é selecionada, aleatoriamente, uma posição denominada ponto de cruzamento de forma uniforme entre 1 e o comprimento do vetor (cromossomo) menos um, e;

3. são trocados entre os indivíduos os dados ou os genes a partir do ponto de cruzamento mais um até o tamanho do cromossomo. Os dois novos indivíduos gerados entram no lugar dos seus pais.

Após o *crossover*, é aplicada a mutação nos indivíduos da seguinte maneira. Ela é aplicada indivíduo por indivíduo. É gerada para cada vetor uma variável aleatória uniforme entre 0 e 1. Se essa variável é menor ou igual à taxa de mutação que nesse exemplo é de 0,01, é invertido um dos *bits* do vetor (escolhido por uma variável aleatória uniformemente distribuída entre 1 e o comprimento do cromossomo).

Finalmente é produzida a 1<sup>a</sup> geração. A condição de parada do AG é feita pelo número de gerações desejadas como ocorre na maioria dos programas que usa essa técnica.

A Figura 3.6 mostra a população inicial e a 1<sup>a</sup> geração para um total de 50 indivíduos. Nessa figura a função  $f(x)$  é representada pela linha contínua. Os indivíduos da população inicial e os indivíduos dessa geração são representados por círculos e quadrados, respectivamente.

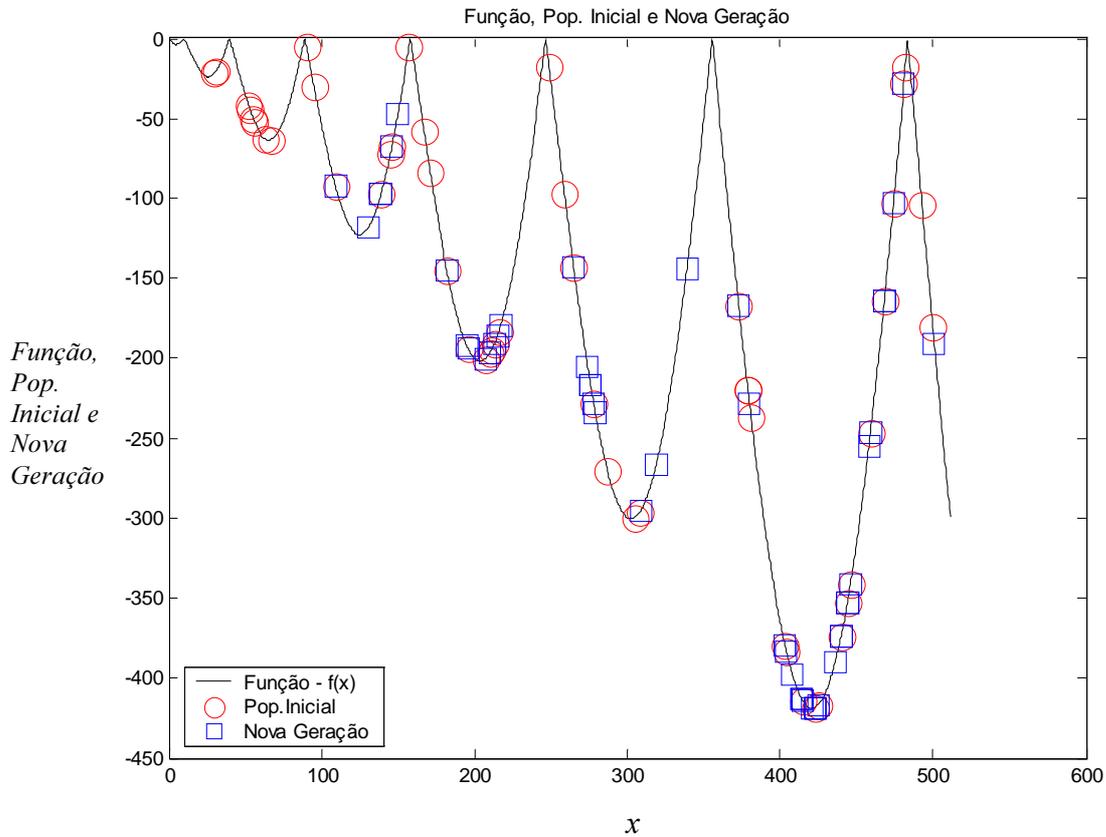


Figura 3.6 – Gráfico da função  $f(x)$ , população inicial e 1ª geração de indivíduos.

Pode-se observar na Figura 3.6, que os indivíduos da 1ª geração estão bem mais concentrados próximos ao valor mínimo da função  $f(x)$ , enquanto os indivíduos da população inicial encontram-se bem mais dispersos e espaçados.

A Figura 3.7 mostra a 4ª e a 5ª gerações da população.

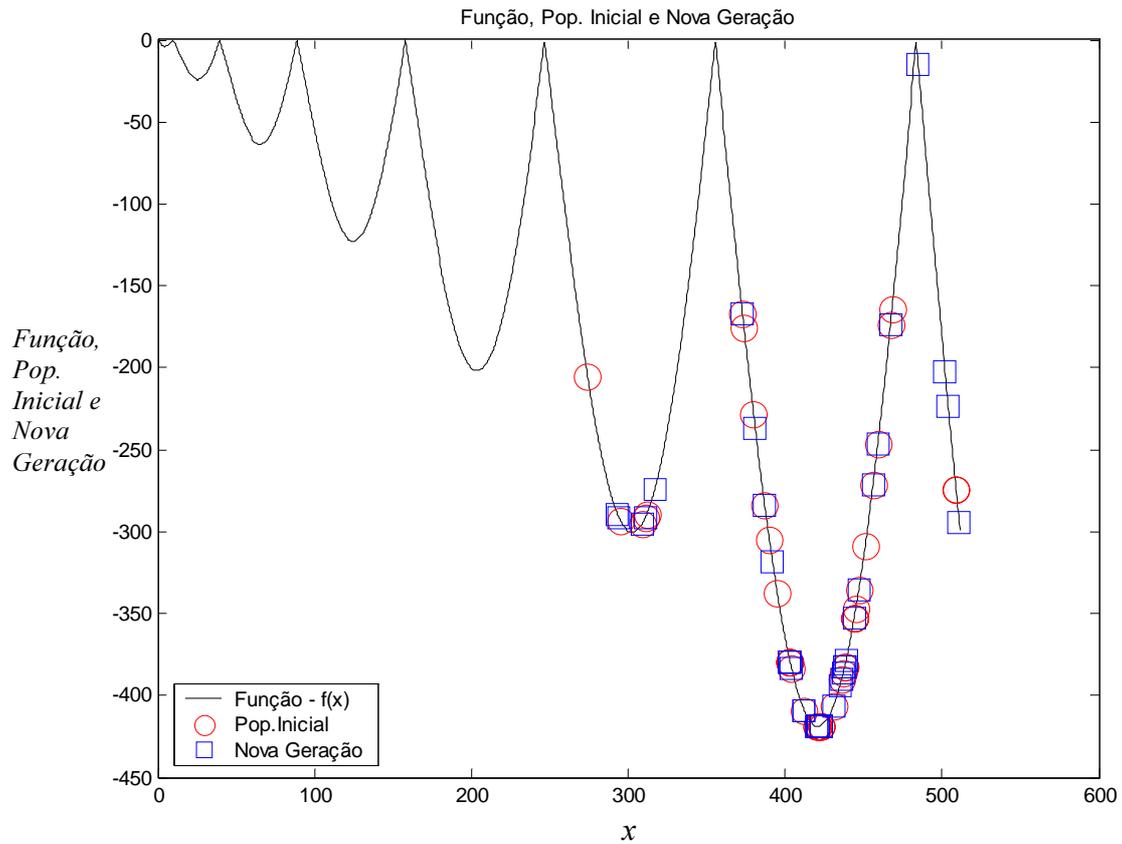


Figura 3.7 – Gráfico da função  $f(x)$ , 4ª e 5ª gerações.

Pode-se observar na Figura 3.7, que à medida que passam as gerações mais indivíduos localizam-se mais próximo do ponto mínimo.

Finalmente a Figura 3.8 mostra a 9ª e a 10ª gerações da população.

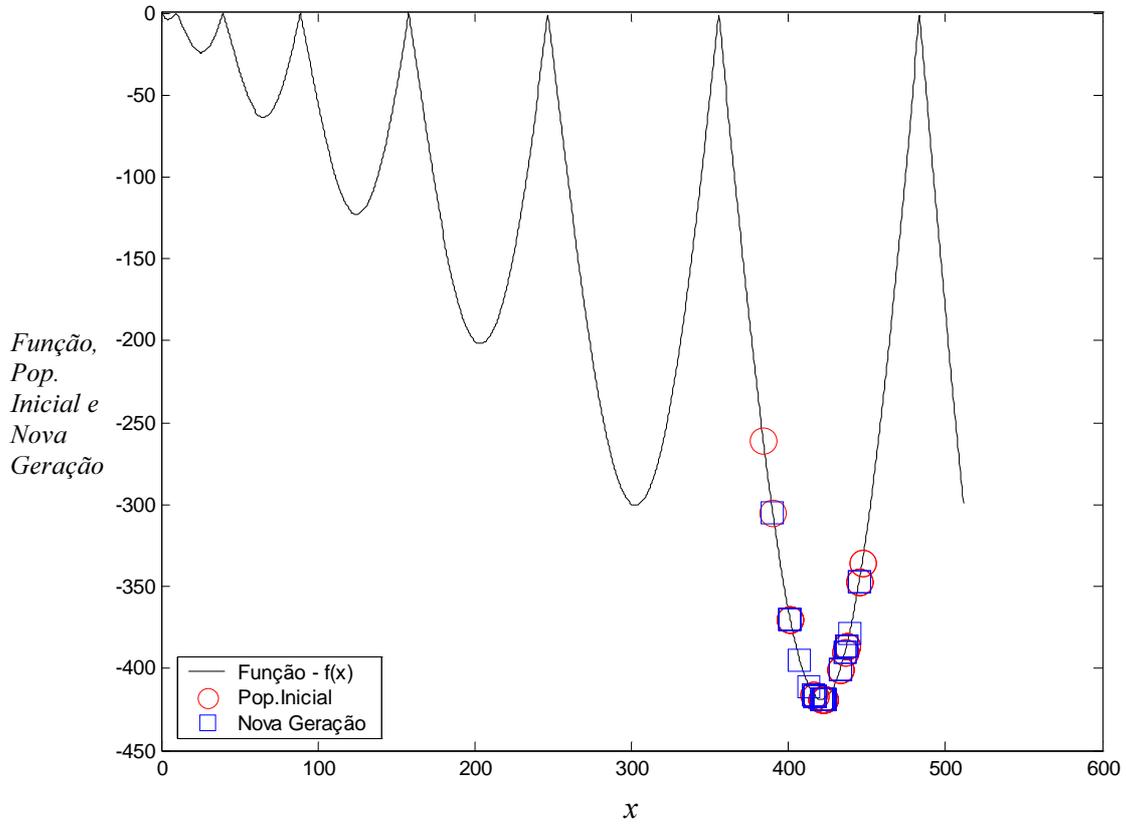


Figura 3.8 – Gráfico da função  $f(x)$ , 9<sup>a</sup> e 10<sup>a</sup> gerações.

Pode-se notar claramente na Figura 3.8, que todos os indivíduos estão localizados bem próximos do ponto de mínimo valor da função  $f(x)$ . Isso mostra que à medida que as gerações vão passando os indivíduos evoluem para os valores desejados.

A Tabela 3.2 mostra os valores relativos à evolução da população mostrados nas Figuras 3.6, 3.7 e 3.8.

Tabela 3.2 –Valores relativos à evolução da população mostrada nas Figuras 3.6, 3.7 e 3.8.

Geração	Média dos valores de $x$	Menor valor de $f(x)$	Valor de $x$ do menor $f(x)$
População Inicial	265,62	- 418,462	423
1ª geração	336,34	- 418,983	421
4ª geração	416,72	- 418,983	421
5ª geração	414,90	- 418,983	421
9ª geração	422,02	- 418,983	421
10ª geração	422,24	- 418,983	421

Pode-se observar na Tabela 3.2, que o menor valor da função  $f(x)$  é - 418,983. Já na 1ª Geração foi encontrado esse valor. Isso significa que pelo menos um dos indivíduos dessa geração já alcançou o valor máximo da função como pode ser verificado na Figura 3.6. Nessa tabela, a evolução dos indivíduos pode também ser observada claramente pelos valores da média dos valores de  $x$ . Na população inicial a média é de 265,02 e na última geração é 422,24. O valor de  $x$  que corresponde ao mínimo da função  $f(x)$  é 421. Portanto, à medida que as gerações vão se alternando a média dos valores de  $x$  vai se aproximando do valor de  $x$  que corresponde ao mínimo da função. Na última geração os valores de  $x$  estão bem mais próximos de 421 do que na população inicial.

± 2 SHUDGRUHV GH 5 HRUGHQDP HQWR HP \$ OJ RULWP RV \* HQpWLFrv

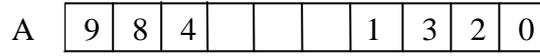
Apesar de serem utilizados em algoritmos genéticos mais parâmetros binários, podem-se usar parâmetros de números inteiros e letras. Além destes parâmetros possuem uma maior possibilidade de modificações, eles precisam de diferentes tipos de *crossover* e de mutação.

± ***Partially matched crossover*** 30 ;

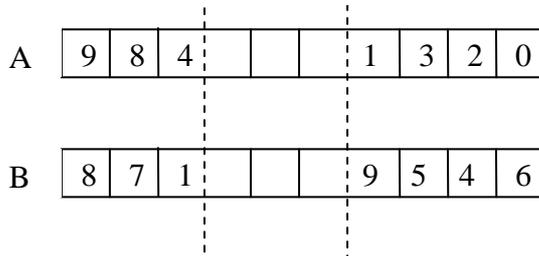
O *partially matched crossover* [12] é um dos principais métodos de reordenamento que pode ser usado em AG tanto para parâmetros de números inteiros como letras.

Supondo que existem dois indivíduos A e B sendo cromossomos de números inteiros de dimensão igual a nove, como mostrado na Figura 3.9. Com uma variável aleatória uniformemente distribuída entre 1 e o comprimento do cromossomo são escolhidos dois pontos de corte de forma que são trocados nos cromossomos os elementos ou os genes entre esses pontos. Entretanto, após essa operação pode acontecer dos cromossomos ficarem com valores repetidos. Devido a esse fato, deve ser estabelecida uma correspondência entre os valores trocados desses cromossomos para que seja realizado um reordenamento ao final dessa operação. Esse reordenamento verifica se existem genes que estão repetidos em A em relação aos novos genes que foram trocados, ou seja, os que estão entre os pontos de corte. Se existirem esses valores eles são trocados pelos valores correspondentes. O mesmo procedimento é feito para o indivíduo B.

Indivíduos



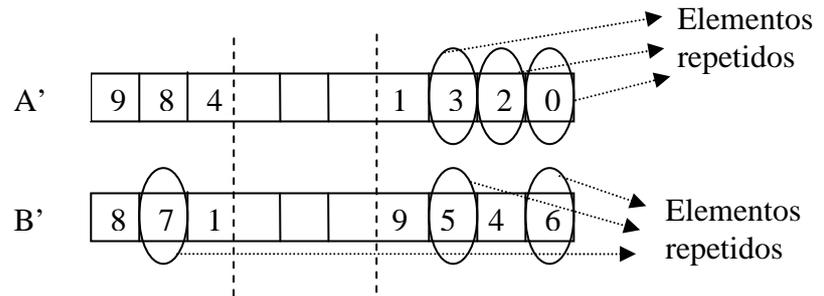
*PMX – com pontos de corte 4 e 6*



*Genes correspondentes para reordenamento*

A	com	B
5	com	2
6	com	3
7	com	0

*PMX – com pontos de corte 4 e 6 e elementos repetidos*



*Indivíduos reordenados após PMX*

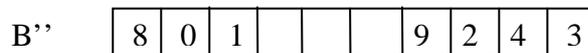
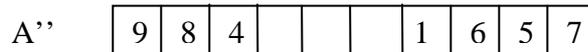


Figura 3.9 – Exemplo de PMX entre os indivíduos A e B de números inteiros.

± , QYHUVmR

A inversão [12] também é um dos principais métodos de reordenamento que é usado em AG para valores com letras e números inteiros.

Na inversão são escolhidas duas posições aleatórias distribuídas uniformemente no cromossomo e os valores dos dois genes são trocados. A Figura 3.10 mostra um exemplo de inversão no cromossomo A''.

A''

9		4	2	3	0	1	6		7
---	--	---	---	---	---	---	---	--	---

Inversão entre as posições 2 e 9

A'''

9		4	2	3	0	1	6		7
---	--	---	---	---	---	---	---	--	---

Figura 3.10 – Exemplo de inversão no cromossomo A''.

± \$ 0J RULWP RV \* HQpWLFrv FRP 9 D0RUHV & RQWtQXRv

O algoritmo genético, também, pode ser implementado usando-se valores contínuos como, por exemplo, números reais. Porém, nesse caso, os operadores genéticos têm de ser diferentes.

Primeiro é definida pela Equação (3.5) qual a escala de valores é permitida para cada elemento do cromossomo.

$$x = (l_{\text{sup}} - l_{\text{inf}}) \cdot \text{rand} + l_{\text{inf}} \quad (3.5)$$

onde:

$x$  – valor de cada elemento do cromossomo;

$l_{sup}$  – limite superior ou valor máximo permitido para  $x$ ;

$l_{inf}$  – limite inferior ou valor mínimo permitido para  $x$ , e;

$rand$  – valor aleatório uniformemente distribuído entre zero e um.

Definidos os valores de  $x$ , considerando-se 2 cromossomos  $p_1$  e  $p_2$  da seguinte maneira:

$$p_1 = [p_{m1}, p_{m2}, \dots, p_{mN}] \quad (3.6)$$

$$p_2 = [p_{d1}, p_{d2}, \dots, p_{dN}] \quad (3.7)$$

onde:

$N$  - máximo de elementos de cada cromossomo.

± 2 SHUDGRUHV GH **crossover** S DUD YD0RUHV FROWIQX RV

Os operadores de *crossover* para valores contínuos são:

#### A) Método de *Radcliff* [15]

O método de *Radcliff* combina os valores dos pais ( $p_m$  e  $p_d$ ) para gerar os filhos ( $p_{nov1}$  e  $p_{nov2}$ ) de acordo com as Equações (3.8) e (3.9).

$$p_{novo1} = \beta \cdot p_{mn} + (1 - \beta) \cdot p_n \quad (3.8)$$

$$p_{novo2} = (1 - \beta) \cdot p_{mn} + \beta \cdot p_{dn} \quad (3.9)$$

onde:

$n$  –  $n$ -ésimo elemento do cromossomo.

$\beta$  - pode ser fixo ou variável e seu valor está entre zero e um.

A desvantagem do método de *Radcliff* é que os valores dos parâmetros contínuos ficam limitados aos máximos valores dos parâmetros existentes na população inicial.

B) Método de *Wright* ou método da extrapolação [15]

O método de *Wright* [15] resolve o problema do método de *Radcliff* pela extrapolação de valores. Nesse método são gerados três descendentes conforme as Equações (3.10), (3.11) e (3.12).

$$p_{novo1} = 0,5 \cdot p_{mn} + 0,5 \cdot p_{dn} \quad (3.10)$$

$$p_{novo2} = 1,5 \cdot p_{mn} - 0,5 \cdot p_{dn} \quad (3.11)$$

$$p_{novo3} = -0,5 \cdot p_{mn} + 1,5 \cdot p_{dn} \quad (3.12)$$

onde:

$n$  –  $n$ -ésimo elemento do cromossomo.

Dos três indivíduos gerados pelo método de Wright exclui-se o que extrapola os valores máximos ou mínimos permitidos para os elementos, ou se nenhum dos valores extrapolar escolhem-se os melhores valores para os elementos.

C) Método da combinação de *crossover* e extrapolação [15]

Considere dois cromossomos  $p_3$  e  $p_4$ , conforme as Equações (3.13) e (3.14).

$$p_3 = [p_{m1}, p_{m2}, \dots, p_{m\alpha}, \dots, p_m] \quad (3.13)$$

$$p_4 = [p_{d1}, p_{d2}, \dots, p_{d\alpha}, \dots, p_{dN}] \quad (3.14)$$

onde:

$\alpha$  - posição do cromossomo gerada aleatoriamente e uniformemente distribuída.

Dois novos valores são gerados conforme as Equações (3.15) e (3.16).

$$p_{\alpha 1} = p_{\alpha} - \beta \cdot (p_{\alpha} - p_{d\alpha}) \quad (3.15)$$

$$p_{\alpha 2} = p_{d\alpha} - \beta \cdot (p_{d\alpha} - p_{\alpha}) \quad (3.16)$$

Os elementos que se encontram nas posições após  $\alpha$  são submetidos ao operador de *crossover* gerando os seguintes descendentes:

Descendente 1:

$$p_3' = [p_{d1}, p_{d2}, \dots, p_{d1}, \dots, p_{dN}] \quad (3.17)$$

Descendente 2:

$$p_4' = [p_{d1}, p_{d2}, \dots, p_{novo2}, \dots, p_{mN}] \quad (3.18)$$

± 2 S HUDGRU GH P XWDomR S DUD YD0RUHV FRQWtQX RV

O novo valor do gene escolhido aleatoriamente é substituído por outro valor aleatório uniformemente distribuído dentro da faixa de valores permitidos, como mostrado na Figura 3.11.

Faixa de valores permitidos:  $-2 \leq x \leq 2$ , onde  $x$  é o valor do gene

Indivíduo antes da mutação

-1,2		0,5	-0,4	0,2	0,9
------	--	-----	------	-----	-----

Indivíduo depois da mutação na posição 2

-1,2		0,5	-0,4	0,2	0,9
------	--	-----	------	-----	-----

Figura 3.11 – Exemplo de mutação para valores contínuos.

± 2 SHUDGRU \* HQpWLFER GH ( OLWLVPR

Um operador muito útil em algoritmos genéticos é o operador de elitismo [12]. Este operador preserva os melhores valores já encontrados pelo programa passando-os para as próximas gerações.

Basicamente, o operador de elitismo trabalha com uma taxa de elitismo que deve ser definida no início do programa. Essa taxa pode variar de zero a um. Quando ela é zero nenhum indivíduo passa pelo elitismo. Quando ela é um, todos os indivíduos passam pelo elitismo. Porém o mais comum é um elitismo de até 0,5, ou seja, no máximo 50% dos melhores indivíduos da população são preservados. Em síntese, o elitismo exclui os melhores indivíduos dos demais operadores genéticos como *crossover* e mutação, portanto ele é utilizado antes da fase de seleção. Quando os demais indivíduos passam por todos os operadores, os indivíduos da chamada “elite” são reincorporados a população e passam para a próxima geração sem nenhuma modificação.

± \$ 0J RULWP R \* X0RVR ***Greedy Algorithm***

O algoritmo genético guloso [12] trabalha com a avaliação contínua da aptidão dos indivíduos obtidos dos operadores de *crossover* e de mutação.

Após cada *crossover* o algoritmo guloso calcula a aptidão daqueles novos indivíduos gerados e compara com a de seus pais. Os dois indivíduos que tiverem os maiores valores de aptidão são colocados na população e os demais são retirados.

O mesmo acontece após cada mutação. A aptidão do novo indivíduo é comparada com a aptidão do indivíduo que o gerou. Se a aptidão do novo indivíduo é melhor do que a aptidão

do indivíduo que o gerou então, o novo indivíduo permanece na população e substitui o indivíduo que o gerou, caso contrário, o indivíduo que o gerou é inserido na população e o novo indivíduo é retirado.

O algoritmo guloso é uma ferramenta importante para melhorar continuamente as soluções, sempre buscando os melhores resultados. Esse algoritmo garante que sempre a melhor solução será escolhida.

A única desvantagem do algoritmo guloso é o tempo de processamento que aumenta consideravelmente. Se o cálculo da aptidão exigir muito processamento isso pode prejudicar o desempenho em termos de tempo no método que está utilizando este algoritmo. Essa desvantagem será mostrada no Capítulo 5 deste trabalho.

± & RQVLGHUO} HV ) LQDLV

Este capítulo descreveu os conceitos básicos de algoritmos genéticos. Foram apresentados os métodos de representação, seleção, reprodução, operadores genéticos e exemplos de aplicações. Todas essas técnicas e métodos apresentados neste capítulo são úteis quando elas são aplicadas às técnicas de análise e síntese de texturas em imagens digitais.

# CAPÍTULO IV

## SÍNTESE DE TEXTURAS

### 4.1 – Introdução

Uma textura pode ser vista como um modelo estocástico, assim como um caso particular de modelo de campos aleatórios de Markov. A síntese desse modelo fornece uma amostra da textura [11].

Além do modelo estocástico, existem outras formas de se obter texturas. Uma delas é pela amostragem digital de formas naturais tais como: uma parede de tijolos, grãos de areia, mármore, tecidos e outros.

O modelo estocástico, geralmente é representado por um vetor de probabilidades. A representação de imagens por esse modelo é uma forma de minimizar alguns tipos de problemas. Nesse modelo uma imagem pode ocupar um pequeno espaço de armazenagem, o que não é possível se for necessário armazená-la toda. O grande problema de imagens digitais é o espaço requerido para armazená-las. Por exemplo, uma imagem com 512 x 512 pixels codificada com 16 *bits* precisa de  $2^{16} \times 512 \times 512 = 16384$  *Megabits* ou 2 *Megabytes* para o seu armazenamento [11]. Outro problema que surge devido ao tamanho da imagem é sua transmissão que pode resultar em falhas na imagem final e em um maior tempo para transmiti-la.

Usando a representação de imagens pelo modelo estocástico, amostras de texturas podem ser geradas de tal forma que a amostra possua a mesma aparência da imagem original. Portanto, várias representações por esse modelo podem ser armazenadas sem ocupar um

grande espaço e elas podem ser transmitidas mais rapidamente para serem reproduzidas ou sintetizadas no receptor.

O sistema proposto e implementado neste trabalho possui basicamente dois algoritmos genéticos. O primeiro algoritmo é responsável por encontrar a melhor estrutura para a amostra original que deve ser reproduzida. Essa estrutura é chamada de *foot pattern*. O segundo algoritmo procura um modelo estocástico que represente o mais próximo possível à textura desejada, ou seja, com o menor erro em relação à amostra original fornecida, usando os melhores resultados obtidos no primeiro algoritmo. A vantagem da utilização da técnica de algoritmos genéticos é devido ao grande espaço de busca tanto do primeiro como do segundo algoritmos.

O primeiro algoritmo trabalha com o *foot pattern*. Com uma população inicial de *foot patterns* e a amostra da imagem original, a função de aptidão desse algoritmo busca o *foot pattern* que tem a menor entropia, ou seja, o menos aleatório considerando a distribuição dos valores binários dos pixels da amostra dessa imagem e seleciona o melhor *foot pattern* para ser utilizado no segundo algoritmo. Com o *foot pattern* selecionado, o segundo algoritmo constrói o histograma de eventos da amostra da textura original. Como neste trabalho foram utilizadas somente imagens binárias, a partir desse ponto, quando estiver escrito valor binário, significa que pode ser o valor zero, que representa a cor preta, ou um, que representa a cor branca.

O segundo algoritmo utiliza uma população de indivíduos (modelos estocásticos de textura) para encontrar o modelo que melhor represente a textura da amostra da imagem original. Para cada modelo da população é sintetizada uma amostra da textura que o modelo representa. Usando essa amostra e o melhor *foot pattern* selecionado pelo primeiro algoritmo, o segundo algoritmo constrói um histograma de eventos para cada modelo. A função de

aptidão desse algoritmo procura o modelo em que a soma do quadrado da diferença entre o histograma de eventos da amostra da imagem original e o histograma de eventos da amostra da imagem sintetizada a partir de um modelo seja o menor possível. O resultado desse algoritmo são os modelos estocásticos de textura que melhor representam a textura da amostra da imagem original.

Este capítulo mostra o método proposto e implementado neste trabalho e suas definições específicas. São apresentados as definições básicas de entropia, histograma de eventos, foot patterns e modelos de Markov parcialmente ordenados (*POMMs*). É descrito o método desenvolvido por Ashlock e Davidson [11]. E é apresentado um modelo de representação e síntese de textura para imagens binárias basru.680Tcb()Tj 5.28 0 Td (n)Tj 6 0 Td Td

### 4.2.1 – Entropia

Considere a saída de uma fonte modelada por uma variável aleatória discreta,  $S$ , que possui um alfabeto finito [16], definido pela Equação (4.1).

$$\vartheta = \{s_0, s_1, \dots, s_{K-1}\} \quad (4.1)$$

A probabilidade de cada símbolo do alfabeto  $\vartheta$  é definida pelas Equações (4.2) e (4.3).

$$P(S = s_k) = p_k \quad k = 0, 1, \dots, K - 1 \quad (4.2)$$

$$\sum_{k=0}^{K-1} p_k = 1 \quad (4.3)$$

Supondo que o símbolo  $s_k$  é emitido pela fonte, se  $p_k = 1$  e  $p_i = 0$  para qualquer  $k \neq i$ , então não existe “surpresa” e, portanto, não existe “informação” quando esse símbolo é emitido. Se  $p_k < p_i$

De acordo com a Equação (4.4) pode-se concluir que:

- Se  $p_k = 1$  então  $I(s_k) = 0$ ;
- $I(s_k) \geq 0$  para  $0 \leq p_k \leq 1$ , e;
- $I(s_k) > I(s_i)$  para  $p_k < p_i$ .

A quantidade de informação,  $I(s_k)$ , produzida por uma fonte durante um intervalo arbitrário de tempo depende do símbolo  $s_k$  emitido por essa fonte a cada momento. Então,  $I(s_k)$  é uma variável aleatória discreta que possui os valores  $I(s_0), I(s_1), \dots, I(s_{K-1})$  com probabilidades  $p_0, p_1, \dots, p_{K-1}$ , respectivamente [16]. Portanto, a entropia dessa fonte ou a média dos valores de  $I(s_k)$  sobre os valores do alfabeto  $\vartheta$  é obtida pela Equação (4.5) ou pela Equação (4.6).

$$H(\vartheta) = \sum_{k=0}^{K-1} p_k \cdot I(s_k) \quad (4.5)$$

ou

$$H(\vartheta) = \sum_{k=0}^{K-1} p_k \cdot \log_2 \left( \frac{1}{p_k} \right) \quad (4.6)$$

A medida de entropia é o conteúdo de informação média por símbolo da fonte [16].

Três considerações importantes devem ser feitas em relação à entropia:

- $0 \leq H(\vartheta) \leq \log_2 K$ , onde  $K$  é o número total de símbolos da fonte;
- se a probabilidade  $p_k = 1$  para algum  $k$  e as demais probabilidades são zero,  $H(\vartheta) = 0$ . Este valor de entropia representa que não existe “incerteza” em relação à fonte, e;
- se  $p_k = \frac{1}{K}$  para todo  $k$ , ou seja, todos os elementos possuem valores iguais de probabilidade, isso corresponde ao máximo de “incerteza” em relação à fonte, então  $H(\vartheta) = \log_2 K$ .

Portanto, um valor alto de entropia significa que a

processamento de imagens como síntese e classificação de texturas, segmentação, restauração e compressão de imagens.

Considere a matriz  $N \times N$  de uma imagem, com  $L$  níveis de cinza, onde  $I(i,j)$  representa o valor de intensidade de nível de cinza do pixel e  $i$  e  $j$  a coordenada do pixel na matriz ( $i$  corresponde à linha e  $j$  à coluna). O valor de  $I(i,j)$  varia de acordo com  $L$ . Por exemplo, se  $L = 32$ , a imagem possui 32 níveis de cinza, portanto  $I(i,j)$  pode variar de 0 a 31. Essa representação pode também ser usada em imagens coloridas. O *MRF* é um campo aleatório cuja probabilidade de massa possui as seguintes propriedades [7]:

- Positividade:  $p(I) > 0$  para todo  $I$ ;
- Markovianidade:  $p(I(i,j)|\text{todo pixel na imagem exceto } (i,j)) = p(I(i,j)|\text{pixels vizinhos de } (i,j))$ , e;
- Homogeneidade:  $p(I(i,j)|\text{pixels vizinhos de } (i,j))$  depende apenas da configuração dos pixels vizinhos e é invariante a translação.

Geralmente para definir o valor do pixel  $I(i,j)$  usa-se os pixels vizinhos mais próximos a ele.

#### 4.2.2.1 – Modelos de Markov parcialmente ordenados (*partially ordered Markov models*)

Modelos de Markov parcialmente ordenados (*POMMs*) é uma subclasse de campos aleatórios de Markov. Os *POMMs* e outros modelos são tentativas de generalizar a propriedade de Markov no tempo para a propriedade de Markov no espaço.

Se  $\{A_i : i = 1, 2, \dots\}$  é um processo aleatório unidimensional, então existem duas formas equivalentes para especificar que o processo é de Markov [10]:

$$P(a_i | a_1, \dots, a_{i-1}) = P(a_i | a_{i-1}) \quad \text{para todo } i \geq 2 \quad (4.7)$$

ou

$$P(a_2, \dots, a_i | a_1) = \prod_{l=2}^i P(a_l | a_{l-1}) \quad \text{para todo } i \geq 2 \quad (4.8)$$

A forma mais popular de generalização para processos de Markov de duas dimensões é a Equação (4.7) que não possui ordenamento ou impõe direcionalidade para as localizações dos pixels [10].

Nos *POMMs* considera-se que os pixels da textura estão parcialmente ordenados estabelecendo-se uma relação de dependência local [11]. Como esse modelo é uma subclasse de *MRF*, o valor do pixel de referência  $I(i,j)$  possui uma relação de dependência local com os valores dos pixels vizinhos. Estes pixels são denominados pixels adjacentes de  $I(i,j)$ . Os pixels vizinhos de  $I(i,j)$  e o próprio pixel  $I(i,j)$ , devem descrever um grafo direto e ser duplicado em todas as regiões da imagem. A união de todos esses subgrafos é uma grafo acíclico direto de uma matriz  $M \times N$  da imagem inteira. Esse grafo tem uma associação com ordem parcial [11].

Considere  $A_{i,j}$  uma variável aleatória relacionada com o valor do pixel  $I(i,j)$  e os pixels vizinhos de  $I(i,j)$  denominados de  $adj_{<} A_{i,j}$  e:

$$P(A_{i,j} = a_{i,j} | \{A_{h,k} = a_{h,k} : A_{h,k} \in \text{adj}_{<} A_{i,j}\}) \quad (4.9)$$

A Equação (4.9) mostra a probabilidade condicional local de  $A_{i,j}$  considerando os pixels vizinhos adjacentes ao pixel  $I(i,j)$ . Devido a essa equação o *POMM* satisfaz a Equação (4.10).

$$P(\{A_{i,j} = a_{i,j}\}) = \prod_{i,j} P(A_{i,j} = a_{i,j} | \{A_{h,k} = a_{h,k} : A_{h,k} \in \text{adj}_{<} A_{i,j}\}) \quad (4.10)$$

A função densidade de probabilidade de massa  $P(\{A_{i,j} = a_{i,j}\})$  é simplesmente o produto das probabilidades condicionais locais [11]. Considerando  $A_{i,j}$  a variável aleatória e  $a_{i,j}$  uma realização dessa variável, pode-se reescrever a Equação (4.10) obtendo-se a Equação (4.11).

$$P(\{A_{i,j} = a_{i,j}\}) = \prod_{i,j} P(a_{i,j} | \text{adj}_{<} a_{i,j}) \quad (4.11)$$

Portanto, um *POMM* é completamente determinado em função das probabilidades condicionais  $P(a_{i,j} | \text{adj}_{<} a_{i,j})$  [11].

### 4.3 – Introdução ao Método Desenvolvido neste Trabalho

As texturas utilizadas neste trabalho possuem dois níveis de cinza, ou seja, preto e branco. Elas são representadas pelos valores binários 0 (zero) para a cor preta e 1 (um) para a cor branca.

#### 4.3.1 - *Foot pattern*

Um *foot pattern*  $(n,k)$  é uma estrutura onde uma série de  $k$  posições de pixels é selecionada de uma janela  $n \times n$ . Por exemplo, um *foot pattern*  $(7, 8)$  é uma janela com  $7 \times 7$  pixels onde 8 de suas posições são selecionadas aleatoriamente, como mostrado na Figura 4.1.

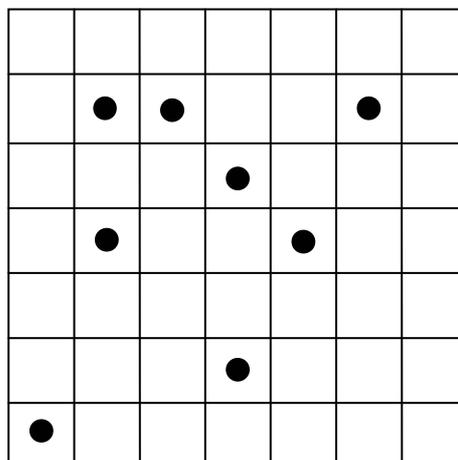


Figura 4.1 –Exemplo de um *foot pattern*  $(7, 8)$ .

Os *foot patterns* são ordenados internamente de tal forma que para um determinado *foot pattern* sempre é possível visitar as posições dos pixels selecionados na mesma ordem [11].

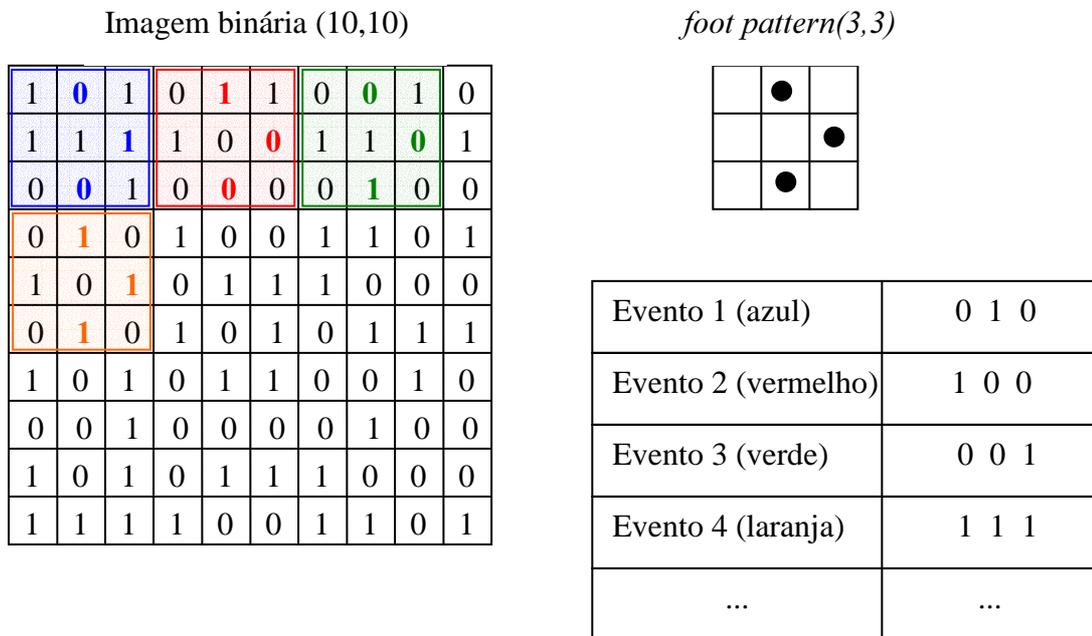
Os *foot patterns* são usados tanto em síntese quanto em classificação de textura. Eles são importantes, pois, dependendo de suas posições dentro da janela, “capturam” certas características da imagem. Para diferentes tipos de imagens ocorrem diferentes *foot patterns*. Portanto, para cada imagem existem alguns *foot patterns* mais adequados para representá-las.

#### 4.3.2 – Histograma de eventos e entropia

Com o objetivo de obter o histograma de eventos  $H_F(T)$  de um *foot pattern*  $(n, k)$   $F$  de uma amostra de textura  $T$ , toda a imagem dessa amostra é varrida utilizando-se um quadrado de dimensão  $n \times n$ . Cada vez que essa imagem é varrida, esse quadrado é colocado em cada posição de forma que nenhum pixel dele fique fora da área da imagem e que ele não fique sobre um mesmo pixel mais de uma vez. As seqüências dos valores dos  $k$  pixels das posições determinadas pelo *foot pattern* são gravadas para cada posição do quadrado na imagem. Cada seqüência de valores é um evento, então com esses eventos é construído o histograma.

Por exemplo, para obter o histograma de eventos de uma imagem binária  $10 \times 10$  utilizando um quadrado  $3 \times 3$ . Primeiro, a extremidade esquerda superior desse quadrado é posicionada na 1ª linha e na 1ª coluna dessa imagem e é gravada, em um vetor, a seqüência dos  $k$  valores binários que ocorrem nas respectivas posições do *foot pattern*. Esse é o primeiro evento. A seguir esse quadrado continua na mesma linha e na 4ª coluna dessa imagem e é gravada, em outro vetor, a seqüência dos  $k$  valores binários que ocorrem nas respectivas posições do *foot pattern* nessa nova posição. Esse é o segundo evento. Da mesma maneira é gravado o terceiro evento. No quarto evento, a extremidade esquerda superior do quadrado teria de ficar na 1ª linha e na 10ª coluna da imagem. Porém, duas colunas do quadrado ficariam fora da área da imagem, e isso não pode ocorrer. Nesse caso, o quarto evento será na

4ª linha e 1ª coluna dessa imagem. Esse processo se repete até que toda a imagem seja varrida e sejam gravados todos os eventos. Ao terminar de varrer toda a imagem constrói-se com os vetores de cada evento o histograma de eventos  $H_F(T)$  dessa imagem. Esse histograma é normalizado, ou seja, cada valor de um determinado evento é dividido pelo total de eventos e é atribuído a cada evento  $e_i$  um valor de probabilidade  $P(e_i)$  conforme a quantidade de cada evento no histograma. A Figura 4.2 ilustra esse exemplo.



*Figura 4.2 – Construção do histograma de eventos.*

Com o valor de  $P(e_i)$  é possível calcular o valor da entropia de um *foot pattern*  $F$  pela Equação (4.12).

$$\varepsilon_F(T) = -\sum_{i=1}^{2^k} P(e_i) \cdot \log_2(e_i) \quad (4.12)$$

Em termos de probabilidade pode-se dizer que na Equação (4.12) quando existe uma distribuição uniforme de eventos, o valor da entropia é máximo, ou seja, a “incerteza” é maior em relação àquele *foot pattern*. Porém, quando a entropia alcança o menor valor significa que um dos eventos possui probabilidade um e os demais zero e conseqüentemente a “incerteza” é pequena em relação ao *foot pattern* em questão. O valor da entropia é utilizado no primeiro algoritmo genético deste trabalho como valor de aptidão.

#### 4.3.3 – A estrutura do *POMM*

Nesta dissertação a estrutura do *POMM* possui cinco pixels denominados de:

- $x_R$  é o pixel de referência cujo valor binário depende dos valores binários dos pixels adjacentes a ele, e;
- $x_0$ ,  $x_1$ ,  $x_2$  e  $x_3$  são os pixels adjacentes ao pixel  $x_R$ .

A Figura 4.3 mostra a estrutura do *POMM* utilizada neste trabalho [11].

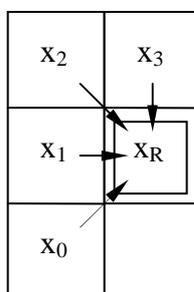


Figura 4.3 – Estrutura do *POMM* utilizada neste trabalho.

A Figura 4.3 mostra um grafo direto com cinco pixels, ou seja,  $x_R$  e seus quatro pixels adjacentes. Considere que em uma imagem essa estrutura é duplicada em todas as localizações possíveis. A união desses sub-grafos diretos é um grafo acíclico direto de  $M \times N$  pixels de toda imagem [11]. Esse grafo está associado com a ordem parcial [19] que pode ser usada para definir o *POMM* como mostrado na sub-seção 4.2.2.1 deste capítulo.

No modelo *POMM* deste trabalho, o valor binário do pixel  $x_R$  depende diretamente dos valores binários dos pixels  $x_0$ ,  $x_1$ ,  $x_2$  e  $x_3$ . Existem várias expressões paramétricas para os *POMMs*, porém os modelos utilizados para representar as texturas utilizadas nesta dissertação são *POMMs* não paramétricos, ou seja, sem a utilização de parâmetros [11], de forma a facilitar a implementação e diminuir o tempo de processamento do algoritmo.

#### 4.3.4 – O vetor de probabilidades dos modelos de textura utilizados neste trabalho

A forma utilizada para representar cada textura nesta dissertação é um vetor de probabilidades com 16 posições, ou seja,  $2^4$ . Cada posição desse vetor refere-se à probabilidade de ocorrência do valor binário de um determinado pixel em relação a seus pixels vizinhos conforme a estrutura do *POMM* utilizado. A Figura 4.4 mostra esse vetor.

$P(e_0)$	$P(e_1)$	$P(e_2)$	$P(e_3)$	$P(e_4)$	$P(e_5)$	$P(e_6)$	$P(e_7)$	$P(e_8)$	$P(e_9)$	$P(e_{10})$	$P(e_{11})$	$P(e_{12})$	$P(e_{13})$	$P(e_{14})$	$P(e_{15})$
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-------------	-------------	-------------	-------------	-------------	-------------

Figura 4.4 – Estrutura do vetor do modelo estocástico de textura utilizado neste trabalho.

Considere  $x_0$ ,  $x_1$ ,  $x_2$  e  $x_3$  os pixels adjacentes ao pixel  $x_R$ . Se os valores binários dos pixels  $x_0$ ,  $x_1$ ,  $x_2$  e  $x_3$  são todos iguais a zero, a probabilidade  $P(e_0)$  é a probabilidade do valor

binário do pixel  $x_R$  ser igual a um. Se os valores binários dos pixels  $x_0$ ,  $x_1$  e  $x_2$  são iguais a zero e o valor binário do pixel  $x_3$  é igual a um, então o valor de  $P(e_i)$  é a probabilidade do valor binário do pixel  $x_R$  ser igual a um e assim por diante até a última posição em que os valores binários correspondentes aos pixels  $x_0$ ,  $x_1$ ,  $x_2$  e  $x_3$  são iguais a um, completando assim o vetor de 16 posições.

#### 4.3.5 – Gerando amostras a partir dos vetores dos modelos estocásticos de textura

O vetor de probabilidades de 16 posições explicado na sub-seção 4.3.4 deste capítulo é utilizado para representar e gerar as amostras das texturas.

Para gerar as amostras de texturas a partir dos vetores de modelos estocásticos, primeiro, cria-se uma matriz 200 x 200. Devido à estrutura do *POMM*, a primeira linha, a primeira coluna e a última linha dessa matriz são preenchidas com valores zeros e uns conforme uma distribuição uniforme. O primeiro valor gerado é na posição (2,2), pois a primeira linha e a primeira coluna já estão preenchidas. Portanto, o valor binário do pixel dessa posição depende dos valores binários dos pixels que estão nas posições (3,1), (2,1), (1,1) e (1,2) nessa ordem. Conforme os valores binários desses pixels é escolhido um dos valores do vetor de 16 posições que corresponde ao valor binário resultante da união desses valores binários. Então, é gerado um número aleatório com uma distribuição uniforme  $\alpha$ . Se  $\alpha < P(e_i)$  o valor binário do pixel que está na posição (2,2) recebe o valor igual a um, caso contrário, esse pixel recebe o valor igual a zero. Cada posição seguinte do pixel da matriz é analisada e o valor gerado é colocado na posição correspondente. A ordem que os pixels são varridos é coluna por coluna.

#### **4.4 – Método de Síntese de Textura pelo Algoritmo Genético Tandem que utiliza Modelos de Markov Parcialmente Ordenados Não Paramétricos**

O método de síntese de textura pelo algoritmo genético Tandem que utiliza modelos de Markov parcialmente ordenados não paramétricos foi desenvolvido por Ashlock e Davidson [11]. Esse método é composto de dois algoritmos genéticos. O primeiro algoritmo genético denominado de algoritmo genético da mínima entropia (AGME) procura o melhor *foot pattern* para a amostra de textura fornecida. O segundo algoritmo genético denominado algoritmo genético do modelo de textura (AGMT) procura o modelo estocástico de textura que seja capaz de produzir uma amostra com aparência bem próxima a amostra fornecida.

##### **4.4.1 – Algoritmo genético da mínima entropia**

No algoritmo genético da mínima entropia (AGME) é fornecido ao programa uma amostra de textura com 200 x 200 pixels [11] e é utilizado nesse algoritmo um *foot pattern* (7, 10). Cada *foot pattern* de dimensão 1 x 10 é um indivíduo da população e é utilizada uma janela 7 x 7 para varrer a imagem. A partir dessa amostra de textura, o algoritmo gera uma população de *foot patterns* utilizando uma variável aleatória com distribuição uniforme. Nesse caso, cada posição do *foot pattern* dentro da janela é representada por uma coordenada, onde x é a abscissa e y é a ordenada. Essa coordenada representa a distância dos pixels ao centro da janela que é o ponto (0,0). A Figura 4.5 mostra um exemplo de *foot pattern* representado de três formas:

- pelos valores das distâncias em relação ao centro da janela;

- pela forma espacial dos pixels dentro da janela, e;
- pelo vetor que é usado como indivíduo da população.

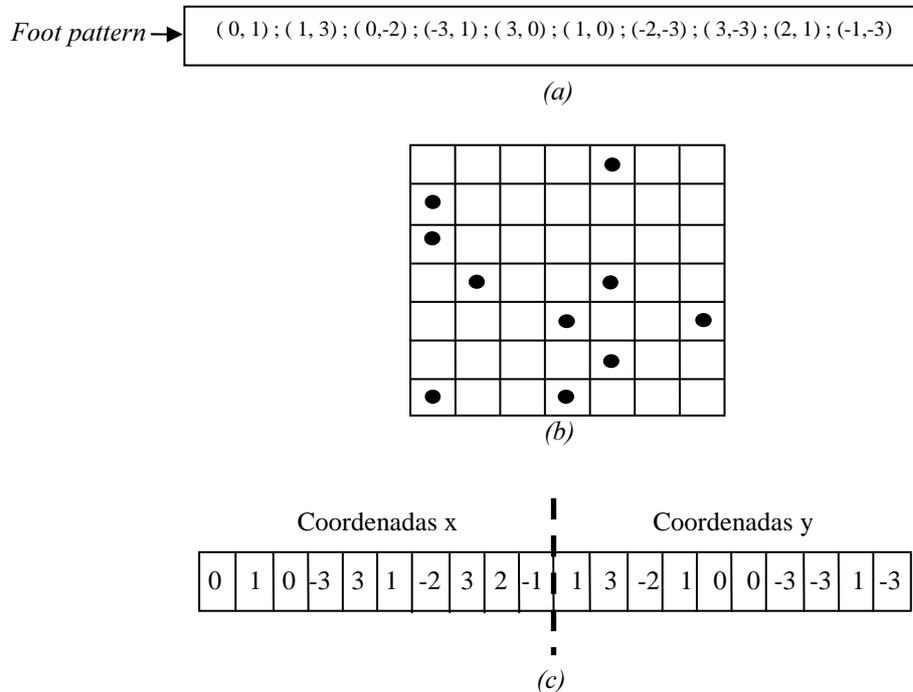


Figura 4.5 – (a) Representação de um foot pattern pelos valores das distâncias em relação ao centro da janela; (b) representação de um foot pattern na forma espacial dos pixels dentro da janela; (c) representação de um foot pattern pelo vetor que é usado no algoritmo como indivíduo da população.

Inicialmente o algoritmo AGME utiliza os *foot patterns* ou indivíduos da população inicial e suas respectivas janelas 7 x 7, e constrói vários histogramas de eventos, um para cada indivíduo. Cada valor desse histograma é normalizado dividindo-o pelo número total de eventos de cada histograma. Esse valor normalizado do histograma é a probabilidade de cada evento (seqüência de zeros e uns, porque a imagem utilizada neste trabalho é binária) para aquele indivíduo. Com o valor da probabilidade de cada *foot pattern* calcula-se o valor da

entropia de cada indivíduo. Este é o valor que é utilizado como resposta da função de aptidão de cada indivíduo. Quanto menor a entropia, menor a “incerteza” em relação ao indivíduo ou *foot pattern* e maior a aptidão desse indivíduo. Em contrapartida o indivíduo menos apto é o que possui maior valor de entropia. Então, o AGME separa os indivíduos em grupos de quatro indivíduos [11].

O primeiro operador utilizado no AGME é o operador de *crossover*. Os dois indivíduos mais aptos de cada grupo passam pelo *crossover*. Neste operador são analisados os genes dos dois indivíduos que estão passando por ele. Se os valores dos genes são iguais nos dois indivíduos ou cromossomos eles são mantidos nos descendentes. Os valores que são exclusivos apenas de um cromossomo são distribuídos uniformemente entre os dois indivíduos descendentes. Os dois novos indivíduos substituem os dois indivíduos com menor aptidão do grupo.

O segundo operador utilizado no AGME é o operador de mutação. Neste operador ocorre uma mudança em uma das posições do indivíduo escolhido para a mutação. O valor da posição a ser alterado é substituído por um valor da janela que não está ocupada por nenhum elemento do cromossomo. Os novos descendentes substituem os de menor aptidão no grupo.

Com uma população de *foot patterns* modificada inicia-se o AGME novamente, repetindo o processo até que a geração desejada seja obtida. O melhor indivíduo ou *foot pattern* da última geração é utilizado pelo segundo algoritmo genético do modelo estocástico de textura (AGMT) [11]. A Figura 4.6 mostra o diagrama de blocos do AGME.

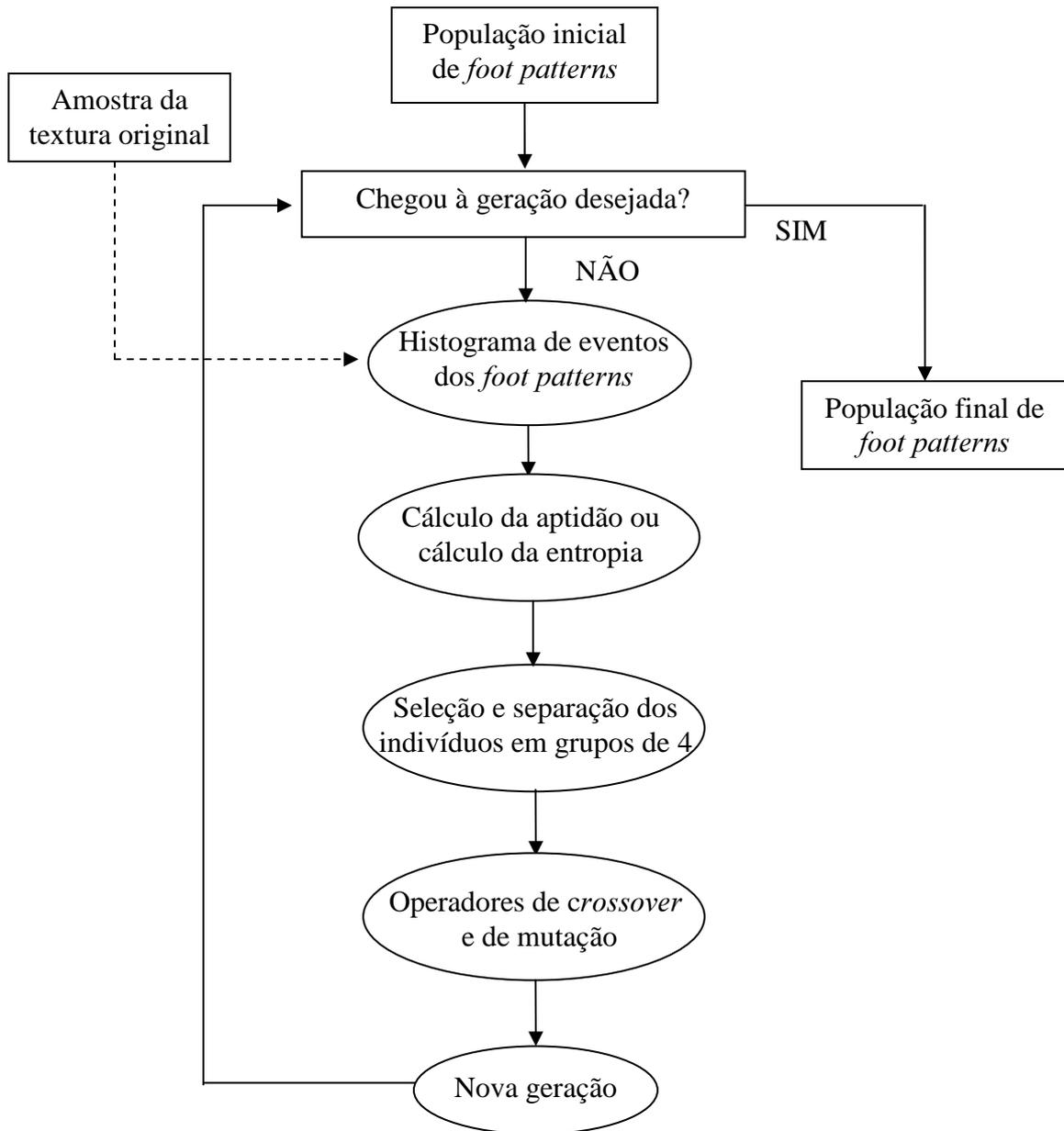


Figura 4.6 – Diagrama de blocos do AGME.

#### 4.4.2 – Algoritmo genético dos modelos de textura

O objetivo do algoritmo genético dos modelos de textura (AGMT) [11] é encontrar um modelo de textura que sintetize uma amostra o mais parecido possível com a amostra da textura original.

Os valores de entrada do AGMT são o melhor indivíduo da última geração do algoritmo AGME e a amostra da textura original.

No AGMT, primeiro é gerada uma população inicial de modelos de textura. Como foi visto na sub-seção 4.3.4 deste capítulo, cada modelo de textura é representado por um vetor de dimensão 16. Então, vários vetores são gerados conforme uma variável aleatória com distribuição uniforme com valores de probabilidade no intervalo de zero a um para cada campo de cada vetor. Cada vetor ou modelo é um indivíduo da população. Com esses vetores são construídas as amostras 200 x 200 pixels de cada modelo. Para cada amostra é construído um histograma de eventos normalizado utilizando o melhor indivíduo ou *foot pattern* obtido no AGME.

O valor da aptidão é calculado pelo quadrado da soma entre a diferença do histograma de eventos da amostra da textura original e do histograma de eventos da amostra da textura de cada modelo ou indivíduo da população. O modelo que possui o menor valor da diferença em relação à textura original é o mais apto dessa geração. Em contrapartida, o que possui o maior valor dessa diferença é o menos apto.

Após o cálculo da aptidão, os indivíduos são separados, novamente, em grupos de quatro. Os dois melhores ou mais aptos de cada grupo passam pelo operador de *crossover* enquanto os outros dois são substituídos pelos descendentes resultantes da reprodução dos dois indivíduos mais aptos.

O primeiro operador utilizado no AGMT é o operador de *crossover*. Este operador escolhe aleatoriamente um ponto com distribuição uniforme entre 1 e 15 para ser o ponto de corte para cada par de cromossomos. Esse operador funciona de forma convencional, ou seja, os valores depois do ponto de corte são trocados entre os cromossomos. Os dois novos indivíduos substituem os dois indivíduos de menor aptidão no grupo.

O segundo operador utilizado no AGMT é o operador de mutação. Cada modelo de textura é submetido de zero a três pontos de mutação conforme uma variável uniformemente distribuída entre 0 e 3 [11]. O ponto escolhido ou os pontos escolhidos onde ocorrerão modificações são aleatoriamente e uniformemente distribuídos entre 1 e 16. A cada ponto escolhido é adicionado um valor uniformemente distribuído entre - 0,2 e 0,2 [11]. Como o número modificado é uma probabilidade ele deve estar sempre entre zero e um. Então, se o número é maior do que um ou menor do que zero as Equações (4.13) e (4.14) devem ser utilizadas para modificá-lo.

$$x = -x \quad \text{para } x < 0 \quad (4.13)$$

$$x = 2 - x \quad \text{para } x > 1 \quad (4.14)$$

Os novos indivíduos gerados por grupo substituem os indivíduos com menor aptidão de cada grupo. Com uma população modificada, o AGMT retorna ao ponto de geração de amostras dos modelos representados pelos vetores e continua nesse *loop* até que a geração desejada seja obtida. A Figura 4.7 mostra o diagrama de blocos do AGMT.

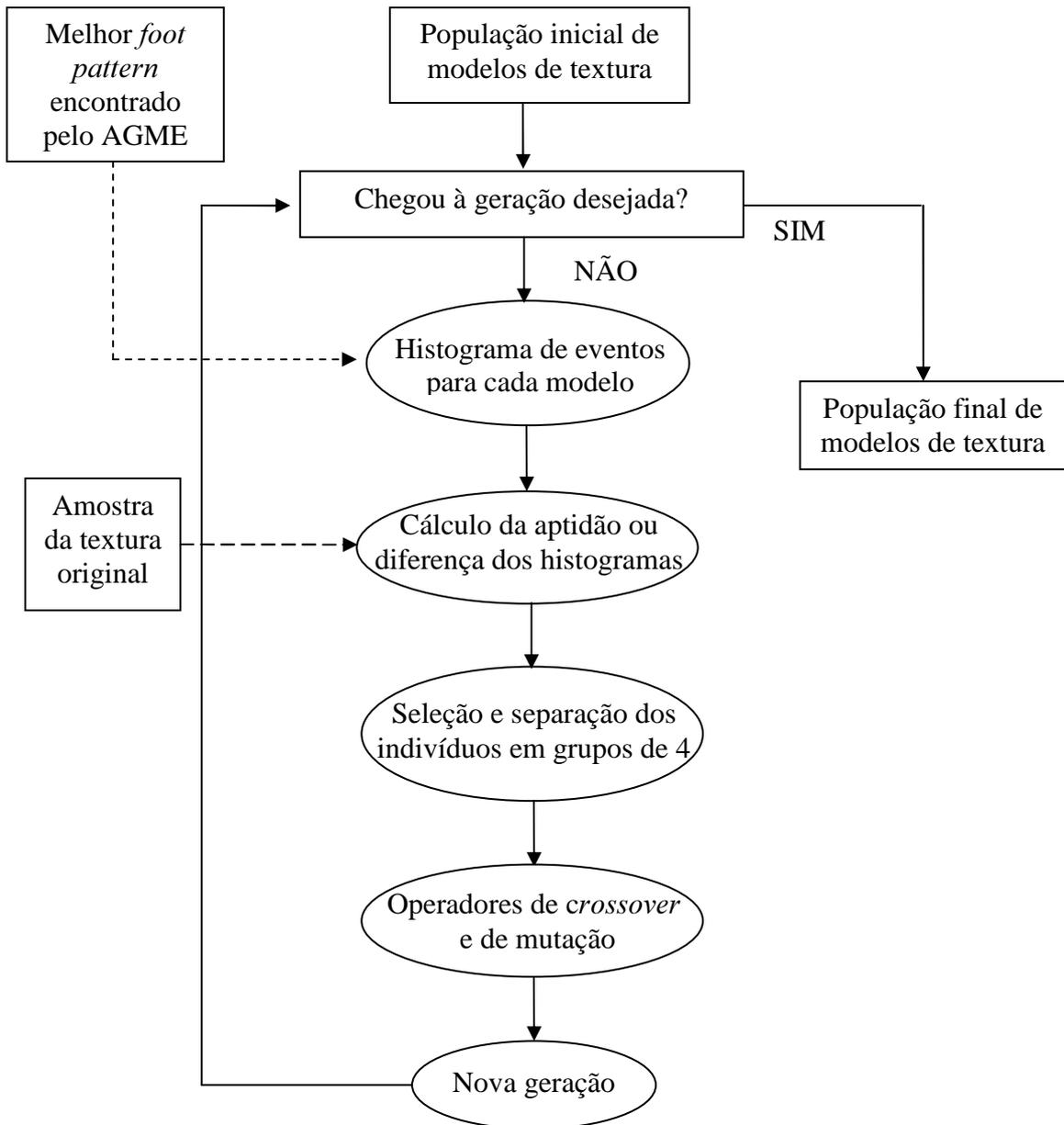


Figura 4.7 – Diagrama de blocos do AGMT.

#### 4.5 – Método de Síntese de Textura pelo Algoritmo Genético Tandem que utiliza Modelos de Markov Parcialmente Ordenados Não Paramétricos Modificado

O método proposto e implementado neste trabalho realiza algumas alterações nos algoritmos AGME e AGMT mostrados na seção 4.4 deste capítulo. Este método também

possui dois algoritmos: o algoritmo genético da mínima entropia modificado (AGMEM) e o algoritmo genético dos modelos de textura modificado (AGMTM). O objetivo desse método é melhorar o desempenho, o tempo de processamento necessário para atingir os resultados desejados e obter, nas primeiras gerações, uma melhor evolução da população.

#### 4.5.1 – Algoritmo genético da mínima entropia modificado

A primeira alteração que o algoritmo genético da mínima entropia modificado (AGMEM) realiza no AGME é na forma de representação dos indivíduos ou dos *foot patterns*. O AGMEM representa cada posição do *foot pattern* dentro da janela por apenas um valor. Para a janela 7 x 7 esse valor varia de 1 a 49, percorrendo a janela linha por linha. Quando a janela é percorrida, os valores que estão na primeira linha variam de 1 a 7, na segunda linha de 8 a 14 e assim por diante até a última linha onde os valores variam de 43 a 49. Essa forma de representação resulta em duas vantagens: a primeira é que esse mesmo vetor é usado como indivíduo na reprodução diminuindo o tempo de processamento e a segunda vantagem é que essa forma de representar possui a metade da dimensão dos vetores utilizados na reprodução do AGME. A Figura 4.8 mostra um exemplo de *foot pattern* do AGMEM representado de 2 formas:

- pelo vetor que contém os valores das posições na janela, e;
- pela forma espacial dos pixels dentro da janela.

5	8	15	23	26	32	35	40	43	46
---	---	----	----	----	----	----	----	----	----

(a)

				●		
●						
●						
	●			●		
			●			●
				●		
●			●			

(b)

Figura 4.8 –(a) Representação de um foot pattern por um vetor de cada indivíduo;

(b) representação espacial de um foot pattern.

A aptidão dos indivíduos no AGMEM é calculada da mesma forma explicada no AGME na seção 4.4 deste capítulo. Logo após esse passo, foram introduzidas mais algumas modificações. Primeiro, os indivíduos são dispostos em ordem decrescente de aptidão, ou seja, do mais apto ao menos apto. Foi introduzido um operador de elitismo com a finalidade de manter os indivíduos mais aptos para as próximas gerações. Os indivíduos da elite são armazenados e são realizadas cópias deles. As cópias vão para a reprodução e os indivíduos armazenados da elite são introduzidos após passar pelos operadores de *crossover* e de mutação para a obtenção da população final de cada geração.

Conforme a taxa de elitismo utilizada, os indivíduos são selecionados para passar pelos operadores de *crossover* e de mutação. Por exemplo, se o tamanho da população total é 20 e a taxa de elitismo é 0,2 então os  $20 - (20 * 0,2) = 16$  indivíduos mais aptos passam pelos operadores de *crossover* e de mutação. Dessa vez, os indivíduos não são separados em grupos.

Os pares passam pelos operadores na ordem que se encontram e substituem seus pais na população final.

O segundo operador utilizado no AGMEM é o operador de *crossover*. Os dois indivíduos que estão passando por esse operador têm seus genes analisados. Os valores dos genes que são iguais nos dois indivíduos ou cromossomos são mantidos nos descendentes. Os valores que são exclusivos apenas de um cromossomo são distribuídos uniformemente entre os dois indivíduos descendentes. Os novos indivíduos substituem os seus pais na população.

O terceiro operador utilizado no AGMEM é o operador de mutação. Para a mutação ocorre uma mudança em uma das posições do indivíduo escolhido para a mutação. Essa posição é substituída pela posição da janela (que para a janela 7 x 7 utilizada neste trabalho varia de 1 a 49) que não está ocupada naquele momento por nenhum elemento do cromossomo. Os novos descendentes substituem os seus pais na população.

Depois da mutação, os novos indivíduos passam pelo algoritmo guloso. Esse algoritmo calcula as aptidões desses indivíduos. Se a aptidão de cada novo indivíduo é maior do que de seu ascendente direto (antes do operador de *crossover*), esses novos indivíduos permanecem na população. Se a aptidão de cada novo indivíduo é menor do que a aptidão de seu ascendente direto (antes do operador de *crossover*), os novos indivíduos são substituídos na população por seus ascendentes diretos.

Com uma população de *foot patterns* modificada o AGMEM é reiniciado a partir do cálculo do histograma de eventos. Esse processo é repetido até que a geração desejada seja obtida. O AGMEM utiliza o melhor indivíduo ou *foot pattern* da última geração do AGMEM. A Figura 4.9 mostra o diagrama de blocos do AGMEM.

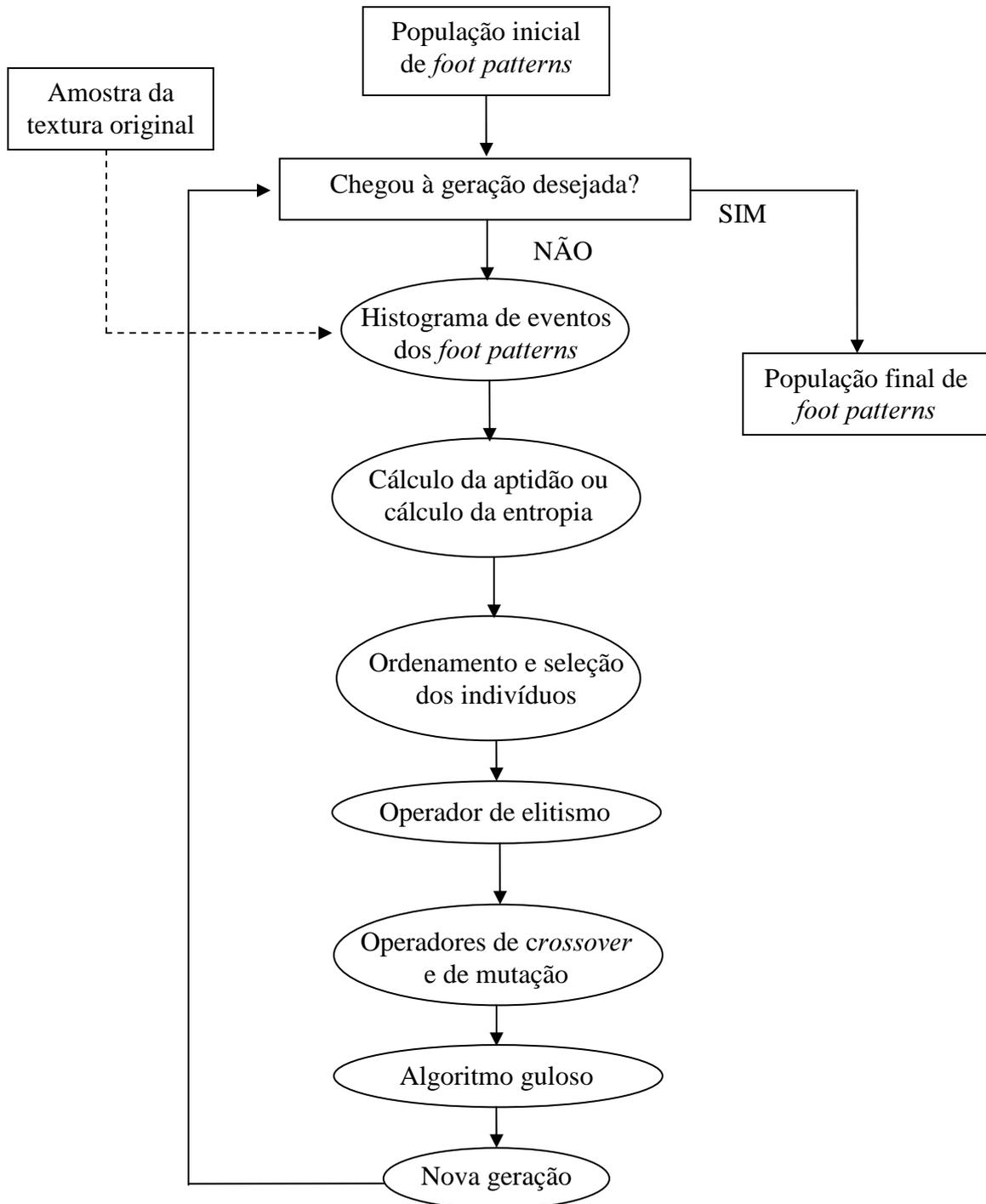


Figura 4.9 – Diagrama de blocos do AGMEM.

#### 4.5.2 – Algoritmo genético dos modelos de textura modificado

Os passos do algoritmo genético dos modelos de textura modificado (AGMTM) até o cálculo da aptidão são idênticos aos passos do AGMT mostrado na sub-seção 4.4.2 deste capítulo. As modificações no AGMTM são logo depois desse cálculo. Os indivíduos são ordenados de forma decrescente, ou seja, dos mais aptos aos menos aptos.

O operador de elitismo foi introduzido no AGMTM e trabalha como mostrado na sub-seção 4.5.1 deste capítulo e da mesma maneira na seleção dos indivíduos para passarem pelos operadores. Nesse algoritmo os indivíduos não são separados em grupos.

O segundo operador utilizado no AGMTM é o operador de *crossover*. Neste operador, escolhe-se, aleatoriamente, para cada par de indivíduos, um ponto com distribuição uniforme entre 1 e 15 para ser o ponto de corte. Logo após, os valores depois do ponto de corte são trocados entre os cromossomos e os novos indivíduos substituem os seus ascendentes diretos na população.

O terceiro operador utilizado no AGMTM é o operador de mutação. Cada modelo de textura é submetido de zero a três pontos de mutação, conforme uma variável uniformemente distribuída entre 0 e 3 [11]. O ponto escolhido ou os pontos escolhidos são aleatoriamente e uniformemente distribuídos entre 1 e 16. A cada ponto escolhido é adicionado um valor uniformemente distribuído entre - 0,2 e 0,2 [11]. O número modificado é uma probabilidade e deve estar sempre entre zero e um. Então se esse número é maior do que um ou menor do que zero as Equações (4.13) e (4.14) devem ser utilizadas para modificar esse valor. Os novos indivíduos gerados substituem os seus pais na população.

Após os novos indivíduos gerados substituírem os seus pais na população, o AGMTM utiliza o algoritmo guloso (*greedy algorithm*) para analisar as modificações na população. Se

cada novo indivíduo gerado está mais apto do que seu ascendente direto (antes de passar pelo operador de *crossover*) ele permanece nessa população, caso contrário, ele é substituído pelo seu ascendente. Essa técnica garante que sempre os indivíduos mais aptos são selecionados. Com a população modificada o AGMTM retorna ao ponto da geração das amostras dos modelos de texturas representados pelos vetores e continua nesse *loop* até que a geração desejada seja obtida. A Figura 4.10 mostra o diagrama de blocos do AGMTM.

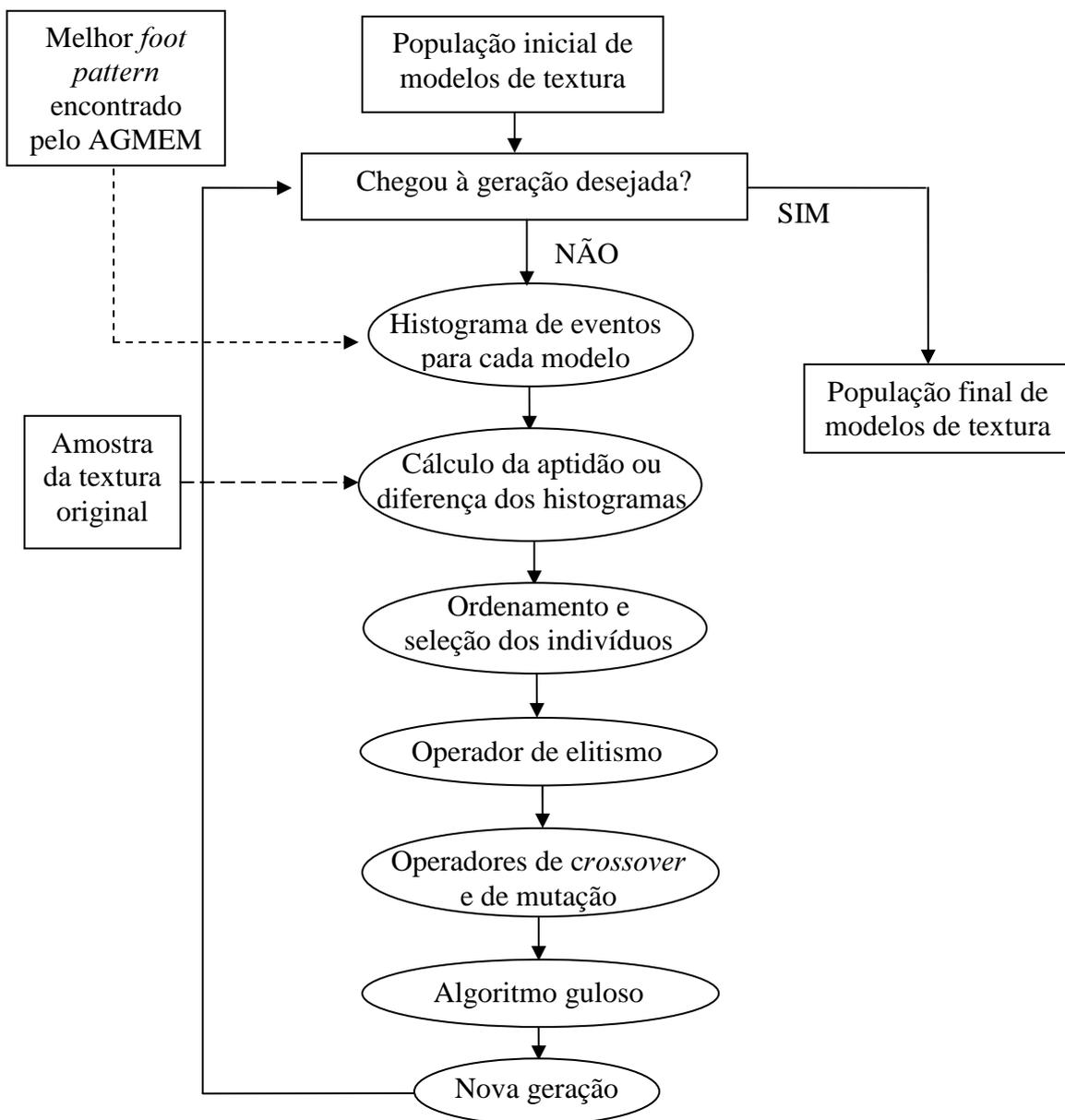


Figura 4.10 – Diagrama de blocos do AGMTM.

#### 4.6 – Considerações Finais deste Capítulo

Este capítulo mostrou o método proposto e implementado neste trabalho e suas definições específicas. Foram apresentadas as definições básicas de entropia, histograma de eventos, *foot patterns* e modelos de Markov parcialmente ordenados (*POMMs*). Foi descrito o método desenvolvido por Ashlock e Davidson [11] e suas definições. E foi apresentado um modelo de representação e síntese de textura para imagens binárias, baseado no método de Ashlock e Davidson, usando apenas uma amostra da textura original.

São desenvolvidas neste trabalho modificações no método de síntese de textura pelo algoritmo genético Tandem que utiliza modelos de Markov parcialmente ordenados não paramétricos, com o objetivo de melhorar o desempenho, o tempo de processamento e a evolução da população nas primeiras gerações. Os resultados dessas modificações são apresentados no capítulo 5 deste trabalho.

# CAPÍTULO V

## RESULTADOS OBTIDOS

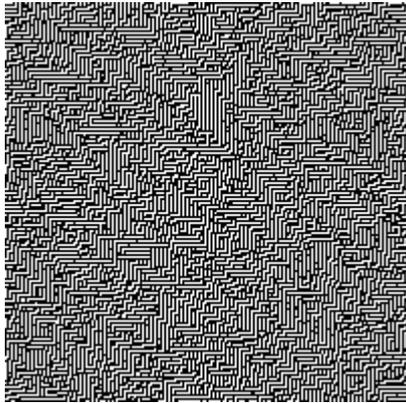
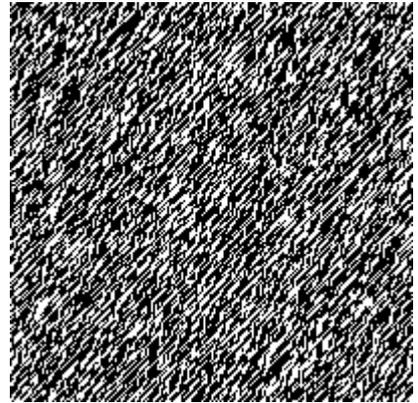
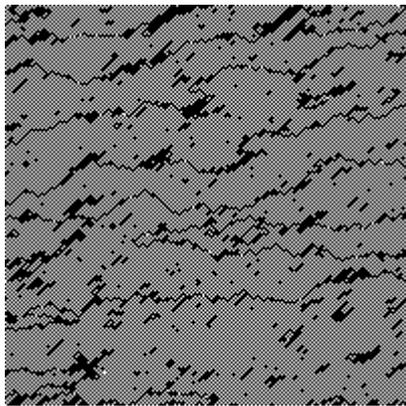
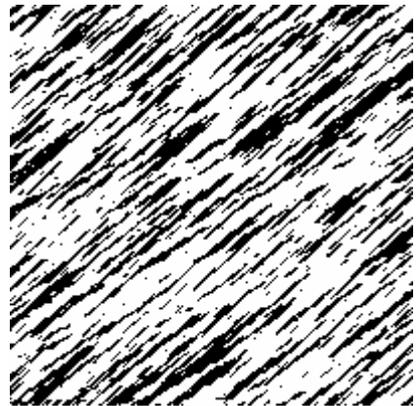
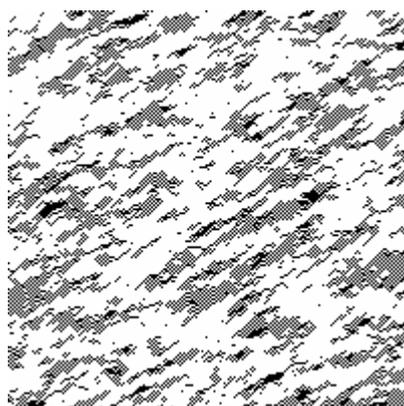
### 5.1 – Introdução

O objetivo deste capítulo é mostrar os resultados obtidos ao utilizar o modelo proposto e implementado descrito no capítulo 4 deste trabalho. São mostrados os resultados da entropia dos *foot patterns* obtidos do algoritmo genético da mínima entropia modificado (AGMEM) e as imagens sintetizadas dos modelos estocásticos de textura encontrados pelo algoritmo genético de modelos de textura modificado (AGMTM).

Com o objetivo de verificar o desempenho do AGMEM e do AGMTM propostos e implementados neste trabalho foram realizados testes com os algoritmos genéticos da mínima entropia (AGME) e dos modelos de textura (AGMT) [11]. Em todos esses testes foram utilizados diferentes tipos de amostras de texturas. Os resultados dos testes com o algoritmo AGMEM são apresentados em forma de gráficos de aptidão das populações, disposição dos pixels nos *foot patterns* e valores de entropia obtidos. Os resultados obtidos ao utilizar o AGMTM são mostrados em gráficos de aptidão das populações e da síntese de uma amostra dos modelos estocásticos de textura encontrados por esse algoritmo.

### 5.2 – Amostras Originais

A avaliação do AGMEM e do AGMTM propostos e implementados neste trabalho foi realizada usando como amostras as imagens mostradas na Figura 5.1 utilizadas no trabalho de Ashlock e Davidson [11].

*(a)**(b)**(c)**(d)**(e)*

*Figura 5.1 – Cinco amostras de texturas utilizados nesse trabalho:  
(a) pipes; (b) diagonals; (c) filaments; (d) thick diagonals; (e) clouds.*

As imagens da Figura 5.1 são imagens binárias com 200 x 200 pixels com dois níveis de cinza [11]. O valor zero corresponde à cor preta e o valor um à cor branca.

### 5.3 – Resultados Obtidos no AGMEM

O objetivo do AGMEM é escolher o *foot pattern* com a menor entropia para a amostra de textura utilizada como entrada desse algoritmo. Os dados fornecidos a esse algoritmo são:

- imagem da amostra de textura com 200 x 200 pixels [11];
- número de indivíduos da população;
- número de gerações desejadas;
- valor da taxa de elitismo;
- valor da taxa de *crossover*,  $e$ ;
- valor da taxa de mutação.

Após realizar vários testes, os valores usados para o número de indivíduos, número de gerações, taxa de elitismo, taxa de *crossover* e taxa de mutação são os valores encontrados que produziram os melhores resultados para a entropia nos testes realizados com o algoritmo AGMEM. Esses valores são os mesmos para as 5 amostras utilizadas neste trabalho. A Tabela 5.1 mostra esses valores e os valores usados no AGME [11].

Tabela 5.1 – Valores utilizados no AGMEM e no AGME.

Parâmetros	AGME [11]	AGMEM
Nº indivíduos da população	200	200
Nº de gerações desejadas	200	200
Taxa de elitismo	-	0,02
Taxa de <i>crossover</i>	1	1
Taxa de mutação	0,5	0,5

Nos testes realizados utilizou-se cada amostra de textura da Figura 5.1.

O gráfico da Figura 5.2 (a) mostra a evolução da população do AGMEM para a amostra de textura *pipes*, ilustrada na Figura 5.1 (a). Nesse gráfico a cor azul mostra o valor da entropia do melhor indivíduo da população a cada geração e a cor vermelha ilustra a média dos valores da entropia de todos os indivíduos da população a cada geração. A Figura 5.2 (b) mostra o melhor *foot pattern* encontrado por esse algoritmo para essa textura.

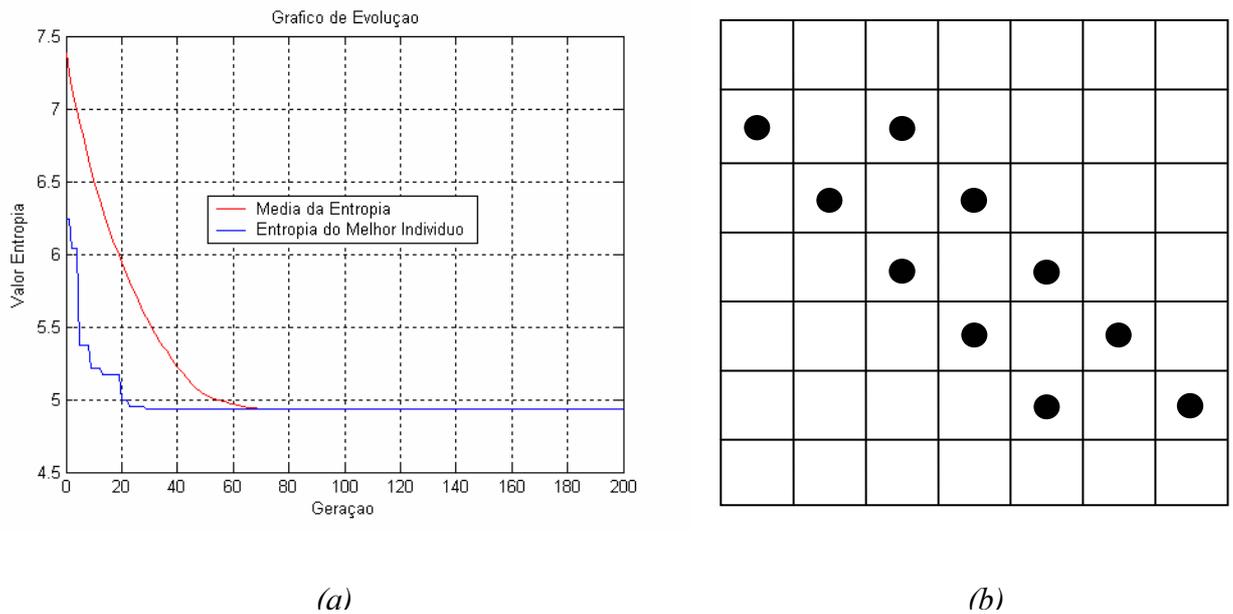
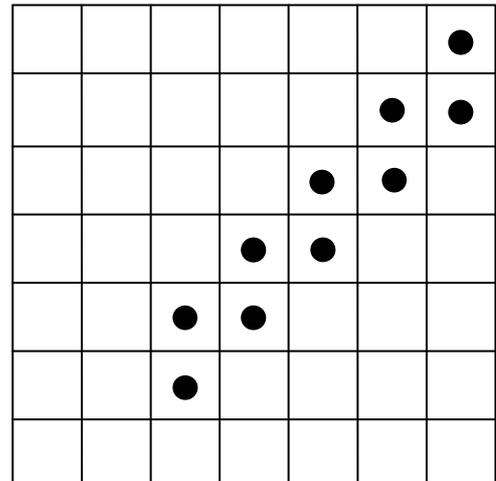
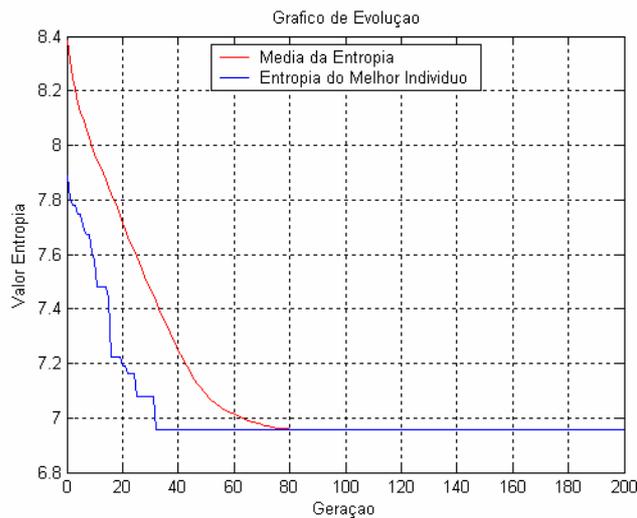


Figura 5.2 – (a) Gráfico da evolução da população do AGMEM para a amostra de textura pipes; (b) melhor foot pattern encontrado para essa textura.

Como citado no capítulo 4 deste trabalho, quanto menor o valor da entropia, mais apto é o indivíduo. Pode-se observar na Figura 5.2(a) uma grande diminuição dos valores da média da entropia até a geração 70 e também uma substancial diminuição do valor da entropia do melhor indivíduo. O menor valor da entropia desse indivíduo ocorreu na geração 30. A média dos valores da entropia e o valor da entropia do melhor indivíduo estão bem próximos a partir da geração 70. Pode-se verificar que logo nas primeiras gerações, essas reduções dos valores observadas nas duas curvas da Figura 5.2(a), refletem o efeito das modificações realizadas no AGMEM que são: a ordenação dos indivíduos em ordem decrescente de aptidão (como uma nova forma de seleção deles para a reprodução), o operador de elitismo e o algoritmo guloso. É importante observar nessa figura que nos valores mostrados pelas duas curvas não ocorreu piora da aptidão ou aumento do valor da melhor entropia ou da média de uma geração para a geração seguinte e isso se deve, principalmente, ao algoritmo guloso e ao operador de elitismo.

Pode-se verificar que na Figura 5.2(a) o valor da entropia para o *foot pattern* da textura *pipes* encontrado pelo AGMEM foi de 4,9368. Esse valor foi obtido na 30<sup>a</sup> geração. No AGME o melhor valor encontrado para a entropia foi de 5,29942 [17] e para encontrar esse valor foram necessárias 200 gerações.

O gráfico da Figura 5.3(a) mostra a evolução da população do AGMEM para a amostra de textura *diagonals*, ilustrada na Figura 5.1(b). Nesse gráfico a cor azul mostra o valor da entropia do melhor indivíduo da população a cada geração e a cor vermelha ilustra a média dos valores da entropia de todos os indivíduos da população a cada geração. A Figura 5.3(b) mostra o melhor *foot pattern* encontrado por esse algoritmo para essa textura.



(a)

(b)

Figura 5.3 – (a) Gráfico da evolução da população do AGMEM para a amostra de textura *diagonals*; (b) melhor *foot pattern* encontrado para essa textura.

Pode-se observar na Figura 5.3(a) uma diminuição acentuada nos valores mostrados das duas curvas. A média das entropias atinge seu menor valor próximo da geração 80 e o valor da entropia do melhor indivíduo atinge seu mínimo valor na geração 33. Além da vantagem de reduzir, consideravelmente, o número de gerações necessárias para alcançar

bons valores de entropia, o método de ordenação, o operador de elitismo e o algoritmo guloso, também, propiciam sempre uma melhora de uma geração para outra. Nunca ocorre um aumento da melhor entropia ou da média da entropia dos indivíduos de uma geração para a geração seguinte.

Na 33<sup>a</sup> geração o valor da entropia do *foot pattern* representado espacialmente na Figura 5.3(b) encontrado pelo AGMEM é de 6,95. No AGME o melhor valor encontrado foi de 7,1 [17] e foram necessárias 200 gerações para encontrá-lo.

O gráfico da Figura 5.4(a) mostra a evolução da população do AGMEM para a amostra da textura *filaments*, ilustrada na Figura 5.1(c). Nesse gráfico a cor azul mostra o valor da entropia do melhor indivíduo da população a cada geração e a cor vermelha ilustra a média dos valores da entropia de todos os indivíduos da população a cada geração. A Figura 5.4(b) mostra o melhor *foot pattern* encontrado por esse algoritmo para essa textura.

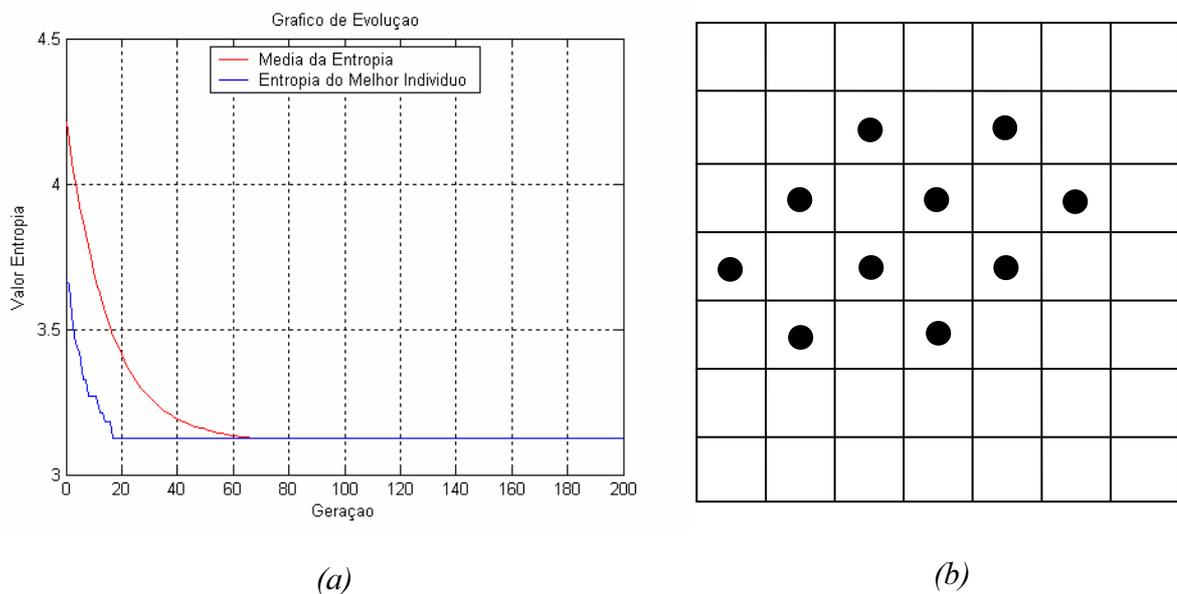


Figura 5.4 – (a) Gráfico da evolução da população do AGMEM para amostra de textura *filaments*; (b) melhor *foot pattern* encontrado para essa textura.

Pode-se observar na Figura 5.4(a) uma diminuição acentuada dos valores da média da entropia até a geração 69 e aconteceu o mesmo com o valor da entropia do melhor indivíduo. O menor valor da entropia do melhor indivíduo ocorreu na geração 18. A média dos valores da entropia e o valor da entropia do melhor indivíduo estão bem próximos a partir da geração 69. Nessa figura, é importante observar que para os dois valores, representados pelas curvas, não ocorreu piora da aptidão ou aumento do valor da melhor entropia ou da média de uma geração para a geração seguinte, isso aconteceu devido ao método de seleção, ao algoritmo guloso e ao operador de elitismo utilizados no AGMEM. Nas primeiras gerações as acentuadas reduções dos valores refletem também o efeito dessas modificações.

O valor da entropia do *foot pattern* representado espacialmente na Figura 5.4 (b) encontrado pelo AGMEM para a textura *filaments* é de 3,1163. Esse valor ocorreu na 18<sup>a</sup> geração. O *foot pattern* encontrado pelo AGME com melhor valor de entropia foi de 3,52305 [17] para essa textura e foram necessárias 200 gerações para encontrá-lo.

O gráfico da Figura 5.5(a) mostra a evolução da população do AGMEM para a amostra de textura *thick diagonals*, ilustrada na Figura 5.1(d). Nesse gráfico a cor azul mostra o valor da entropia do melhor indivíduo da população a cada geração e a cor vermelha ilustra a média dos valores da entropia de todos os indivíduos da população a cada geração. A Figura 5.5(b) mostra o melhor *foot pattern* encontrado por esse algoritmo para essa textura.

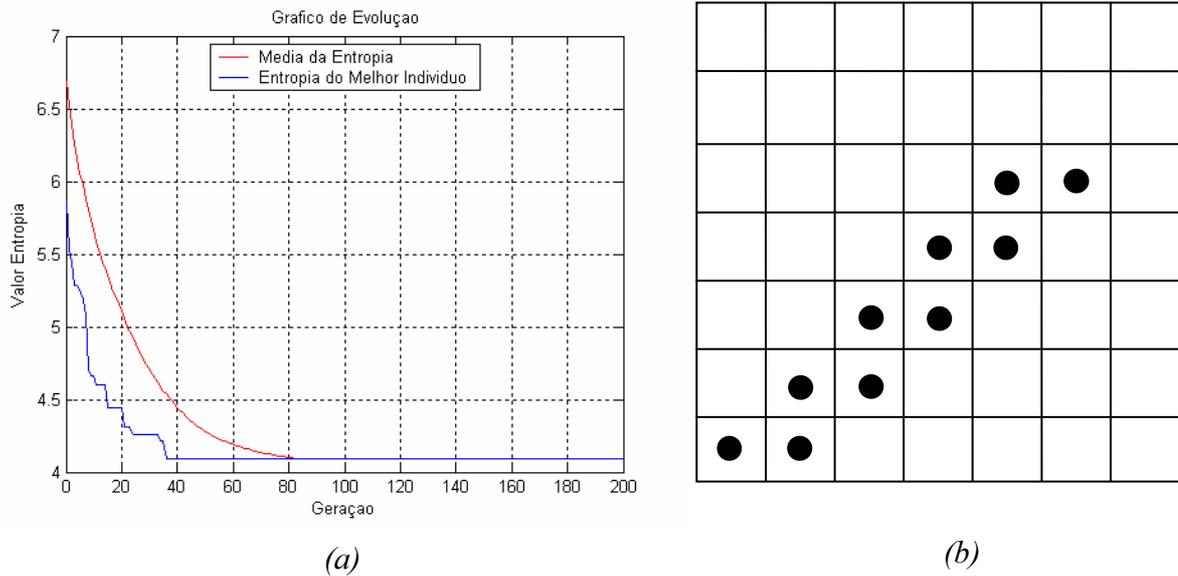


Figura 5.5 – (a) Gráfico da evolução da população do AGMEM para a amostra de textura *thick diagonals*; (b) melhor *foot pattern* encontrado para essa textura.

Pode-se observar na Figura 5.5(a) uma diminuição acentuada nos valores representados pelas duas curvas. O valor mínimo da média ocorreu próximo da geração 87 e o valor mínimo da entropia do melhor indivíduo ocorreu na geração 37. O operador de elitismo e o algoritmo guloso propiciam sempre uma melhora de uma geração para outra. Nunca ocorre um aumento da melhor entropia ou da média da entropia dos indivíduos de uma geração para a geração seguinte. Logo nas primeiras gerações esse operador e esse algoritmo junto com o método de ordenação e seleção dos indivíduos são os responsáveis pelas reduções dos valores da média da entropia e da entropia do melhor indivíduo.

O valor da entropia do *foot pattern* mostrado espacialmente na Figura 5.5(b) encontrado pelo AGMEM é 4,0915. Esse valor ocorreu a partir da 29<sup>a</sup> geração. No AGME a entropia do melhor *foot pattern* da amostra de textura *thick diagonals* é 4,5629 [17] e foram necessárias 200 gerações para encontrá-lo.

O gráfico da Figura 5.6(a) mostra a evolução da população do AGMEM para a amostra de textura *clouds*, ilustrada na Figura 5.1(e). Nesse gráfico a cor azul mostra o valor da entropia do melhor indivíduo da população a cada geração e a cor vermelha ilustra a média dos valores da entropia de todos os indivíduos da população a cada geração. A Figura 5.6(b) mostra o melhor *foot pattern* encontrado por esse algoritmo para essa textura.

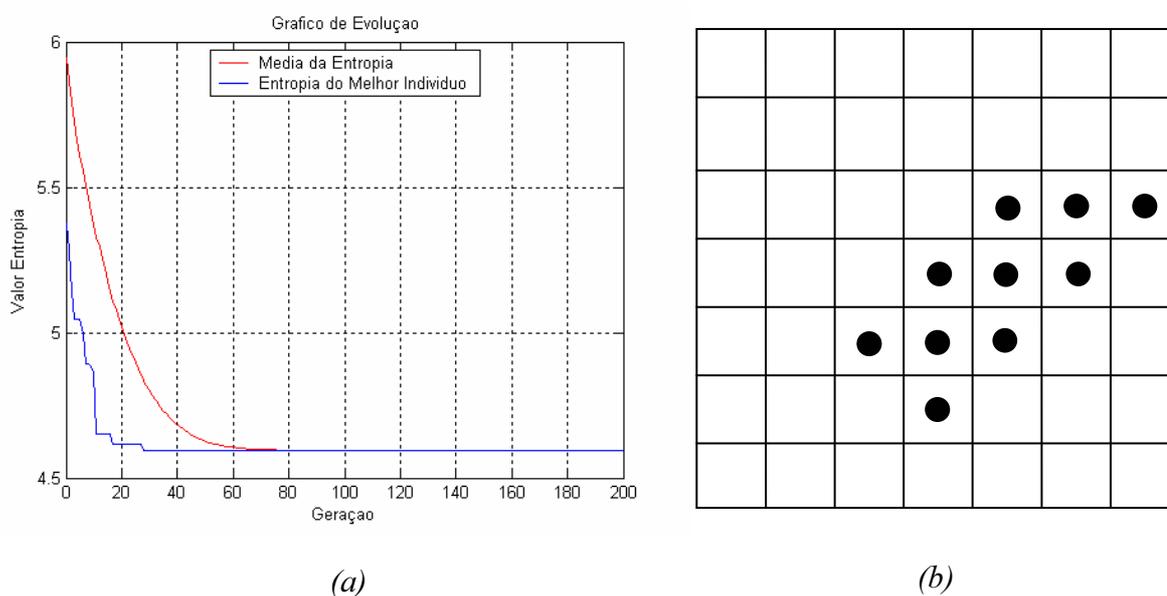


Figura 5.6 – (a) Gráfico da evolução da população do AGMEM para a amostra de textura *clouds*; (b) melhor *foot pattern* encontrado para essa textura.

Pode-se observar na Figura 5.6(a) uma acentuada redução dos valores da média da entropia até a geração 60 e o mesmo ocorreu com o valor da entropia do melhor indivíduo. O menor valor de entropia de um indivíduo ocorreu na geração 29. A média dos valores da entropia e o valor da entropia do melhor indivíduo estão bem próximos a partir da geração 70. Nesta figura, é importante observar que nos dois valores, representados pelas duas curvas, não ocorreu piora da aptidão ou aumento do valor da melhor entropia ou da média da entropia de uma geração para a geração seguinte e isso se deve ao método de ordenamento e de seleção

dos indivíduos, ao algoritmo guloso e ao operador de elitismo utilizados no AGMEM. Logo nas primeiras gerações as acentuadas diminuições observadas nas duas curvas refletem também o efeito do método de ordenação e seleção dos indivíduos, do operador de elitismo e do algoritmo guloso.

O valor da entropia do *foot pattern* da amostra de textura *clouds* mostrado espacialmente na Figura 5.6(b) encontrado pelo AGMEM é 4,5959. Esse valor ocorreu a partir da 29<sup>a</sup> geração. Para o algoritmo AGME a entropia do melhor indivíduo é 5,22743 [17] e foram necessárias 200 gerações para encontrá-la.

Nos testes realizados neste capítulo com as 5 amostras de texturas da Figura 5.1(a), pode-se verificar que tanto as médias dos valores das entropias dos indivíduos a cada geração como as entropias dos melhores indivíduos a cada geração ou melhoram ou se mantêm inalteradas de uma geração para a geração seguinte. A média da entropia dos indivíduos quase sempre melhora com a utilização do algoritmo guloso. Este algoritmo, como foi visto no capítulo 3 deste trabalho, impede que os indivíduos de uma geração sejam modificados se os novos indivíduos possuem menor aptidão, ou um maior valor de entropia. A entropia do melhor indivíduo não piora, primeiro devido ao operador de elitismo que exclui os melhores indivíduos do *crossover* e da mutação, segundo devido ao algoritmo guloso que escolhe sempre os indivíduos mais aptos para passarem para as próximas gerações. A rápida evolução da população nas primeiras gerações, mostrada na curva da média da entropia da população, deve-se ao método de ordenação e seleção, ao operador de elitismo e ao algoritmo guloso.

#### 5.4 – Resultados Obtidos no AGMTM

O AGMTM realiza a escolha do melhor modelo estocástico que representa o tipo de amostra de textura. Os dados fornecidos a esse algoritmo são:

- imagem da amostra de textura com 200 x 200 pixels;
- *foot pattern* escolhido pelo AGMEM;
- número de indivíduos utilizados na população ou modelos de textura que são representados por vetores de dimensão 16;
- número de gerações desejadas;
- valor da taxa de elitismo;
- valor da taxa de *crossover*, e;
- valor da taxa de mutação.

Os valores usados para o número de indivíduos, número de gerações, taxa de elitismo, taxa de *crossover* e taxa de mutação foram valores que produziram os melhores resultados nos testes realizados com o AGMTM. Esses valores são os mesmos para as 5 amostras de texturas utilizadas neste trabalho. A Tabela 5.2 mostra esses valores e os valores utilizados no AGMT [11].

Tabela 5.2 – Valores utilizados pelo AGMT e pelo AGMTM

Parâmetros	AGMT [11]	AGMTM
Nº de indivíduos da população	200	200
Nº de gerações desejadas	200	200
Taxa de elitismo	-	0,02
Taxa de <i>crossover</i>	0,5	0,5
Taxa de mutação	0,75	0,75

Foram realizados vários testes com o AGMTM utilizando-se cada uma das 5 amostras de texturas mostradas na Figura 5.1. Os resultados obtidos nesses testes são mostrados nos gráficos de evolução e nas texturas sintetizadas pelos modelos estocásticos selecionados por esse algoritmo. Nesses gráficos a curva em vermelho é a média do quadrado da diferença dos histogramas de eventos entre a amostra original e cada modelo sintetizado por cada indivíduo da população que é denominado neste trabalho de forma compacta na análise dos resultados deste trabalho de média das diferenças. A curva em azul é o valor do quadrado da diferença dos histogramas de eventos da amostra da textura original e da amostra sintetizada pelo melhor indivíduo da população que é denominado nesta dissertação de forma compacta na análise dos resultados de diferença do melhor indivíduo. Nesses testes, os indivíduos mais aptos são aqueles que possuem o menor valor desse quadrado da diferença dos histogramas. Foi utilizado no final da execução do AGMTM um programa que sintetiza uma amostra utilizando o vetor de modelo estocástico obtido por esse algoritmo, como explicado na subseção 4.3.5 do capítulo 4 deste trabalho.

O gráfico da Figura 5.7(a) mostra a evolução da população do AGMTM para a amostra de textura *pipes* mostrada na Figura 5.1(a). Nesse gráfico a cor azul ilustra o valor da

diferença do melhor indivíduo da população a cada geração e a cor vermelha mostra a média das diferenças de todos os indivíduos da população a cada geração. A Figura 5.7(b) ilustra o melhor *foot pattern* encontrado pelo AGMEM e o valor da sua entropia. Esse *foot pattern* é usado pelo AGMTM para encontrar um modelo estocástico que represente com o menor erro a textura da amostra fornecida.

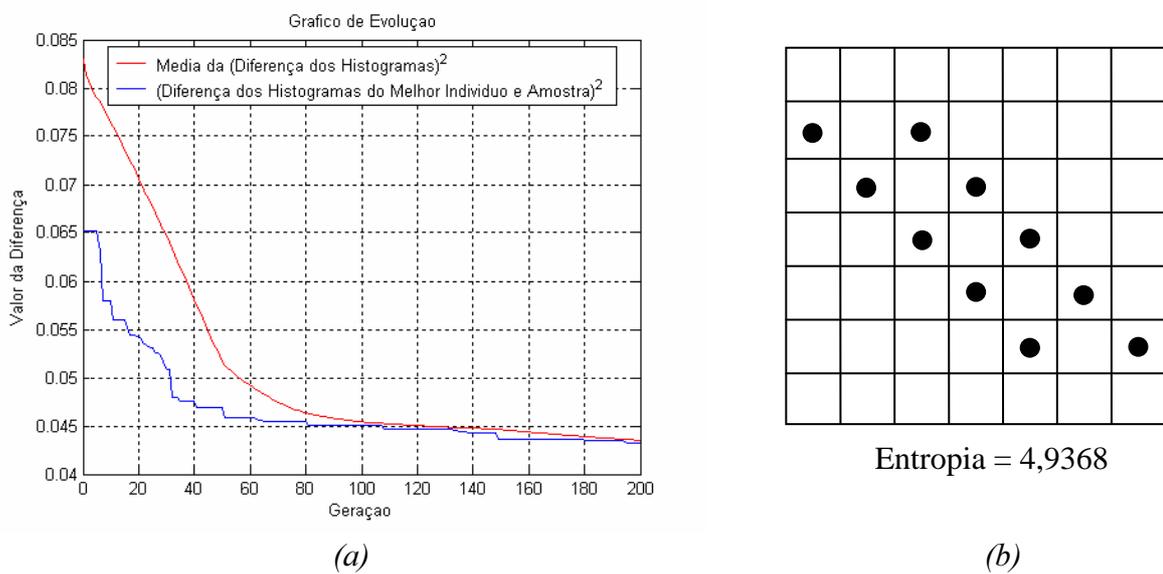


Figura 5.7 – (a) Gráfico da evolução da população para a textura pipes;

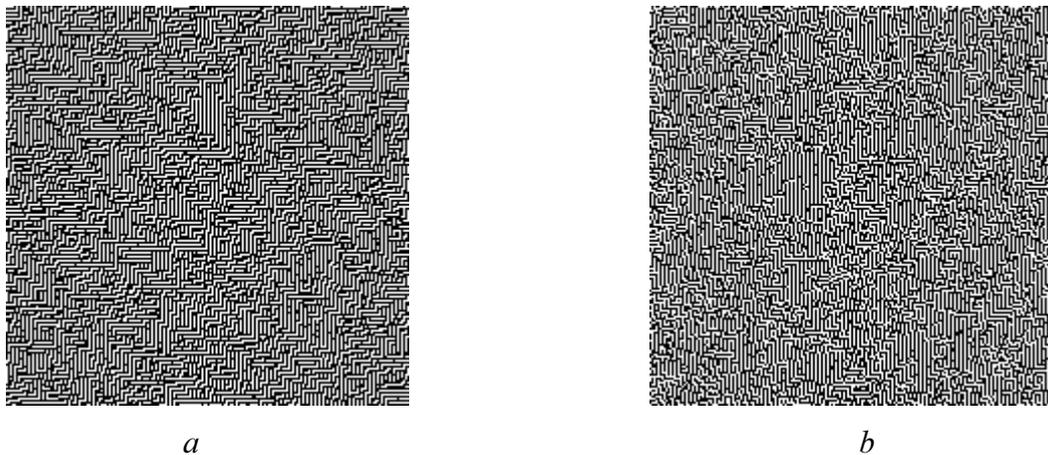
(b) *foot pattern* utilizado pelo AGMTM para essa textura.

Pode-se observar na Figura 5.7(a) no início uma diminuição nos valores da média das diferenças e da diferença do melhor indivíduo. Os valores da média das diferenças mostram uma redução acentuada no início das gerações e próximo à geração 100 o valor está bem próximo do valor da diferença do melhor indivíduo e do valor final da média das diferenças. Por sua vez, os valores da diferença do melhor indivíduo mostram uma diminuição mais acentuada dos valores até a geração 82. As reduções desses valores, representadas pelas duas

curvas, são devido ao novo método de ordenação e seleção de indivíduos para a reprodução, o operador de elitismo e o algoritmo guloso. O método de ordenação e seleção garante que sempre os melhores indivíduos são selecionados para passar pelos operadores genéticos de elitismo, de *crossover* e de mutação. O operador de elitismo preserva os melhores indivíduos de toda a população a cada geração. Finalmente o algoritmo guloso só permite modificações nos indivíduos se elas melhorarem a sua aptidão.

Pode-se verificar na Figura 5.7(a) que na geração 100 o valor da média das diferenças é de 0,045449 e o valor final dessa média na geração 200 é de 0,043514. Na geração 82 o valor da diferença do melhor indivíduo é 0,045001. Depois dessa geração, esse valor diminui quatro vezes, mais lentamente chegando na geração 200 a 0,043256.

A Figura 5.8(a) mostra a amostra da textura *pipes* original [11] e a Figura 5.8(b) a amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do AGMTM.



*Figura 5.8 - (a) Amostra original da textura pipes; (b) amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do AGMTM.*

O gráfico da Figura 5.9(a) mostra a evolução da população do AGMTM para a amostra de textura *diagonals* ilustrada na Figura 5.1(b). Nesse gráfico a cor azul mostra o valor da diferença do melhor indivíduo da população a cada geração e a cor vermelha ilustra a média das diferenças de todos os indivíduos da população a cada geração. A Figura 5.9(b) mostra o melhor *foot pattern* encontrado pelo AGMEM e o valor da sua entropia. Esse *foot pattern* é usado pelo AGMTM para encontrar um modelo estocástico que mais se aproxime dessa amostra, ou seja, com o menor erro em relação à amostra de textura original fornecida.

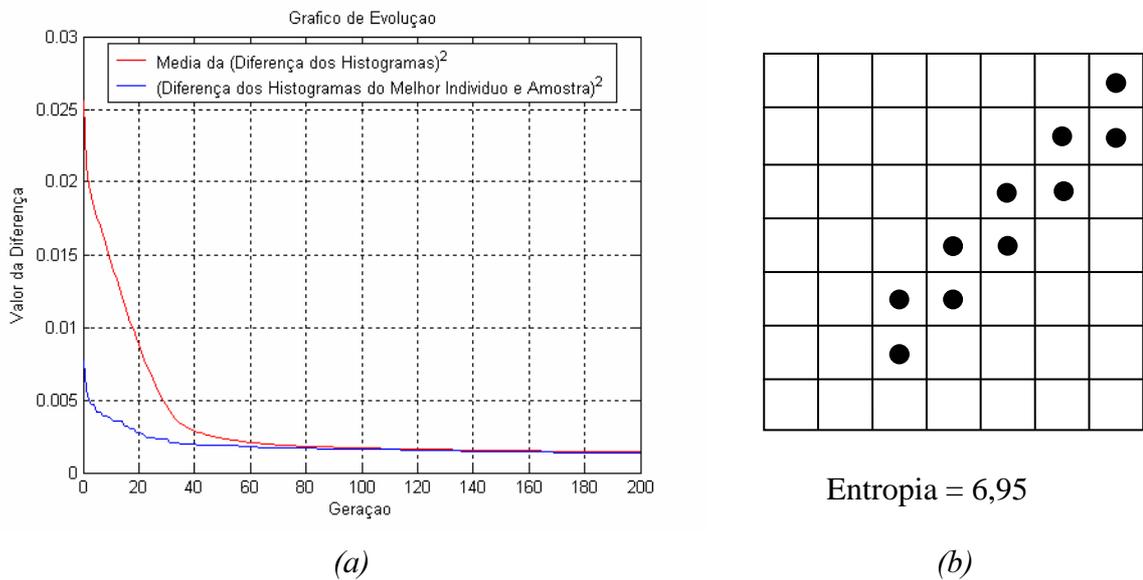


Figura 5.9 – (a) Gráfico da evolução da população para a textura *diagonals*;

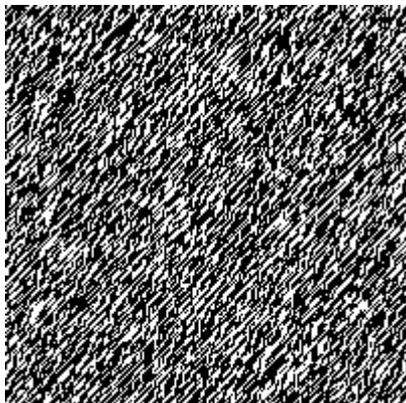
(b) *foot pattern* utilizado pelo AGMTM para essa textura.

Pode-se observar no gráfico da evolução da Figura 5.9(a) nas primeiras gerações uma acentuada redução dos valores das duas curvas. A média das diferenças mostra uma diminuição dos valores no início das gerações até, aproximadamente, a geração 80 e próximo à geração 100, esses valores estão bem próximos do valor da diferença do melhor indivíduo e do valor final da média das diferenças. Por sua vez, os valores da diferença do melhor

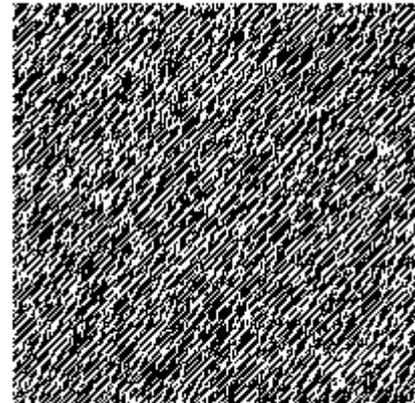
indivíduo mostram uma acentuada redução até a geração 42. Essas reduções desses valores, ilustradas pelas duas curvas, são devido ao novo método de ordenação e seleção de indivíduos para a reprodução, ao operador de elitismo e ao algoritmo guloso da mesma forma como foi explicado neste capítulo para a textura *pipes*.

Na geração 80 na Figura 5.9 (a) o valor da média das diferenças é 0,00182 e na geração 200 esse valor é 0,00140. Na geração 42 o valor da diferença do melhor indivíduo é 0,00192. Depois dessa geração, esse valor reduz mais lentamente chegando na geração 200 a 0,00134.

A Figura 5.10(a) mostra a amostra original da textura *diagonals* [11] e a Figura 5.10(b) ilustra a amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do AGMTM.



(a)



(b)

Figura 5.10 - (a) Amostra original da textura *diagonals*;

(b) amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do AGMTM.

O gráfico da Figura 5.11(a) mostra a evolução da população do AGMTM para a amostra de textura *filaments* ilustrada na Figura 5.1(c). Nesse gráfico a cor azul mostra o valor da diferença do melhor indivíduo da população a cada geração e a cor vermelha ilustra a média das diferenças de todos os indivíduos da população a cada geração. A Figura 5.11(b) mostra o melhor *foot pattern* encontrado pelo AGMEM e o valor da sua entropia. Esse *foot pattern* é usado pelo AGMTM para encontrar um modelo estocástico que mais se aproxime dessa amostra, ou seja, com o menor erro em relação à amostra da textura original fornecida.

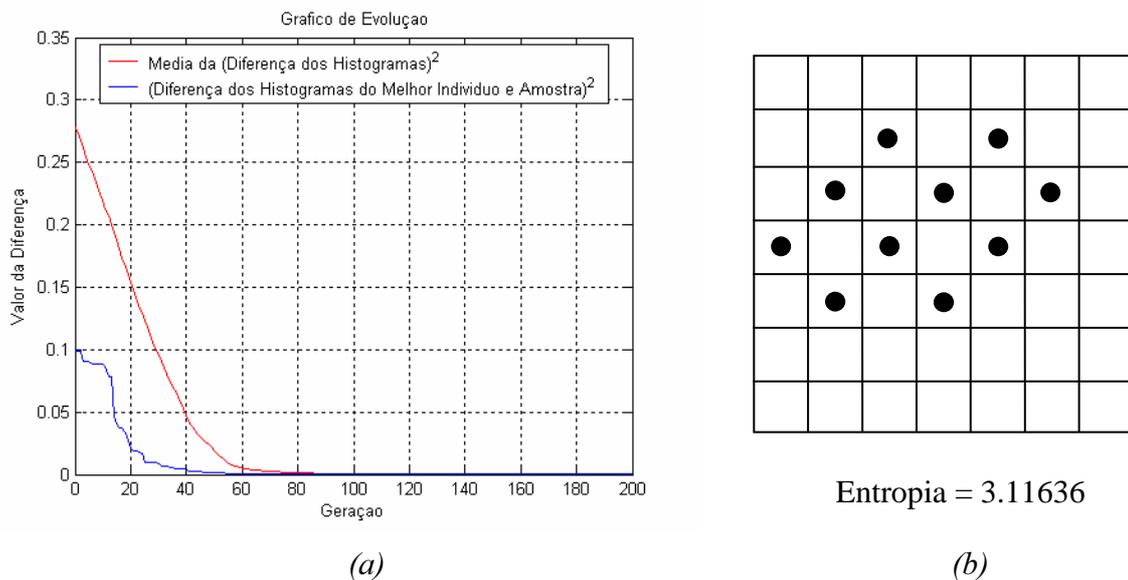


Figura 5.11 – (a) Gráfico da evolução da população da textura *filaments*;

(b) *foot pattern* utilizado pelo AGMTM para essa textura.

Nas primeiras gerações do gráfico da Figura 5.11(a), pode-se observar uma diminuição acentuada dos valores da média das diferenças e da diferença do melhor indivíduo. Esses valores da média das diferenças reduzem bastante até a geração 55, depois essa redução diminui. Aproximadamente na geração 84 o valor da média das diferenças está bem próximo do valor da diferença do melhor indivíduo e do valor final da média das diferenças. Por sua

vez, os valores do melhor indivíduo mostram uma redução mais acentuada até a geração 40. Após essa geração, o valor da diferença do melhor indivíduo continua a diminuir, porém de forma menos acentuada. Como já foi citado nos testes anteriores com o AGMTM essas reduções nos valores da média das diferenças e da diferença do melhor indivíduo são, também, devido ao novo método de ordenação e seleção de indivíduos para a reprodução, ao operador de elitismo e ao algoritmo guloso.

Pode-se verificar na Figura 5.11(a) que na geração 80 o valor da média das diferenças é 0,0011698 e o valor dessa média na geração 200 é 0,0005254. Na geração 60, o valor da diferença do melhor indivíduo é 0,0006898. Depois dessa geração, o valor dessa diferença diminui mais lentamente chegando na geração 200 a 0,0005206.

A Figura 5.12(a) mostra a amostra original da textura *filaments* [11] e a Figura 5.17(b) ilustra a amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do algoritmo AGMTM.

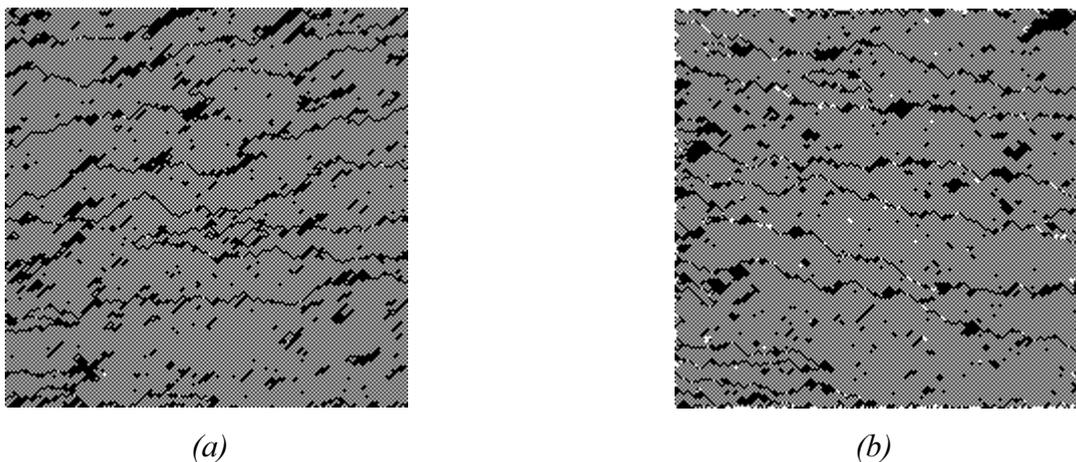


Figura 5.12 - (a) Amostra original da textura *filaments*;

(b) amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do  
*AGMTM*.

O gráfico da Figura 5.13

Da geração 20 até a geração 40 ocorre uma diminuição mais acentuada dos valores da diferença do melhor indivíduo. Após a geração 40, esses valores continuam a cair, porém de forma menos intensa. Como já foi citado nos testes realizados neste capítulo, essas reduções acentuadas nos valores da média das diferenças e da diferença do melhor indivíduo são, também, devido ao novo método de ordenação e seleção de indivíduos para a reprodução, ao operador de elitismo e ao algoritmo guloso.

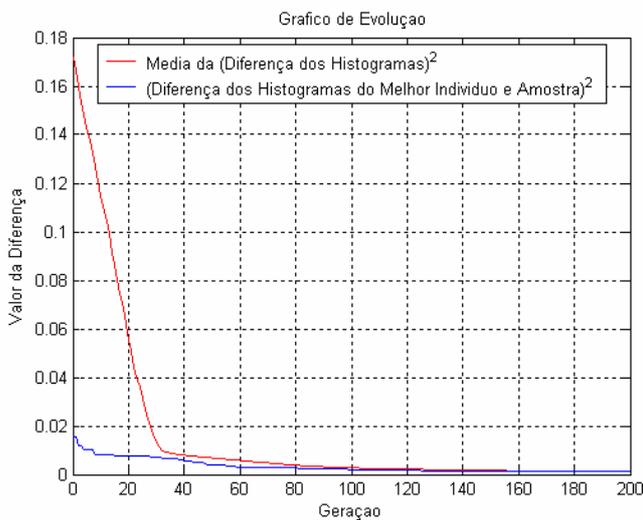
Pode-se verificar na Figura 5.13 que na geração 80 o valor da média das diferenças é 0,002369 e na geração 200, esse valor é 0,001064. Na geração 40 o valor da diferença do melhor indivíduo é 0,004269. Depois dessa geração esse valor diminui mais lentamente chegando na geração 200 a 0,001057.

A Figura 5.14(a) mostra a amostra original da textura *thick diagonals* [11] e a Figura 5.14(b) ilustra a amostra dessa textura sintetizada usando modelo de textura obtido do AGMTM.

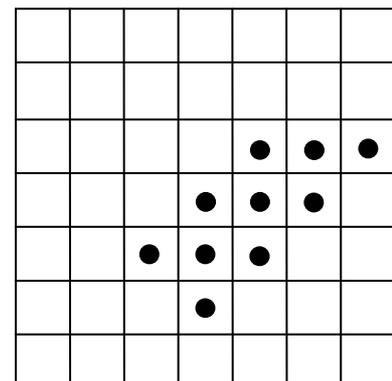


Figura 5.14 - (a) Amostra original da textura *thick diagonals*;  
(b) amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do AGMTM.

O gráfico da Figura 5.15(a) mostra a evolução da população do AGMTM para a amostra de textura *clouds* ilustrada na Figura 5.1(e). Nesse gráfico a cor azul mostra o valor da diferença do melhor indivíduo da população a cada geração e a cor vermelha ilustra a média das diferenças de todos os indivíduos da população a cada geração. A Figura 5.15(b) mostra o melhor *foot pattern* encontrado pelo AGMEM e o valor da sua entropia. Esse *foot pattern* é usado pelo AGMTM para encontrar um modelo estocástico que mais se aproxime dessa amostra de textura, ou seja, com o menor erro em relação à amostra da textura original fornecida.



(a)



Entropia = 4.5959

(b)

Figura 5.15 – (a) Gráfico da evolução da população da textura *clouds*;

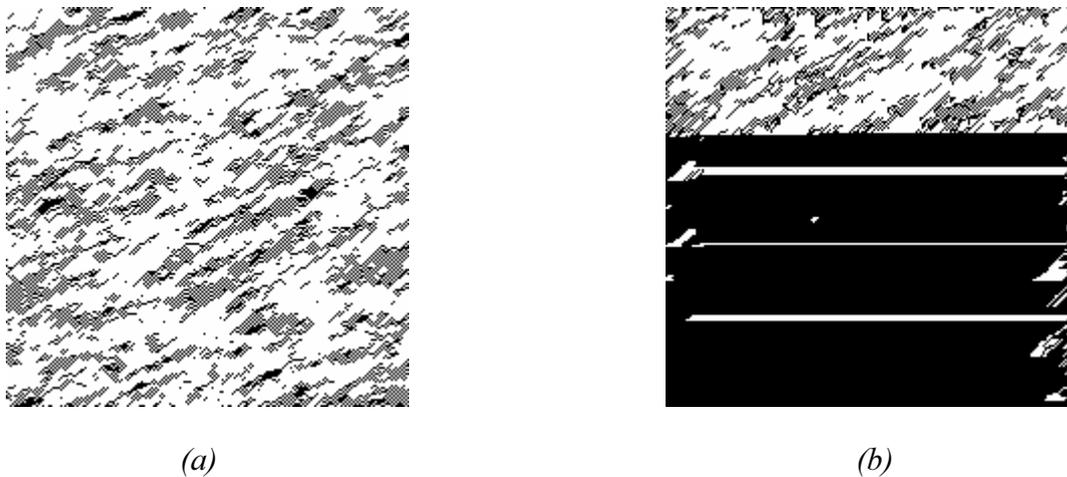
(b) *foot pattern* utilizado pelo AGMTM para essa textura.

Logo nas primeiras gerações, pode-se observar no gráfico da Figura 5.15(a), uma grande diminuição dos valores da média das diferenças persistindo até a geração 40. Depois dessa geração, os valores da média das diferenças continuam diminuindo, porém mais lentamente chegando próximo da geração 100, a um valor bem próximo do valor da diferença do melhor indivíduo. Até a geração 60 os valores dessa diferença diminuem bastante. Após

essa geração os valores dessa diferença continuam a diminuir, porém mais lentamente. Como citado nos testes realizados neste capítulo com o AGMTM, essas diminuições nos valores das duas curvas são, também, devido ao novo método de ordenação e seleção de indivíduos para a reprodução, ao operador de elitismo e ao algoritmo guloso.

Pode-se verificar na Figura 5.15(a) que na geração 40 o valor da média das diferenças é de 0,008111, nas gerações 100 e 200 esse valor é 0,002748 e 0,001258, respectivamente. O valor da diferença do melhor indivíduo na geração 60 é de 0,003117. Depois dessa geração esse valor cai mais lentamente chegando a 0,001184 na geração 200.

A Figura 5.16(a) mostra a amostra original da textura *clouds* [11] e a Figura 5.16(b) ilustra a amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do AGMTM.



*Figura 5.16 - (a) Amostra original da textura clouds;*  
*(b) amostra dessa textura sintetizada usando o modelo estocástico de textura obtido do*  
*AGMTM.*

## 5.5 - Análise dos Resultados Obtidos do AGMEM e do AGMTM

### 5.5.1 – Análise dos resultados obtidos do AGMEM

A Figura 5.17 e a Tabela 5.3 mostram os valores da entropia obtidos do AGMEM e do AGME [17] quando foram utilizadas as amostras mostradas na Figura 5.1.

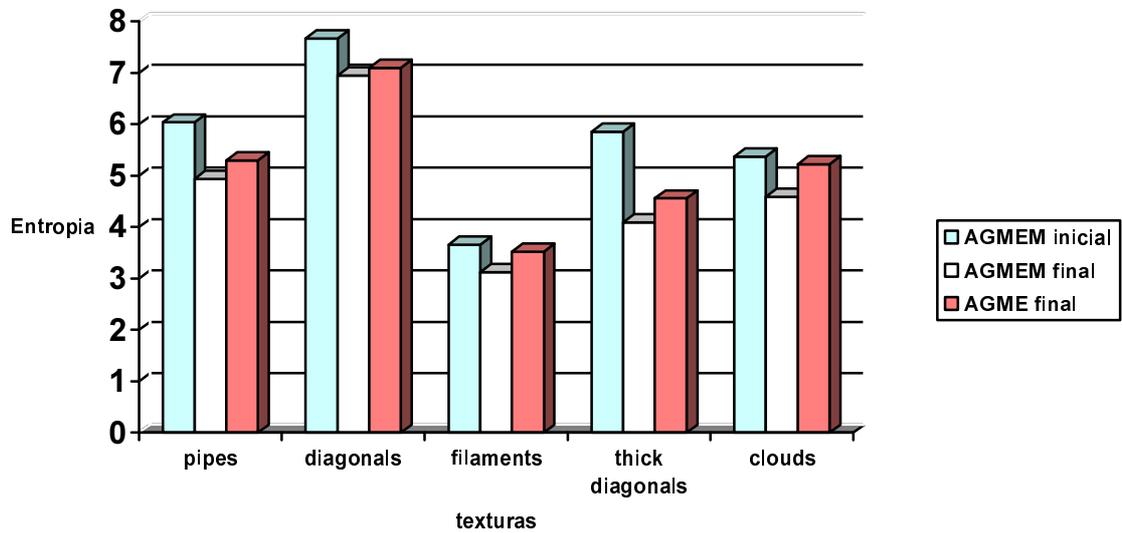


Figura 5.17 – Valores da entropia dos foot patterns obtidos do AGME e do AGMEM.

Tabela 5.3 – Valores de entropia obtidos do AGME e do AGMEM.

<b>Textura</b>	<b>Valor</b>	<b>AGME</b>	<b>AGMEM Menor Entropia</b>	<b>AGMEM Média da Entropia</b>
<i>Pipes</i>	Inicial	-	6,242	7,383
	Geração 200	5,29942	4,9368	4,9368
	Geração (menor valor)	-	30	75
<i>diagonals</i>	Inicial	-	7,8856	8,3828
	Geração 200	7,1	6,9574	6,9574
	Geração (menor valor)	-	33	83
<i>filaments</i>	Inicial	-	3,6649	4,2137
	Geração 200	3,5230	3,1246	3,1246
	Geração (menor valor)	-	18	69
<i>Thick diagonals</i>	Inicial	-	5,8636	6,6859
	Geração 200	4,5629	4,0915	4,0915
	Geração (menor valor)	-	37	87
<i>Clouds</i>	Inicial	-	5,3764	5,9474
	Geração 200	5,2274	4,5959	4,5959
	Geração (menor valor)	-	29	78

Os valores da entropia dos *foot patterns* ilustrados na Tabela 5.3 e na Figura 5.17 mostram, claramente, que os *foot patterns* encontrados pelo AGMEM possuem um valor menor de entropia que os *foot patterns* encontrados pelo AGME. Os valores desse último algoritmo foram obtidos na página do autor Ashlock [17]. As diferenças da melhor entropia do AGMEM comparadas com as diferenças da melhor entropia do AGME são: 7,34% para a textura *pipes*, 2,15% para a textura *diagonals*, 12,75% para a textura *filaments*, 11,52% para a

textura *thick diagonals* e 13,74% para a textura *clouds*. Pode-se observar nessa tabela e nessa figura que os resu

amostras da Figura 5.1, o número de gerações necessárias para obter esses *foot patterns* de 81,5% a 91%.

A Tabela 5.4 e a Figura 5.18 mostram o tempo total de processamento do AGME e do AGMEM considerando-se os melhores valores de entropia.

Tabela 5.4 – Comparação do tempo total de processamento do AGME e do AGMEM.

Tempo total	<i>pipes</i>	<i>diagonals</i>	<i>filaments</i>	<i>thick diagonals</i>	<i>clouds</i>
AGME (seg.)	2100	2100	2100	2100	2100
AGMEM (seg.)	450	495	270	555	435
Redução(%)	78,57	76,42	87,14	73,57	79,28

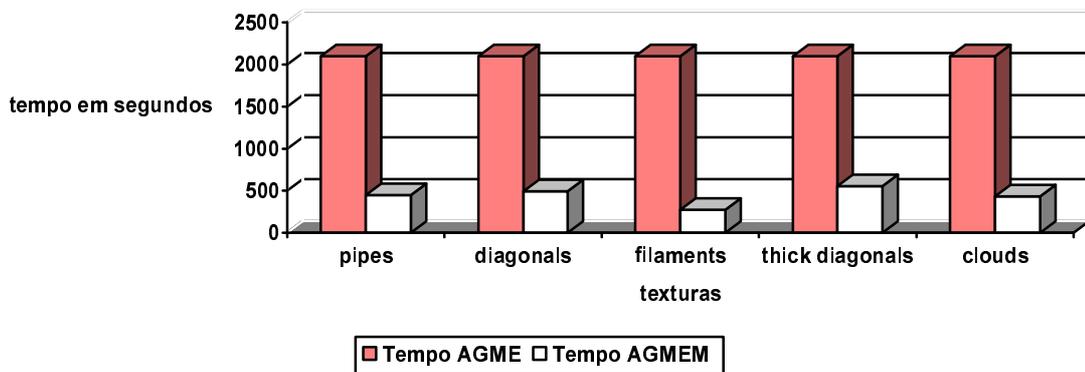


Figura 5.18 – Tempo total de processamento do AGME e do AGMEM para encontrar o *foot pattern* com a menor entropia.

O tempo total de processamento do AGME foi obtido implementando-se esse algoritmo com todas as características descritas pelos autores [11], no mesmo computador e

com a mesma ferramenta de programação (MATLAB 6.5) usados nos testes realizados com o AGMEM.

Pode-se verificar na Tabela 5.4 e na Figura 5.18 que a utilização no algoritmo AGMEM do operador de elitismo e do algoritmo guloso diminuíram bastante o tempo total de processamento. Esse algoritmo sempre após a mutação verifica cada novo indivíduo comparando-o com o seu ascendente direto. No AGME esse tempo médio de processamento em cada geração foi aproximadamente de 10,5 segundos. No AGMEM esse tempo com a implementação do algoritmo guloso e com a introdução do operador de elitismo foi de aproximadamente 15 segundos. Portanto, o aumento do tempo de processamento em cada geração para o AGMEM foi de 42,8%. Porém, como foi visto na Tabela 5.3 e na Figura 5.7 deste capítulo, a redução do número de gerações necessárias para obter um *foot pattern* adequado para cada textura foi muito grande no AGMEM em relação ao AGME.

No AGME, considerando-se o tempo de processamento de cada geração e o número de gerações necessárias para encontrar um *foot pattern* adequado, em todas as cinco amostras de texturas da Figura 5.1 deste capítulo, o tempo total de processamento foi de, aproximadamente, 2100 segundos ou 35 minutos. No AGMEM, para a textura *thick diagonals* que precisou de mais gerações para encontrar o *foot pattern* adequado, o tempo total de processamento foi de aproximadamente 555 segundos ou 9,25 minutos. Portanto, a redução nesse tempo nesse último algoritmo em relação ao AGME foi de 73,57%. Para a textura *filaments*, que mostrou o melhor resultado no AGMEM, a redução desse tempo em relação ao AGME foi de 87,14%.

Ao observar as Tabelas 5.3 e 5.4 e as Figuras 5.17 e 5.18, pode-se verificar que todos os resultados obtidos pelo AGMEM foram bem superiores aos resultados do AGME, devido as seguintes modificações: ao método de ordenação dos indivíduos em ordem decrescente de

aptidão e seleção para a reprodução, seguida da introdução do operador de elitismo e finalmente, após a mutação, pela implementação do algoritmo guloso.

### **5.5.2 - Análise dos resultados do AGMTM**

Como foi visto na Seção 5.4 deste capítulo, as texturas sintetizadas usando o modelo estocástico escolhido pelo AGMTM são bastante semelhantes às amostras de texturas originais, ou seja, possuem o menor erro em relação às amostras fornecidas.

É importante citar que na análise do AGMTM os termos:

- média das diferenças corresponde à média dos quadrados das diferenças entre os histogramas de eventos da amostra original e de cada indivíduo da população, e;
- diferença do melhor indivíduo corresponde ao quadrado da diferença dos histogramas de eventos da amostra original e da amostra da textura sintetizada usando o melhor indivíduo de cada geração.

Existem duas vantagens do AGMTM em relação ao AGMT. A primeira é que nunca aumenta o valor da média das diferenças e o valor da menor diferença a cada geração, obtendo-se, portanto uma evolução contínua da população. A segunda vantagem é a acentuada diminuição, nas primeiras gerações, desses valores resultando em menores valores de diferença. Três modificações foram fundamentais para a obtenção dessas vantagens:

- a primeira modificação que contribuiu foi a ordenação de forma decrescente dos indivíduos mais aptos para os menos aptos seguida do método de seleção antes da aplicação dos operadores de elitismo, de *crossover* e de mutação. No AGMT essa seleção é realizada de forma aleatória dividindo-se a população em grupos de 4 indivíduos, como explicado na sub-seção 4.4.2 do capítulo 4 deste trabalho. A forma de seleção do AGMTM garante que os indivíduos mais aptos passem pelo *crossover* trocando, entre si, informações importantes. Isto não é possível pela forma de seleção aleatória do AGMT;
- a segunda modificação foi a utilização do operador de elitismo antes da aplicação dos operadores de *crossover* e de mutação. Esse operador permite que os melhores indivíduos da geração, ou seja, os indivíduos mais aptos sejam preservados. Cópias desses indivíduos são feitas e são essas cópias que passam para os operadores de *crossover* e de mutação. Após concluída a mutação, os indivíduos da elite são inseridos na população da nova geração. Isso garante que o valor da diferença do melhor indivíduo nunca aumente de uma geração para outra. Se isso ocorresse esse aumento diminuiria sua aptidão, e;
- a terceira modificação é a utilização do algoritmo guloso. Este algoritmo foi introduzido logo após a mutação e sua função é verificar se as modificações feitas pelos operadores genéticos de *crossover* e de mutação melhoraram a aptidão dos indivíduos. Ele faz essa verificação da seguinte maneira: se a aptidão do novo indivíduo é melhor do que a do seu ascendente direto (indivíduo antes de passar por esses operadores), então o novo indivíduo permanece na população. Caso contrário, a

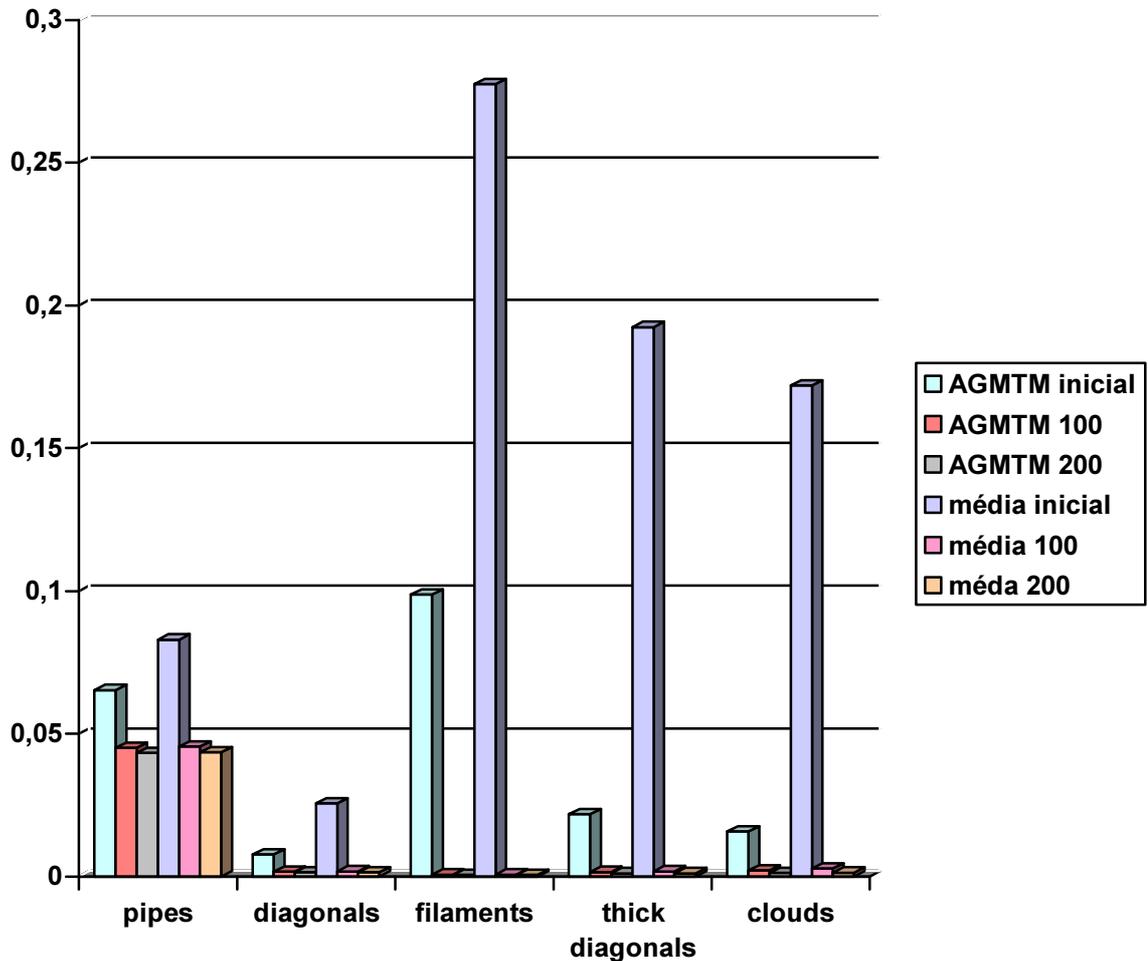
aptidão do novo indivíduo é pior do que a aptidão do seu ascendente direto, então esse novo indivíduo é retirado da população e o seu ascendente direto ocupa o seu lugar na nova geração.

A utilização dessas três modificações fez com que, nas primeiras gerações, o valor da média das diferenças e o valor da menor diferença diminuíssem bastante, já que essas modificações garantem: com a ordenação e seleção uma otimização da reprodução dos indivíduos; com a utilização do operador de elitismo garante o melhor valor encontrado até o momento e assegura, ao utilizar o algoritmo guloso, que nunca ocorrerá um aumento do valor da diferença de qualquer indivíduo. Isso possibilita que o valor da média das diferenças nunca aumente e sua tendência seja sempre diminuir aumentando a aptidão da população como um todo.

A Tabela 5.5 e a Figura 5.19 mostram os resultados obtidos nos testes realizados com o AGMTM, responsável pela escolha do modelo estocástico de textura que consiga representar o mais próximo possível à amostra de textura original fornecida, ou seja, com o menor erro em relação a essa amostra. Essa tabela e essa figura mostram, também, as porcentagens de aperfeiçoamento que ocorrem da população inicial até a geração 100 e da população inicial até a geração 200.

Tabela 5.5 – Resultados obtidos nos testes realizados com o AGMTM.

<b>Textura</b>	<b>Valor</b>	<b>AGMTM Diferença do melhor indivíduo</b>	<b>Aperfeiçoamento obtido do melhor indivíduo em relação à população inicial</b>	<b>AGMTM Média das Diferenças</b>	<b>Aperfeiçoamento obtido da média em relação à população inicial</b>
<i>pipes</i>	Inicial	0,065259	-	0,082990	-
	Geração 100	0,045001	31,04 %	0,045449	45,23 %
	Geração 200	0,043257	33,71 %	0,043515	47,56 %
<i>diagonals</i>	Inicial	0,0076791	-	0,0255300	-
	Geração 100	0,0015944	72,23 %	0,0016981	93,34 %
	Geração 200	0,0013471	82,45 %	0,0014000	94,51 %
<i>filaments</i>	Inicial	0,0987900	-	0,27758000	-
	Geração 100	0,0005824	99,41 %	0,00066289	99,76 %
	Geração 200	0,0005206	99,47 %	0,00052543	99,81 %
<i>thick diagonals</i>	Inicial	0,0218010	-	0,1924500	-
	Geração 100	0,0014772	93,22 %	0,0017342	99,09 %
	Geração 200	0,0010575	95,14 %	0,0010640	99,44 %
<i>clouds</i>	Inicial	0,0157750	-	0,1719200	-
	Geração 100	0,0020890	86,75 %	0,0027174	98,41 %
	Geração 200	0,0011844	92,49 %	0,0012583	99,26 %



*Figura 5.19 – Valor da diferença do melhor indivíduo e da média das diferenças da população inicial até as gerações 100 e 200.*

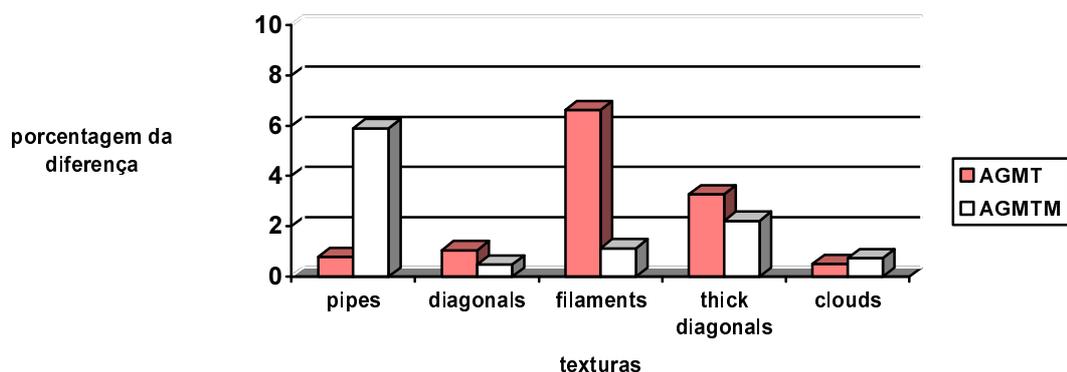
É importante realçar que quanto menor o valor da diferença, tanto da média das diferenças quanto da diferença do melhor indivíduo, maior é a aptidão dos indivíduos.

Pode-se observar na Tabela 5.5 e na Figura 5.19 que para todas as amostras de texturas, mostradas na Figura 5.1, ocorreu uma diminuição de mais de 93% do valor da média das diferenças nas 100 primeiras gerações em relação à diminuição total desse valor ocorrida nas 200 gerações. Isso, também, pode ser evidenciado pela pequena diferença do valor da média das diferenças na geração 100 e na geração 200. Para todas essas amostras mais de

87% da diminuição dos valores da diferença do melhor indivíduo ocorreram nas 100 primeiras gerações.

Na Tabela 5.5 e na Figura 5.19, os resultados obtidos da média das diferenças na geração 200 estão bem próximos das diferenças dos melhores indivíduos nessa geração para cada uma das texturas mostradas na Figura 5.1 deste capítulo. Isso mostra a excelente evolução geral de toda a população. A única textura que ocorreu a redução dos valores da média das diferenças menor do que 94% foi a *pipes*. Essa evolução foi obtida, principalmente, pela utilização da ordenação dos indivíduos em ordem decrescente de aptidão junto com o método de seleção de indivíduos e pelo algoritmo guloso.

A Figura 5.20 e a Tabela 5.6 mostram os erros entre o modelo estocástico encontrado pelo AGMTM e o modelo da imagem sintetizada do AGMT, em relação ao modelo estocástico da amostra da textura da imagem original. Esses erros foram calculados usando a soma das diferenças entre o modelo estocástico da textura da imagem original e o modelo estocástico da textura da imagem sintetizada dividido por 16, que é o máximo valor possível para a soma dos valores do vetor dos modelos de textura.



*Figura 5.20 – Erros entre o modelo estocástico encontrado pelo AGMTM e o modelo da imagem sintetizada do AGMT, em relação ao modelo estocástico da amostra da textura da imagem original.*

Tabela 5.6 – Erros entre o modelo estocástico encontrado pelo AGMTM e o modelo da imagem sintetizada do AGMT, em relação ao modelo estocástico da amostra da textura da imagem original.

Texturas	Erro do modelo estocástico do AGMT em relação ao modelo estocástico da amostra da textura original (%)	Erro do modelo estocástico do AGMTM em relação ao modelo estocástico da amostra da textura original (%)
<i>pipes</i>	0,79	5,90
<i>diagonals</i>	1,05	0,47
<i>filaments</i>	6,63	1,13
<i>thick diagonals</i>	3,27	2,22
<i>clouds</i>	0,50	0,74

Pode-se observar na Tabela 5.6 e na Figura 5.20 que os erros entre os modelos estocásticos encontrados pelo AGMTM e os modelos das amostras das texturas originais fornecidas: *diagonals*, *filaments*, *thick diagonals* e *clouds* são pequenos e menores do que 2,3% e para a textura *pipes* é um pouco maior e igual a 5,9%. Essa tabela e essa figura mostram, também, que os erros encontrados pelo AGMTM foram menores do que os erros do AGMT para as texturas *diagonals*, *filaments* e *thick diagonals*. Para as texturas *clouds* e *pipes* os erros do AGMTM foram maiores do que os erros do AGMT em 0,24% e 5,1%, respectivamente. Essa diferença dos erros dessas duas últimas texturas pode ser devido à aleatoriedade dos algoritmos genéticos utilizados e ao fato dos autores do AGMT não terem disponibilizado o modelo de textura encontrado por esse algoritmo. Os valores dos erros encontrados pelo AGMTM mostram que as imagens sintetizadas pelos modelos encontrados por esse algoritmo estão bem próximas das imagens dos modelos estocásticos das amostras de texturas originais fornecidas.

O tempo total de processamento do AGMT foi obtido implementando esse algoritmo no mesmo computador e usando a mesma ferramenta de programação (MATLAB 6.5) utilizados nos testes realizados no AGMTM. Todas as características descritas pelos autores em [11] foram consideradas na implementação.

A única desvantagem do AGMTM em relação ao algoritmo AGMT foi no tempo de processamento. Como, após a mutação, o algoritmo guloso requer vários cálculos adicionais de aptidão e para cada um desses cálculos no AGMTM tem de ser sintetizada uma amostra de textura para cada indivíduo diferente, o tempo médio de processamento passou de aproximadamente 14,0 segundos no AGMT para 21,5 segundos no AGMTM, resultando em um aumento total de 53,5% no tempo de processamento de cada geração. Porém, nas 100 primeiras gerações ocorreu uma acentuada redução dos valores da média das diferenças e da diferença do melhor indivíduo, atenuando o efeito negativo do aumento nesse tempo. Essa acentuada diminuição, em todas as texturas testadas, significa que foram encontrados modelos estocásticos de textura próximos a essa geração, que conseguiram representar o mais próximo possível às amostras de texturas originais fornecidas, ou seja, com o menor erro em relação a essas amostras.

### **5.5.3 – Tempo total de processamento dos algoritmos AGME e AGMT e dos algoritmos AGMEM e AGMTM**

A Tabela 5.7 e a Figura 5.21 mostram os resultados obtidos do tempo total de processamento dos algoritmos AGMEM e AGMTM e dos algoritmos AGME e AGMT.

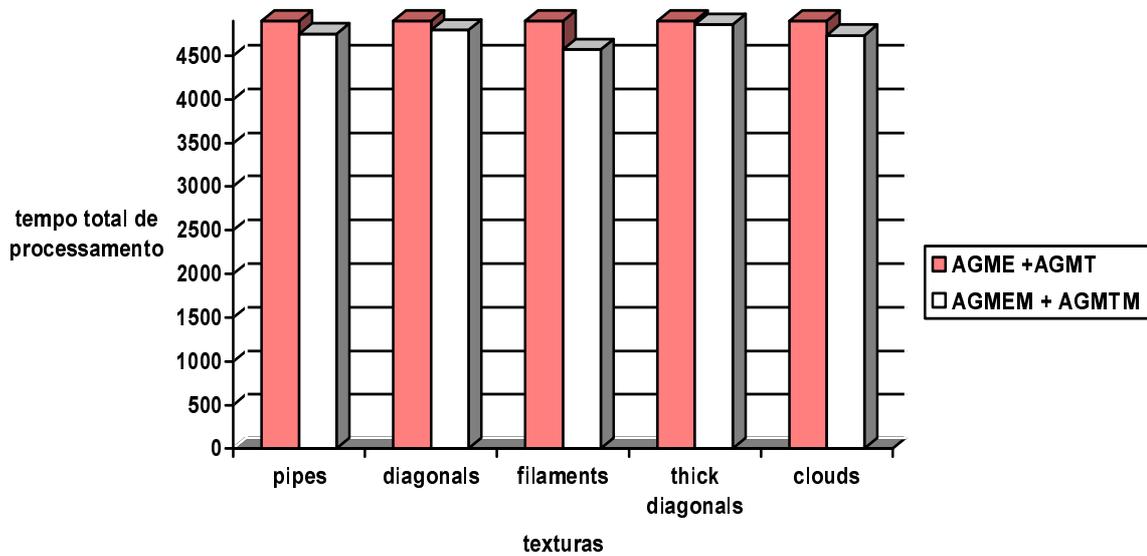


Figura 5.21 – Tempo total de processamento dos algoritmos AGME e AGMT e dos algoritmos AGMEM e AGMTM.

Tabela 5.7 – Comparação do tempo total de processamento entre os algoritmos AGMEM e AGMTM e os algoritmos AGME e AGMT

Tempo total de processamento	<i>pipes</i>	<i>diagonals</i>	<i>filaments</i>	<i>thick diagonals</i>	<i>clouds</i>
AGME + AGMT (seg.)	4900	4900	4900	4900	4900
AGMEM + AGMTM (seg.)	4750	4795	4570	4855	4735
Redução do tempo total (%)	3,06	2,14	6,73	0,92	3,37

Pode-se observar na Tabela 5.7 e no gráfico da Figura 5.21 que apesar do tempo de processamento do AGMTM ser maior do que o tempo de processamento do algoritmo AGMT, o tempo total de processamento dos algoritmos AGMEM e AGMTM é, para todas as amostras de texturas da Figura 5.1 deste capítulo, menor do que o tempo total de

processamento dos algoritmos AGME e AGMT. A diminuição desse tempo é de 0,9% a 6,73%.

## 5.6 – Conclusões

Este capítulo mostrou os resultados obtidos ao utilizar o modelo proposto e implementado descrito no capítulo 4 deste trabalho. Foram mostrados os resultados da entropia dos *foot patterns* obtidos pelo AGMEM e as imagens sintetizadas dos modelos estocásticos de textura encontrados pelo AGMTM.

O desempenho dos algoritmos AGMEM e AGMTM foi medido pelo tempo necessário para se obter baixos valores de entropia no AGMEM e no AGMTM em relação ao quadrado da diferença dos histogramas de eventos, pela evolução da população, pelo erro do modelo estocástico do AGMTM em relação ao modelo estocástico da amostra de textura original. O tempo total de processamento depende do tempo necessário para processar cada geração e do número de gerações necessárias para se conseguir resultados que forneçam modelos estocásticos de texturas que sejam capazes de produzir amostras sintetizadas o mais próximo possível das amostras originais de texturas, ou seja, com o menor erro em relação a essas amostras.

Os *foot patterns* encontrados pelo AGMEM apresentaram valores de entropia menores do que os valores de entropia dos *foot patterns* encontrados pelo AGME.

Nas primeiras gerações, outra vantagem do AGMEM foi a acentuada redução dos valores tanto da média das entropias quanto da menor entropia, o que resultou em um menor número de gerações até encontrar o *foot pattern* com o menor valor de entropia. Enquanto o AGME, para todas as amostras de texturas, trabalha com 200 gerações, o AGMEM trabalha

com no máximo 37 gerações para a textura *thick diagonals*, como mostrado na Tabela 5.3, para encontrar um *foot pattern* com um pequeno valor de entropia e ele é capaz de produzir um bom modelo de textura para todas amostras testadas neste trabalho.

No AGMEM, a eficiente evolução da população inicial pode ser mostrada, principalmente nas gerações iniciais, pela redução dos valores da média da entropia a cada geração. Nesse algoritmo, antes de chegar na geração 100, esses valores coincidem com os valores da menor entropia. Isso indica que existe a tendência de próximo a essa geração todos os indivíduos da população serem iguais.

Como mostrado na Tabela 5.4 e na Figura 5.18 o tempo total de processamento do AGMEM para encontrar o *foot pattern* foi bem menor (mais de 70% menor) do que o tempo total de processamento do AGME para encontrar o *foot pattern*.

Utilizando os modelos estocásticos de textura encontrados pelo AGMTM conseguiu-se obter amostras que estão bem próximas das amostras fornecidas, ou seja, com o menor erro em relação a essas amostras.

No AGMTM a acentuada diminuição dos valores da diferença do melhor indivíduo e da média das diferenças mostra que praticamente 90% das reduções desses valores ocorreram nas 100 primeiras gerações. Isso mostra a eficiência desse algoritmo nas gerações iniciais.

No AGMTM a boa evolução da população inicial pode ser observada pelos gráficos da evolução da média das diferenças, na Tabela 5.5 e no gráfico da Figura 5.19 para todas as texturas testadas neste trabalho. Na geração 200 o valor da média das diferenças está bem próximo do valor da menor diferença.

Outro ponto positivo do AGMEM e do AGMTM, é que os valores da média das entropias e da média das diferenças das populações nunca aumentam, ou seja, sua aptidão

nunca diminui. Ou os valores da média diminuem ou se mantêm no mesmo valor. Isso é um fator que influencia muito na evolução geral da população.

A Tabela 5.6 e a Figura 5.20 mostraram que os erros entre os modelos estocásticos encontrados pelo AGMTM e os modelos das amostras de texturas originais fornecidas: *diagonals*, *filaments*, *thick diagonals* e *clouds* foram pequenos e menores do que 2,3% e para a textura *pipes* foi um pouco maior e igual a 5,9%. Essa tabela e essa figura mostraram, também, que os erros encontrados pelo AGMTM foram menores do que o erro do AGMT para as texturas *diagonals*, *filaments* e *thick diagonals*. Para as texturas *clouds* e *pipes* os erros do AGMTM foram maiores do que os erros do AGMT em 0,24% e 5,1%, respectivamente. Essa diferença dos erros dessas duas últimas texturas pode ser devido à aleatoriedade dos algoritmos genéticos e ao fato dos autores do AGMT não terem disponibilizado o modelo de textura encontrado por esse algoritmo. Os valores dos erros encontrados pelo AGMTM mostram que as imagens sintetizadas pelos modelos encontrados por esse algoritmo estavam bem próximas das imagens dos modelos estocásticos das amostras de texturas originais fornecidas.

A única desvantagem do AGMTM é o aumento do tempo de processamento desse algoritmo em relação ao AGMT, devido, principalmente, ao algoritmo guloso. Nos testes realizados neste trabalho. O aumento desse tempo em relação a esse último algoritmo foi, aproximadamente, 53,5 %.

A Tabela 5.7 e o gráfico da Figura 5.21 mostraram que apesar do tempo de processamento do AGMTM ser maior do que o tempo de processamento do AGMT, o tempo total de processamento dos algoritmos AGMEM e AGMTM é, para todas as amostras de texturas da Figura 5.1 deste capítulo, menor do que o tempo total de processamento dos algoritmos AGME e AGMT. A diminuição desse tempo foi de 0,9% a 6,73%.

As principais características dos algoritmos AGMEM e AGMTM foram: mudança no modelo de representação dos *foot patterns* no AGMEM, ordenação dos indivíduos em forma decrescente de aptidão, o método de seleção dos indivíduos para a reprodução, a utilização do operador de elitismo e do algoritmo guloso. Essas características resultaram em uma melhor evolução da população como um todo. Favoreceram, também, nas primeiras gerações, uma acentuada redução dos valores da média da entropia, da menor entropia, da média das diferenças e da diferença do melhor indivíduo. Essas características propiciaram, também, uma redução no tempo total de processamento para encontrar um modelo estocástico de textura que representasse o mais próximo possível uma amostra da textura original fornecida, ou seja, com o menor erro em relação a essa amostra. Essa redução foi muito grande, principalmente no AGMEM.

# CAPÍTULO VI

## CONCLUSÕES, CONTRIBUIÇÕES DESTE TRABALHO E SUGESTÕES PARA FUTUROS TRABALHOS

### 6.1 – Conclusões

Este trabalho propôs e implementou modificações e adaptações no método de síntese de textura que utiliza *POMMs* e algoritmos genéticos da mínima entropia (AGME) e modelos de textura (AGMT) desenvolvidos por Ashlock e Davidson [11]. O principal objetivo dessa dissertação era obter um método de síntese de textura que a partir de uma amostra de qualquer tipo de textura construir uma nova textura que seja o mais próxima possível e com as mesmas características da amostra da textura original, ou seja, com o menor erro em relação a essa amostra. Outro objetivo desta dissertação era diminuir o tempo total de processamento desses algoritmos. Os algoritmos implementados neste trabalho foram os algoritmos genéticos da mínima entropia modificado (AGMEM) e dos modelos de textura modificado (AGMTM).

Os algoritmos AGMEM e AGMTM apresentaram bons resultados. As texturas sintetizadas possuem uma aparência a mais próxima possível das amostras de texturas originais fornecidas.

O AGMEM responsável por escolher um *foot pattern* adequado para cada amostra original fornecida teve um desempenho muito superior ao AGME. As vantagens desse algoritmo são:

- encontrar *foot patterns* com entropia menor do que os encontrados pelo AGME;

- a população nunca piora sua aptidão de uma geração para outra ou a aptidão melhora ou se mantém a mesma;
- grande evolução da população como um todo mostrada pelos valores da média da entropia. Estes valores decrescem acentuadamente nas primeiras gerações, já na geração 100 os valores dessa média coincidem na maioria dos casos com os valores da menor entropia;
- o acentuado decréscimo dos valores da menor entropia nas primeiras gerações, representou uma diminuição no tempo de processamento do AGMEM em relação ao algoritmo AGME. Devido a esse decréscimo, são necessárias poucas gerações para obter bons *foot patterns*.

O AGMTM é responsável pela escolha do modelo de textura que represente uma textura a mais próxima possível da amostra fornecida utilizando o *foot pattern* escolhido pelo AGMEM. Esse algoritmo mostrou-se bastante eficaz, principalmente, nas gerações iniciais. Suas vantagens em relação ao AGMT são:

- encontrar modelos de textura que produzem amostras sintetizadas de texturas o mais próximo possível das amostras originais de texturas fornecidas, ou seja, com o menor erro em relação a essas amostras;
- a população nunca piora sua aptidão de uma geração para outra ou a aptidão melhora ou se mantém a mesma, e;

- a contínua evolução da população como um todo evidenciada pelos valores da média da diferença que decrescem rapidamente nas primeiras gerações. Na maioria dos casos, já na geração 100 esses valores estão bem próximos dos valores da menor diferença. E na geração 200 esses valores quase coincidem.

A única desvantagem do AGMTM em relação ao AGMT foi o aumento do tempo de processamento para cada geração que foi de aproximadamente 53 %. Isso ocorreu devido aos cálculos adicionais de aptidão realizados no AGMTM pelo algoritmo guloso.

Ao analisar o desempenho dos algoritmos AGMEM e AGMTM em relação aos algoritmos AGME e AGMT, pode-se verificar que ocorreu uma diminuição do tempo total de processamento de 0,9 % a 6,7 % para os cinco tipos de amostras de texturas utilizadas neste trabalho.

## **6.2 – Contribuições deste Trabalho**

Os algoritmos AGMEM e AGMTM propostos e implementados neste trabalho possibilitaram encontrar os modelos de textura que produziram amostras de textura o mais próximo possível da amostra de textura original fornecida, ou seja, com menor erro em relação a essa amostra.

A escolha de *foot patterns* com menores valores de entropia possibilita uma maior confiabilidade na escolha do modelo de textura do AGMTM.

A diminuição no tempo total de processamento dos algoritmos AGMEM e AGMTM em relação aos algoritmos AGME e AGMT foi devido aos métodos de otimização que

reduziram em mais de 70%, no AGMEM, o número de gerações necessárias para alcançar um *foot pattern* adequado com baixa entropia.

### **6.3 – Sugestões para Trabalhos Futuros**

Com o objetivo de otimizar ainda mais o método proposto e implementado neste trabalho podem-se sugerir como futuros trabalhos:

- otimização do cálculo da aptidão do algoritmo AGMTM, com a finalidade de reduzir o tempo de processamento de cada geração ou reduzir o número de gerações necessárias para encontrar um bom modelo estocástico de textura;
- a partir da redução do tempo de processamento, aplicar esse modelo combinado com técnicas de otimização a amostras com maior número de intensidades de níveis de cinza, com objetivo de utilizar texturas mais complexas, e;
- utilização de técnicas avançadas de algoritmos genéticos, tais como evolução conjunta e algoritmo híbridos, com o objetivo de diminuir mais ainda o tempo total de processamento dos dois algoritmos AGMEM e AGMTM.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] – CHEN, C. H., PAU, L. F. and WANG, P. S. P. Texture Analysis In: The Handbook of Pattern Recognition and Computer Vision 2.ed. World Scientific Publishing Co., 1998. p 207-248
- [2] – MEDEIROS, Gilmar Caiado Fleury. Uso de Texturas para Acompanhamento da Evolução do tratamento de Úlceras Dermatológicas. São Carlos: Universidade de São Paulo, Escola de Engenharia de São Carlos, Departamento de Engenharia Elétrica, 2001.
- [3] – ZHU, S. C., WU, Y. e MUMFORD, D. Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling. International Journal of Computer Vision 27(2), p.107-126, 1998
- [4] – FILGUEIRAS, Eduardo Quesado. Síntese de Texturas Usando Modelo Estatísticos Espaciais. São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 1998.
- [5] – GONZALEZ, P. C e WOODS, R. E. Digital Image Processing, Massachusetts: Addison Wesley, 1992
- [6] – BRODATZ, P. Textures: A Photographic Album for Artists and Designers. Dover, New York, 1966.
- [7] – CROSS, G. R. and JAIN, A. K. Markov Random Fields Texture Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, The Institute of Electrical and Electronics Engineers, Inc. vol. PAMI-5, no.1, p. 25-39, Janeiro 1983.

- [8] – CHELLAPPA, R.; KASHYAP, R.L. Synthetic Generation and Estimation in Random Field Models of Images. In: PRIP' 81 IEEE Computer Society Conference on Pattern Recognition and Image Processing, Dallas, 1981. Proceedings. Piscataway: IEEE, 1981, p. 577-582.
- [9] – HARALICK, R.M. Statistical and structural approaches to texture. Proceedings of the IEEE, vol. 67, no. 5, p. 786-804, 1979
- [10] – DAVIDSON, J. L., CRESSIE, N. e HUA, X. Texture Synthesis and Pattern Recognition for Partially Ordered Markov Models. The Journal of the Pattern Recognition Society, Pattern Recognition 32, p. 1475-1505, 1999
- [11] – ASHLOCK, D. e DAVIDSON, J. Texture Synthesis with Tandem Genetic Algorithms using Nonparametric Partially Ordered Markov Models, Proceedings of the 1999 Congress on CEC 99. Washington, vol. 2, p. 1163, julho 1999
- [12] – GOLDBERG, David E. Genetic Algorithms in Search, Optimization, and Machine Learning. 23 ed. Addison Wesley Longman, Inc., 2002.
- [13] – Wikipedia a enciclopédia livre- [www.wikipedia.org](http://www.wikipedia.org) – Acessado em 18/06/2005
- [14] – TOMASSINI, M. A Survey of Genetic Algorithms. Annual Reviews of Computational Physics, vol. 3.
- [15] – HAUPT, Randy L. e HAUPT, Sue Ellen. Practical Genetic Algorithms John Wiley & Sons, 1998
- [16] – HAYKIN, S. Digital Communications, John Wiley/Sons, 1988
- [17] – Homepage de Dan Ashlock - <http://orion.math.iastate.edu/danwell/> Acessado em 01/06/2005

- [18] – ASHLOCK, D. e DAVIDSON, J. Genetic Algorithms for Automated Texture Classification. In SPIE International Symposium on Optical Science, Engineering and Instrumentation, p. 140-151, San Diego, agosto, 1997.
- [19] – BAKER, K. A. e WILLE R., Lattice Theory and Applications. Heldermann, Lemgo, Alemanha, 1995.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)