UNIVERSIDADE DE SÃO PAULO
Escola de Artes Ciências e Humanidades – EACH
Programa de Pós-graduação em Sistemas de Informação - PPgSI

Ricardo Wandré Dias Pedro

Inferência de gramáticas estocásticas para
reconhecimento de padrões em imagens
utilizando quadrees

2013

Ricardo Wandré Dias Pedro

Inferência de gramáticas estocásticas para reconhecimento de padrões em imagens utilizando quadrtrees

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências. Programa: Sistemas de Informação.

Versão corrigida contendo as alterações solicitadas pela comissão julgadora. A versão original encontra-se disponível na Escola de Artes, Ciências e Humanidades.

Orientador(a): Profa. Dra. Arianne Machado Lima

São Paulo
2013

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

CATALOGAÇÃO-NA-PUBLICAÇÃO
Biblioteca

Escola de Artes, Ciências e Humanidades da Universidade de São Paulo

Pedro, Ricardo Wandré Dias

Inferência de gramáticas estocásticas para reconhecimento de padrões em imagens utilizando quadrees / Ricardo Wandré Dias Pedro ; orientadora, Ariane Machado Lima. – São Paulo, 2013.

132 f. : il.

Dissertação (Mestrado em Ciências) - Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo.

Versão corrigida.

1. Processamento de imagens – Técnicas digitais. 2. Linguística computacional. 3. Algoritmos para imagens. 4. Reconhecimento de imagem. 5. Modelos para processos estocásticos. 6. Processamento digital de imagens. I. Lima, Ariane Machado, orient. II. Título.

CDD 22.ed. – 621.367

Dissertação sob o título “*Inferência de gramáticas estocásticas para reconhecimento de padrões em imagens utilizando quadrtrees*”, defendida por Ricardo Wandré Dias Pedro e aprovada em 23 de novembro de 2013, em São Paulo, estado de São Paulo, pela banca examinadora constituída pelos doutores:

Profa. Dra. Arianne Machado Lima
Orientador

Prof. Dr. João José Neto
Universidade de São Paulo

Prof. Dr. Helton Hideraldo Biscaro
Universidade de São Paulo

Dedico este trabalho a toda minha família e a todos meus amigos, pois hoje eu sou um pouco de cada um de deles.

Agradeço a toda minha família (seu Osmar, dona Maricota e Josefa), presente em todos os momentos de minha vida, não só durante o curso de mestrado.

As Profas. Dra. Arianne Machado Lima e Dra. Fátima de Lourdes dos Santos Nunes Marques, pelas orientações, explicações e incansáveis correções dos meus textos.

Aos professores do curso de Bacharelado em Ciências da Computação (ICMC-USP) e do curso de mestrado em Sistemas de Informação (EACH-USP), pelos conhecimentos compartilhados.

Termino agradecendo aos três mosqueteiros, pois sem eles, nada disso teria acontecido.

Por mais longa que seja a caminhada o mais importante é dar o primeiro passo.

(Vinícius de Moraes)

Resumo

PEDRO, Ricardo Wandré Dias. **Inferência de gramáticas estocásticas para reconhecimento de padrões em imagens utilizando quadtrees**. 2013. 132 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2013

Na última década métodos sintáticos vêm sendo bastante empregados para reconhecimento de padrões em imagens. Uma revisão sistemática sobre o assunto indicou que há trabalhos sobre a utilização de gramáticas para reconhecimento de objetos específicos em imagens, reconhecimento de texturas, construção de objetos, segmentação de imagens, mudança de escala de imagens e reconhecimento de layouts de páginas e documentos. Foi percebido que apenas alguns dos trabalhos analisados apresentavam métodos de inferência gramatical. Percebida essa lacuna, este trabalho teve como objetivo avaliar a utilização de gramáticas estocásticas inferidas a partir de amostras de treinamento que continham as estruturas de *quadtrees* das imagens no problema de classificação de imagens. Para atingir tal objetivo foram utilizados e adaptados dois algoritmos de inferência gramatical e um algoritmo de estimação de probabilidades responsável por transformar as gramáticas livres de contexto inferidas em gramáticas livres de contexto estocásticas. Os resultados obtidos mostram que as técnicas propostas podem ser utilizadas na classificação de figuras geométricas. Entretanto, embora o desempenho com *quadtrees* não tenha sido excelente, ficou claro que o uso de gramáticas pode ser uma abordagem interessante para a classificação de imagens.

Palavras-chave: inferência gramatical, gramáticas estocásticas, sintático, linguística, modelos hierárquicos, modelos de composição, modelos reconfiguráveis, imagem, quad-trees

Abstract

PEDRO, Ricardo Wandré Dias. **Inference of stochastic grammars for pattern recognition in images using quadrees**. 2013. 132 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2013.

In the last decade syntactic methods have been widely used for pattern recognition in images. A systematic review on this subject indicated that there are works about the usage of grammars for recognizing specific objects in images, recognition of textures, construction of objects, image segmentation, image scaling, and recognition of layouts in documents. It was noticed that only some of the analyzed studies showed methods of grammatical inference. Perceived this gap, this study aimed to evaluate the usage of stochastic grammars inferred from training samples containing structures of quadrees of images in image classification problem. To achieve this goal it was used and adapted two algorithms for grammatical inference and one algorithm of probability estimation responsible for transforming the inferred context-free grammars in stochastic context-free grammars. The results show that the proposed techniques can be used in the classification of geometric figures. However, although the performance with quadrees has not been excellent, it was clear that the use of grammars can be an interesting approach for image classification.

Keywords: grammatical inference, stochastic grammars, syntactic, linguistic, hierarchical models, models of composition, reconfigurable models, image, quadrees

Lista de Figuras

1	Hierarquia de Chomsky	23
2	Esqueleto de uma frase.	28
3	Um autômato a árvore, sua trie correspondente e a trie com a nova rotulação.	32
4	Árvore que representa a cadeia $((a(bc)d)(a(bc)d))$	34
5	Representação de $t/3$ (esquerda) e de direita $t\backslash 3$ (direita)	35
6	Exemplo de uma quadtree	41
7	Gráfico com a quantidade de artigo publicados ano a ano	47
8	Gráfico com a tendência da utilização de métodos sintáticos	48
9	Gráfico com as principais técnicas utilizadas	49
10	Gráfico com o número de publicações para cada País	49
11	Exemplo de <i>quadtree</i>	50
12	Produções dos símbolos não terminais para subdivisões das regiões retangulares	51
13	Grafo AND-OR que representa uma regra de produção	53
14	Exemplo de grafo AND-OR	54
15	Exemplo de um grafo AND-OR para reconhecimento de relógios e dois possíveis grafos de derivação	55
16	Oriented triplets	56
17	AFs e os triângulos orientados	56
18	Regras de produções utilizadas na gramática para reconhecimento de retângulos	58
19	Resultados de alguns experimentos. Os retângulos reconhecidos são apresentados na última coluna	58
20	Elementos de um símbolo	62

21	Geração de grafos gramaticais utilizados para construção de uma gramática	64
22	Decomposição de um objeto em suas partes	68
23	Figuras geradas pelo sistema SAGS	73
24	Aplicação de <i>L-System</i> para modelagem de plantas e de construções	74
25	Resultados da aplicação do modelo - <i>L-System</i>	75
26	Resultados da aplicação do modelo para simular o desenvolvimento de plantas - <i>L-System</i>	75
27	Exemplos de árvores criados com o modelo e seus tamanhos	76
28	Modelo físico baseado na superfície Genr8	76
29	Resultados da segmentação de imagens	77
30	Sequência de rastreamento de um carro	78
31	Resultado da técnica adaptive image display	79
32	Arquitetura do arcabouço utilizado. O módulo de processamento de imagens, em azul, foi desenvolvido nesse projeto de mestrado.	90
33	Estrutura de <i>quadtree</i> de uma imagem e a string parentisada equivalente.	91
34	Estrutura de uma <i>quadtree</i> que será utilizada para inferência gramatical. .	91
35	Diagrama de classes de streams de entrada.	93
36	UML do relacionamento das classes de streams de saída.	95
37	Exemplo de imagens utilizadas. A primeira linha exibe amostras de quadrados, na segunda pode-se observar alguns círculos, na terceira, triângulos e na quarta, retângulos.	99

Lista de Tabelas

1	Principais trabalhos levantados durante a revisão sistemática	82
2	Síntese dos testes realizados	96
3	Características das sequências utilizadas no teste	104
4	Tamanho médio das gramáticas utilizadas no teste	105
5	Características das sequências utilizadas no teste	106
6	Tamanho médio das gramáticas utilizadas no teste	106
7	Características das sequências utilizadas no teste	109
8	Tamanho médio das gramáticas utilizadas no teste	110
9	Comparação das gramáticas de quadrados geradas pelo IgRT e pelo IgMa- kinen	113
10	Comparação das gramáticas de círculos geradas pelo IgRT e pelo IgMakinen	113
11	Comparação das gramáticas de triângulos geradas pelo IgRT e pelo IgMa- kinen	114
12	Comparação das gramáticas de retângulos geradas pelo IgRT e pelo IgMa- kinen	114
13	Tabela com a síntese de todos os artigos analisados.	119

Lista de Abreviaturas e Siglas

GR Gramática Regular

GLC Gramática Livre de Contexto

GSC Gramática Sensível ao Contexto

GI Gramática Irrestrita

G_s Gramática Estocástica

GLCE Gramática Livre de Contexto Estocástica

EM Expectation-Maximization

HMM Hidden Markov Model

TGR Tree Grammar Reestimator

NW Northwest

NE Northeast

SW Southwest

SE Southeast

ROC Receiver Operating Characteristic

SRT Spatial Random Trees

AF Attributed Feature

MHMM Mixture Hidden Markov Model

ANFIS Adaptive Neuro-Fuzzy Inference System

ARE Augmented Regular Expressions

FSA Finite-State Automata

SAM Surface-to-Air Missile

SVM Support Vector Machine

ETPL(k) Embedding Transformation-preserved Production-ordered K-Left nodes unambiguous

HSM Hierarchical and Syntactic Models

EDT Extended Dominator Tree

NACE n Attaching-Curve Entity

2D Two-dimensional

GF Gramática Fuzzy

3D Three-dimensional

CAD Computer Aided Design 3D

SAGS Splicing Array Grammar Systems

BHA Bidimensional Hierarchical Automata

HTML HyperText Markup Language

POO Programação Orientada a Objetos

Sumário

1	Introdução	18
1.1	Justificativa	19
1.2	Organização deste documento	20
2	Conceitos Fundamentais	21
2.1	Gramáticas	21
2.1.1	A Hierarquia de Chomsky	22
2.1.2	Gramáticas Livres de Contexto	24
2.1.3	Gramáticas Estocásticas	25
2.1.4	Inferência Gramatical	26
2.1.4.1	Inferência de GLCs a partir de exemplos estruturais	27
2.1.4.2	Inferência de gramáticas reversíveis	28
2.1.4.3	Inferência de gramáticas tipo-inversíveis	29
2.1.5	Estimação de parâmetros de GLCEs	32
2.1.5.1	Algoritmos EM	33
2.1.5.2	O algoritmo TGR	34
2.2	Classificação bayesiana	37
2.3	Processamento de Imagens	38
2.3.1	Definições básicas	39
2.3.2	<i>Quadtrees</i>	40
2.4	Gramáticas Aplicadas a Imagens	41

3	Objetivos	43
3.1	Objetivo geral	43
3.2	Objetivos específicos	43
4	Revisão Bibliográfica Sistemática	44
4.1	Planejamento e Condução	44
4.2	Análise Global	46
4.3	Gramáticas de Imagens para Reconhecimento de Padrões	49
4.3.1	Reconhecimento de Objetos	50
4.3.1.1	Gramáticas Estocásticas	50
4.3.1.2	Modelos de Markov e Lógica <i>Fuzzy</i>	60
4.3.1.3	Expressões Regulares Aumentadas	60
4.3.1.4	Lógica de Predicado	61
4.3.1.5	Gramáticas de Adjacência	62
4.3.1.6	Grafos Gramaticais ETPL(k)	62
4.3.1.7	Gramáticas Livres de Contexto	63
4.3.1.8	Grafos EDT	65
4.3.1.9	Gramáticas Regulares	66
4.3.1.10	Gramática Plex Tridimensional	66
4.3.1.11	Abordagem Sintático-Estrutural	69
4.3.1.12	<i>Shock Grammar</i>	69
4.3.2	Reconhecimento de Texturas	71
4.3.3	Construção de Objetos	72
4.3.3.1	Gramática de Atributos	72
4.3.3.2	Gramáticas Matriciais	73
4.3.3.3	<i>L-System</i>	73
4.3.4	Segmentação de Imagens	77

4.3.5	Mudança de Escalas	78
4.3.6	Reconhecimento de <i>Layout</i>	79
4.3.6.1	Gramáticas Regulares	79
4.3.6.2	Gramáticas Livres de Contexto	79
4.3.6.3	Gramáticas Sensíveis ao Contexto	80
4.4	Discussão	80
4.4.1	Vantagens e Aplicações de Gramáticas em Visão Computacional . .	83
4.4.2	Desvantagens da Utilização de Gramáticas em Visão Computacional	85
4.4.3	Breve Comparação com Técnicas Não-Gramaticais	85
4.4.4	Desafios e Possíveis Direções Futuras	86
4.4.5	Conclusões	88
5	Métodos	89
5.1	Arquitetura Básica	89
5.2	Adaptações no arcabouço	93
5.3	Técnicas de avaliação do método proposto	95
6	Resultados e Discussão	101
6.1	Testes utilizando o algoritmo IgRT	101
6.1.1	Teste 1: Quadrados <i>versus</i> Retângulos que respeitam os quadrantes	101
6.1.2	Teste 2: Quadrados <i>versus</i> Círculos, Triângulos e Retângulos que não respeitam os quadrantes - (resolução 1024x1024)	104
6.1.3	Teste 3: Quadrados <i>versus</i> Círculos, Triângulos e Retângulos que não respeitam os quadrantes - (resolução 256x256)	106
6.2	Testes utilizando o algoritmo IgMakinen	107
6.2.1	Teste 4: Quadrados <i>versus</i> Retângulos que respeitam os quadrantes	107
6.2.2	Teste 5: Quadrados <i>versus</i> Círculos, Triângulos e Retângulos que não respeitam os quadrantes - (resolução 256x256)	109

6.3	Comparação entre os algoritmos IgRT e IgMakinen	112
7	Conclusão	115
7.1	Principais Resultados Obtidos	115
7.2	Principais Contribuições	117
7.3	Trabalhos Futuros	118
	Apêndice A Síntese dos artigos analisados	119
	Referências	126

Capítulo 1

Introdução

A teoria das linguagens formais foi desenvolvida em meados de 1950 e tinha como objetivo desenvolver teorias relacionadas a linguagens naturais. Entretanto, não levou muito tempo até que fosse percebido que esta teoria também era importante para estudos de linguagens artificiais, notadamente para linguagens oriundas da computação e da informática [1]. Desde então, o estudo de linguagens formais vem sendo amplamente utilizado em análise sintática de linguagens de programação, modelagem de circuitos lógicos, sistemas biológicos, entre outros. Ainda mais recentemente, é interessante destacar aplicações nas áreas de sistemas de animação, hipertextos e hipermídias, bem como o tratamento de linguagens não lineares, como linguagens espaciais e linguagens n-dimensionais [1].

Esta área do conhecimento preocupa-se com problemas sintáticos das linguagens. Dá-se o nome de sintaxe de uma linguagem à sua representação visual, sua forma e à forma como suas cadeias são estruturadas, sem considerar qualquer informação acerca de seu significado [2]. Atualmente, essa teoria possui construções matemáticas bem definidas e universalmente reconhecidas, por exemplo, as Gramáticas de Chomsky [1].

A compreensão do conteúdo de imagens sempre foi um dos problemas centrais na área de visão computacional [3], sendo que uma das abordagens para tentar resolver este problema é a utilização de métodos sintáticos. Notadamente, na última década, vários trabalhos publicados fizeram uso de gramáticas e, portanto, da teoria de linguagens formais para lidar com problemas de reconhecimento de padrões em imagens. Em especial, esses trabalhos tinham como foco o reconhecimento de objetos específicos em imagens, reconhecimento de texturas, construção de objetos, segmentação de imagens, mudança de escala de imagens e reconhecimento de layouts de páginas e documentos.

Após a conclusão da revisão bibliográfica apresentada no capítulo 4 foi percebido que, embora vários trabalhos fizessem uso de gramáticas para reconhecimento de padrões em imagens, a maioria deles não tratava do problema de inferência gramatical, ou seja, esses trabalhos utilizavam uma gramática previamente criada, geralmente de forma manual.

Percebida esta lacuna, este trabalho teve como foco a utilização de algoritmos de inferência gramatical mediante treinamento. Inicialmente para reconhecimento de formas

geométricas em imagens.

Nesse sentido, o objetivo principal deste trabalho foi avaliar a utilização de gramáticas estocásticas inferidas a partir de amostras de treinamento que continham as estruturas de *quadttrees* das imagens no problema de classificação de imagens.

Para atingir este objetivo foi utilizado e adaptado o arcabouço desenvolvido em [4] que possuía a implementação de algoritmos de inferência gramatical, de estimação de probabilidades e de outras estruturas necessárias para o processo de reconhecimento e classificação das imagens.

1.1 Justificativa

O processo de inferência gramatical para reconhecimento de padrões em imagens se justifica à medida que tal ferramenta pode ser útil para vários profissionais das mais diversas áreas do conhecimento. Como exemplos podemos citar profissionais da medicina, da botânica ou da microbiologia, pois esses profissionais fazem amplo uso de imagens para apoiar o seus trabalhos.

Como estes profissionais tendem a não ter conhecimento sobre linguagens formais, pois esta é uma área muito relacionada à computação, o trabalho de escrever uma gramática que fosse capaz de representar, por exemplo, imagens de tumores, diferentes folhas de árvores, géis proteicos etc. seria bastante complexo.

Por outro lado, profissionais da área de computação também podem não ter conhecimentos suficientes para identificar quais são os padrões que devem ser observados e reconhecidos em imagens de outras áreas de conhecimento, o que faz com que métodos de inferência gramatical de imagens sejam úteis para favorecer trabalhos interdisciplinares similares aos citados.

A principal vantagem de se realizar inferência gramatical é que se trata de um processo de descoberta de conhecimento. Desta forma, é possível descobrir padrões que ainda não haviam sido identificados, sendo que este tipo de conhecimento pode, por exemplo, ser empregado no auxílio de diagnósticos médicos.

1.2 Organização deste documento

Este documento é dividido em mais 7 capítulos. O capítulo 2 faz uma introdução sobre os conceitos mais importantes na área de linguagens formais, processamento de imagens e as principais diferenças entre linguagens de *strings* e linguagens de imagens. No capítulo 3 são detalhados os objetivos deste projeto. No capítulo 4 é apresentada a revisão bibliográfica realizada, na qual são exibidas as principais técnicas utilizadas para reconhecimento de padrões em imagens. O capítulo 5 traz a metodologia utilizada neste projeto. No capítulo 6 são apresentados e discutidos os resultados obtidos. O último capítulo apresenta as conclusões e os trabalhos futuros.

Capítulo 2

Conceitos Fundamentais

Neste capítulo são apresentados os conceitos fundamentais relacionados a linguagens formais e processamento de imagens. São abordados na seção 2.1 os conceitos de gramáticas de *strings*. São apresentados na seção 2.2 os conceitos sobre classificação bayesiana. Na seção 2.3 são apresentados os principais conceitos de processamento de imagens. Por último, na seção 2.4 são levantadas as principais diferenças entre gramáticas de *strings* e gramáticas de imagens.

2.1 Gramáticas

Métodos sintáticos podem ser utilizados para descrever vários tipos de objetos como linguagens de programação, sentenças em vários idiomas, plantas, construções, etc. Nesta seção serão apresentados os principais conceitos de linguagens formais que serão utilizados neste trabalho.

As definições aqui apresentadas foram extraídas de [5], [6], [7] e [2].

Um **alfabeto** ou um **vocabulário** Σ é qualquer conjunto finito de símbolos que podem ser letras ou dígitos, por exemplo. Uma **cadeia** ou **string** é uma sequência de símbolos concatenados pertencentes ao alfabeto. Denotamos a **cadeia vazia** por ε .

Definição 1: Uma **linguagem** sobre um alfabeto Σ é um conjunto de cadeias sobre este alfabeto.

Definição 2: Uma gramática é uma quádrupla $G = (V_N, V_T, R, S)$ na qual V_N é o conjunto de símbolos não terminais, V_T é o conjunto de símbolos terminais, R representa as regras de produção e S é o símbolo inicial pertencente a V_N .

O conjunto de símbolos não terminais, V_N , auxilia na elaboração das regras sintáticas. As produções R representam as regras sintáticas. O conjunto de símbolos terminais, V_T , corresponde ao alfabeto da linguagem a ser descrita.

Exemplo: A gramática a seguir descreve a linguagem $\{a^n u^m \mid n > 0 \text{ e } m > 0\}$. $G = (V_N, V_T, R, S)$, onde:

$$V_N = \{A, B\}$$

$$V_T = \{a, u\}$$

$$S = A$$

$$R = \{A \rightarrow aA, A \rightarrow B, B \rightarrow uB, B \rightarrow u\}$$

Uma gramática é formada por um alfabeto e uma coleção de regras que manipulam este alfabeto. Por exemplo, o conjunto $\{au, aa u, aaa u, aaaa u\}$ descreve um conjunto de sequências formado pelo alfabeto $\{a, u\}$ no qual os primeiros n elementos são compostos pelo caractere a e os m seguintes elementos são compostos pelo caractere u . Basicamente, uma gramática é capaz de gerar combinações de símbolos pertencentes ao alfabeto formando padrões maiores e mais complexos por meio da aplicação de suas regras de produção [8].

Definição 3: Dadas duas cadeias c_1 e c_2 , c_2 é uma **derivação direta** de c_1 , denotada por $c_1 \Rightarrow c_2$ se e somente se existem quatro cadeias f_1, f_2, α e β , tais que:

- $c_1 = f_1 \alpha f_2$
- $c_2 = f_1 \beta f_2$
- $\alpha \rightarrow \beta \in R$

Definição 4: Uma cadeia c_2 é uma **derivação de uma cadeia** c_1 , denotada por $c_1 \Rightarrow^* c_2$, se e somente se existem cadeias $c_{11}, c_{12}, \dots, c_{1n}$, onde $c_1 \Rightarrow c_{11} \Rightarrow c_{12} \Rightarrow \dots \Rightarrow c_{1n} = c_2$, $n \geq 1$.

Definição 5: Uma cadeia c_1 é uma **sentença** de uma gramática $G = (V_N, V_T, R, S)$ se e somente se $c_1 \in \{c \in V_T^* \mid S \Rightarrow^* c\}$, ou seja, uma sentença é uma cadeia composta de símbolos terminais que seja derivação do símbolo inicial.

Definição 6: A **linguagem** gerada por uma gramática G , denotada por $L(G)$, é o conjunto de todas as sentenças de G .

Definição 7: Duas gramáticas são **equivalentes** se e somente se elas geram a mesma linguagem.

2.1.1 A Hierarquia de Chomsky

Chomsky descreveu, em 1959, quatro tipos de restrições nas regras das gramáticas que fazem com que uma dada gramática pertença a uma das quatro classes gramaticais

descritas por ele. Essas classes compõem uma hierarquia conhecida como hierarquia de Chomsky [1].

A Figura 1 exibe a hierarquia de Chomsky, na qual é possível observar a relação das linguagens geradas por esses tipos de gramáticas.

Figura 1 – Hierarquia de Chomsky

Definição 8: Uma **gramática regular** (GR) é uma gramática *linear à direita* ou *linear à esquerda*.

G é uma *gramática regular linear à direita* se suas produções são da seguinte forma:

- $A \rightarrow b, A \in V_N, b \in V_T \cup \{\varepsilon\}$
- $A \rightarrow bC, A \in V_N, b \in V_T, C \in V_N$

G é uma *gramática regular linear à esquerda* se suas produções são da seguinte forma:

- $A \rightarrow b, A \in V_N, b \in V_T \cup \{\varepsilon\}$
- $A \rightarrow Cb, A \in V_N, b \in V_T, C \in V_N$

Dois exemplos clássicos de linguagens L que as gramáticas regulares não podem descrever são:

1. L contém apenas *strings* da forma $aa, bb, abba, baab, abaaba$ etc; ou seja, gramáticas regulares não podem gerar *strings* palíndromos¹ (a e $b \in V_T$).
2. L contém todas as *strings* da forma $aa, abab, aabaab$; ou seja, gramáticas regulares não podem gerar *strings* que possuam duas partes idênticas (a e $b \in V_T$).

¹Um palíndromo é uma palavra, frase ou qualquer outra sequência de unidades que tenha a propriedade de poder ser lida tanto da direita para a esquerda como da esquerda para a direita.

Definição 9: Uma gramática G é uma **gramática livre de contexto** (GLC) se suas produções são da forma:

- $A \rightarrow \beta, A \in V_N, \beta \in (V_T \cup V_N)^*$

Essas gramáticas são capazes de gerar *strings* palíndromos, mas da mesma forma que as gramáticas regulares não conseguem gerar *strings* que possuam duas partes idênticas. Um exemplo de regras de uma gramática livre de contexto que poderia ser utilizada para gerar *strings* palíndromos é exibido a seguir: $S \rightarrow aSa \mid bSb \mid aa \mid bb$, com a e $b \in V_T$.

Definição 10: G é uma **gramática sensível ao contexto** (GSC) se suas produções são da seguinte forma:

- $\alpha \rightarrow \beta, \alpha \in (V_N \cup V_T)^* V_N (V_T \cup V_N)^*, \beta \in (V_T \cup V_N)^*, |\alpha| \leq |\beta|$, no qual $|\alpha|$ denota o número de símbolos da cadeia α .

Gramáticas sensíveis ao contexto são capazes de gerar *strings* com duas partes idênticas. Exemplo: $G = (V_N, V_T, R, S)$, no qual:

$$V_N = \{S, A, B, T\}$$

$$V_T = \{a, b\}$$

$$S = S$$

$$R = \{S \rightarrow aAS \mid bBS \mid T, Aa \rightarrow aA, Ba \rightarrow aB, Ab \rightarrow bA, Bb \rightarrow bB, BT \rightarrow Tb, AT \rightarrow Ta, T \rightarrow \varepsilon\}$$

Definição 11: G é uma **gramática irrestrita** (GI) se suas regras de produção são da forma:

- $\alpha \rightarrow \beta, \alpha \in (V_N \cup V_T)^* V_N (V_T \cup V_N)^*, \beta \in (V_N \cup V_T)^*$.

Em outras palavras, uma gramática irrestrita permite que o lado esquerdo e o lado direito de uma regra de produção possuam quaisquer combinações de símbolos.

2.1.2 Gramáticas Livres de Contexto

A derivação de uma cadeia segundo sua gramática pode ser descrita por meio de uma *árvore sintática* ou *árvore de derivação*.

Definição 12: Dada uma cadeia c gerada por uma gramática livre de contexto $G = (V_N, V_T, R, S)$, diz-se que uma árvore é uma **árvore sintática** ou **árvore de derivação** de c para a gramática G se:

- cada nó da árvore possui um rótulo pertencente a $V_N \cup V_T \cup \{\varepsilon\}$;
- o rótulo do nó raiz é o símbolo inicial de G ;
- se um determinado nó n , cujo rótulo é A , possui filhos $n_1, n_2, \dots, n_k$ na ordem da esquerda para a direita e seus rótulos são $X_1, X_2, \dots, X_k$, respectivamente, então A pertence ao conjunto de nós não terminais e a produção $A \rightarrow X_1 X_2 \dots X_k$ pertence ao conjunto de produções da gramática.
- se um nó possui o rótulo ε , então esse nó deve ser simultaneamente uma folha e descendente único de seu ancestral direto.

Definição 13: Dizemos que uma gramática G é **ambígua** se existe uma cadeia $c \in L(G)$ que possui duas ou mais árvores sintáticas distintas para a gramática G .

Definição 14: Um **analisador sintático** para uma gramática G é um programa que, dada uma cadeia, é capaz de fornecer pelo menos uma árvore sintática para a cadeia se ela pertencer à $L(G)$, ou uma mensagem de erro, caso contrário.

Se uma gramática for ambígua, uma cadeia c pode possuir mais de uma árvore sintática. Neste caso, um analisador sintático pode fornecer todas as árvores de derivações existentes ou apenas uma delas, dependendo do algoritmo.

2.1.3 Gramáticas Estocásticas

Qualquer gramática da hierarquia de Chomsky pode ser utilizada de forma estocástica. Um modelo de gramática estocástica θ gera diferentes *strings* w com probabilidade $P(w|\theta)$. Em uma gramática estocástica, a soma das probabilidades de todas as produções contendo um mesmo lado esquerdo é 1. A seguir seguem algumas definições formais.

Definição 15: Uma **gramática estocástica** é uma quintupla $G_s = (V_N, V_T, R, S, P)$, onde:

- V_N, V_T, S , e R possuem o mesmo significado que em gramáticas não estocásticas descrito na seção 2.1;

- P é uma função $P:R \rightarrow [0,1]$, ou seja, é uma função que associa a cada produção uma probabilidade p ;
- Considerando as várias regras de produção com o mesmo lado esquerdo α , $\alpha \rightarrow \beta_1|\beta_2|\dots|\beta_{n(\alpha)}$, cada regra $\alpha \rightarrow \beta_i$ é associada a uma probabilidade $p(\alpha \rightarrow \beta_i)$, tal que $\sum_{i=1}^{n(\alpha)} p(\alpha \rightarrow \beta_i) = 1$.

Definição 16: Um **analisador sintático para uma gramática estocástica** G_s , é um analisador em que dada uma cadeia c , se esta cadeia pertencer a $L(G_s)$, não apenas fornece a árvore de derivação de c , mas também fornece a probabilidade de c , segundo sua gramática. Caso contrário, fornece uma mensagem de erro.

A probabilidade de uma árvore de derivação t de uma cadeia c pertencente à linguagem gerada por uma gramática G_s é o produto das produções utilizadas em t .

A probabilidade de uma cadeia c pertencer à linguagem gerada por uma gramática G_s , é o somatório das probabilidades de todas as árvores sintáticas t_i que podem ser derivadas a partir de c e de G_s .

2.1.4 Inferência Gramatical

Inferir uma gramática que representa um padrão de uma determinada classe a partir de um conjunto de exemplos desta classe é um dos problemas encontrados em aprendizado de máquina. A inferência gramatical tem como objetivo aprender o conjunto de regras de produção que farão parte de uma gramática a partir de um conjunto de dados fornecido. Inferência gramatical tem sido um difícil problema quando aplicado a situações práticas, pois o número de exemplos é normalmente finito, de tal forma que a generalização é um importante aspecto a ser considerado pelos algoritmos de aprendizagem. Outro tipo de inferência a ser considerado é o utilizado para estimar parâmetros de uma gramática estocástica. Neste caso, para uma dada gramática espera-se estimar os parâmetros da distribuição de probabilidade que será aplicada às regras de produção existentes.

Uma amostra positiva é uma amostra que contém apenas elementos positivos, ou seja, todos os elementos pertencentes à amostra em questão são exemplares da classe que está sendo analisada. Em contrapartida, em uma amostra negativa todos os elementos não pertencem à classe que está sendo analisada.

Considerando um problema de inferência a ser resolvido e um algoritmo que resolva este problema, o critério de parada é responsável por definir quando o algoritmo deve

terminar sua execução devido ao fato de ter encontrado uma solução para o problema. É introduzido em [9] o critério de parada chamado *identificação no limite*. Uma classe de gramática Υ é identificável no limite se existe um algoritmo de aprendizado A tal que para toda gramática $G \in \Upsilon$ e para cada amostra ϕ de $L(G)$, existe um número n tal que para todo $m \geq n$, $(A(\phi_m)) = L(G)$ e $A(\phi_m) = A(\phi_n)$. O problema de inferência das regras de uma gramática torna-se mais difícil à medida que subimos na hierarquia de Chomsky. No caso particular de gramáticas livres de contexto, essas não podem ser identificadas no limite em tempo polinomial a partir simplesmente de amostras positivas [10]. Porém, uma das alternativas é realizar a inferência a partir de exemplos estruturais [11]. Pode-se ainda é realizar a inferência para subclasses específicas de gramáticas livres de contexto como, por exemplo, gramáticas lineares [12], gramáticas lineares simétricas [13], gramáticas reversíveis [14] e gramáticas tipo-inversíveis [15].

Este trabalho fez uso de dois inferidores de gramaticais. O primeiro inferidor gera gramáticas *reversíveis* enquanto que o segundo inferidor é responsável por gerar gramáticas *tipo-inversíveis*. A seguir são explicados os algoritmos utilizados por cada um desses geradores gramaticais.

2.1.4.1 Inferência de GLCs a partir de exemplos estruturais

Uma maneira de caracterizar sentenças a partir de suas descrições estruturais foi apresentada em [11]. Essas estruturas, no caso de GLCs, seriam as árvores de derivação sem que os nós terminais ou internos fossem rotulados. Os autores sugeriram que é possível desenvolver algoritmos de inferência eficientes quando se possui as informações estruturais das amostras. Em [14] é mostrado que é possível identificar no limite em tempo polinomial a classe das gramáticas livres de contexto utilizando exemplos positivos estruturais.

Se considerarmos a gramática da língua portuguesa, a cadeia “A pasta do menino continha poucos documentos importantes” pode ser analisada utilizando itens sintáticos (sujeito, núcleo do sujeito, adjunto adnominal, predicado, etc.). É possível observar na Figura 2 o esqueleto da árvore de derivação dessa cadeia.

O exemplo estrutural que representa o esqueleto da Figura2 é a seguinte cadeia com parênteses: ((A (pasta (do menino))) (continha (poucos (documentos importantes)))).

Figura 2 – Esqueleto de uma frase.

2.1.4.2 Inferência de gramáticas reversíveis

Sakakibara (1992) propôs um algoritmo para inferir gramáticas reversíveis a partir de exemplos positivos estruturais.

Definição 17: Uma gramática livre de contexto $G = (V_N, V_T, R, S)$ é *inversível* se e somente se $A \rightarrow \alpha$ e $B \rightarrow \alpha \in R$ implica em $A = B$.

Definição 18: Uma gramática livre de contexto $G = (V_N, V_T, R, S)$ é *reset-free* se e somente se para cada dois símbolos não-terminais B, C e $\alpha, \beta \in (V_N \cup V_T)^*$, $A \rightarrow \alpha B \beta$ e $A \rightarrow \alpha C \beta \in R$ implica $B = C$.

Definição 19: Uma gramática livre de contexto G é dita ser *reversível* se e somente se G é inversível e *reset-free*. Uma linguagem livre de contexto L é definida como *reversível* se e somente se existe uma gramática livre de contexto reversível G tal que $L = L(G)$.

Apesar dessas restrições, essa classe de gramáticas pode gerar toda a classe de linguagens livres de contexto [10].

O algoritmo descrito em [14] utiliza GLCs reversíveis e exemplos estruturais para inferir autômatos a árvore reversíveis. A tática de utilizar autômatos pode ser empregada uma vez que toda gramática livre de contexto é equivalente a um *autômato a árvore* [11].

Definição 20: Um autômato a árvore é uma quádrupla $A = (Q, V, \delta, F)$, no qual Q representa o conjunto de estados, V é o alfabeto de terminais, δ é uma função $\delta: (Q \cup V)^* \rightarrow Q$ utilizada para associar um conjunto de estados e terminais a um estado pai, e F é o conjunto de estados finais do autômato.

Em um autômato a árvore os nós internos são os *estados* do autômato e as folhas são os *símbolos* do alfabeto. A raiz da árvore é denominada *estado final*. O conjunto de estados é particionado em blocos. Dados dois blocos b_1 e b_2 , agrupá-los significa unir em um único bloco b_3 os estados pertencentes a b_1 e b_2 , substituindo b_1 e b_2 por b_3 na partição.

Cada sequência utilizada no treinamento será transformada em um autômato a árvore, e agrupar todos eles em um só gera um autômato que reconhece extamante as sequências que geraram o automato maior. A união de blocos irá generalizar o modelo, permitindo que outras sequências sejam reconhecidas. Unindo todos os estados finais de um autômato em um único bloco, esse autômato torna-se equivalente a uma gramática. Quando isso ocorre, temos que o bloco contendo os estados finais equivale ao símbolo inicial, os blocos equivalem aos símbolos não terminais e os símbolos do alfabeto são equivalentes aos símbolos terminais. Ainda, a relação de pai-descendentes no autômato é equivalente a uma produção. Nesse sentido, se o estado e_1 possui os estados e_2 , e_3 e o símbolo a como descendentes, sendo que e_1 , e_2 pertencem ao bloco b_1 e e_3 pertencente ao bloco b_3 , temos a produção²: $\langle b_1 \rangle \rightarrow \langle b_1 \rangle \langle b_3 \rangle a$.

No algoritmo proposto em [14], inicialmente é gerado um autômato a árvore para cada uma das sequências presentes na amostra e para cada autômato é criada uma partição de tal forma que seja criado um bloco para cada estado. Em seguida, os blocos finais de todos os autômatos são agrupados de maneira a formar um único autômato a árvore capaz de reconhecer todas as sequências de entrada. Então, o algoritmo realiza vários testes e agrupamentos sucessivos para que a propriedade de reversibilidade da gramática seja assegurada. Ao término da execução do algoritmo, obtém-se um autômato a a árvore equivalente a uma gramática reversível que é dado como resposta do processo de inferência. A seguir é exibida uma descrição do algoritmo, no qual a entrada é uma amostra positiva não vazia S e a saída é um autômato reversível A . Este algoritmo possui complexidade $O(n^3)$, sendo n o número de parênteses sobre as sequências da amostra de treinamento, o que corresponde ao número de estados do autômato a árvore inicial.

2.1.4.3 Inferência de gramáticas tipo-inversíveis

Makinen (1992) propôs um algoritmo para inferir gramáticas tipo-inversíveis a partir de exemplos positivos estruturais. Assim como as gramáticas reversíveis, as gramáticas

²Os sinais $\langle$ e $\rangle$ serão utilizados para delimitar os símbolos não-terminais quando estes são compostos de mais de um caracter

Entrada: uma amostra positiva não vazia S
Saída: Um autômato reversível A

- 1 Crie o autômato a árvore $A = (Q, V, \delta, F)$ formado a partir da amostra S
- 2 Crie uma partição trivial π_0 sobre Q
- 3 $q \leftarrow$ algum estado pertencente a F
- 4 $lista \leftarrow$ todos os pares (q, q') onde $q' \in F - \{q\}$
- 5 $i \leftarrow 0$
- 6 **enquanto** $(lista \neq \emptyset)$ **faça**
 - 7 remova o primeiro par (q_1, q_2) de $lista$
 - 8 $\%B(q, \pi_i)$ retorna o bloco do estado q dentro da partição π_i
 - 9 $B_1 \leftarrow B(q_1, \pi_0)$ e $B_2 \leftarrow B(q_2, \pi_0)$
 - 10 **se** $B_1 \neq B_2$ **então**
 - 11 $\pi_{i+1} \leftarrow \pi_i$ com B_1 e B_2 agrupados
 - 12 $\text{VerificaPropriedadeInversivel}(\pi_{i+1})$
 - 13 $\text{VerificaPropriedadeResetFree}(\pi_{i+1}, B_1, B_2)$
 - 14 $i \leftarrow i + 1$
- 15 **VerificaPropriedadeInversivel** (π_{i+1}) :
- 16 **para cada** $par\ de\ estado\ (q, q') \in Q \times Q$ **fazer**
 - 17 **se** q e q' têm o mesmo número de descendentes **então**
 - 18 $q = \text{pai}(u_1, u_2, \dots, u_k)$ e $q' = \text{pai}(u'_1, u'_2, \dots, u'_k)$
 - 19 **se** $((u_j, u'_j \in Q \text{ e } B(u_j, \pi_{i+1}) = B(u'_j, \pi_{i+1})) \text{ ou } u_j = u'_j \in V)$ para $1 \leq j \leq k$ e $B(q, \pi_{i+1}) \neq B(q', \pi_{i+1})$ **então**
 - 20 $\quad \quad \quad$ adicione o par (q, q') à $lista$
- 21 **VerificaPropriedadeResetFree** (π_{i+1}, B_1, B_2) :
- 22 **para cada** $(q, q') \in B_1 \times B_2$ **fazer**
 - 23 **se** q e q' têm o mesmo número de descendentes **então**
 - 24 $q = \text{pai}(u_1, u_2, \dots, u_k)$ e $q' = \text{pai}(u'_1, u'_2, \dots, u'_k)$
 - 25 **se** $(u_l, u'_l \in Q \text{ e } B(u_l, \pi_{i+1}) \neq B(u'_l, \pi_{i+1}))$ para algum l ($1 \leq l \leq k$) e $B(u_j, \pi_{i+1}) = B(u'_j, \pi_{i+1})$ ou $u_j = u'_j \in V$, para $1 \leq j \leq k, j \neq l$ **então**
 - 26 $\quad \quad \quad$ adicione o par (u_l, u'_l) à $lista$

Algoritmo 1: Algoritmo IgRT

tipo-inversíveis podem gerar toda a classe de linguagens livres de contexto.

Antes de definir as gramáticas tipo-inversíveis são necessárias algumas definições:

- Seja uma função $h: (V_N \cup V_T)^* \rightarrow \{n, t\}^*$ onde:

Se $x \in (V_N \cup V_T)$,

$h(x) = n$ se $x \in V_N$ e $h(x) = t$ se $x \in V_T$

Se $x \in (V_N \cup V_T)^* \mid x = x_1x_2\dots x_k, k > 1$ e $x_i \in (V_N \cup V_T), 1 \leq i \leq k$

$$h(x) = h(x_1)h(x_2)...h(x_k)$$

ou seja, $h(x_1), h(x_2) \dots h(x_k)$ concatenados

- Dizemos que uma produção $A \rightarrow \alpha$ é terminal se e somente $\alpha \in V_T^*$. Caso contrário ela é dita não terminal.
- Seja $R(G) = \{\alpha \mid \alpha \in (V_N \cup V_T)^*, A \rightarrow \alpha \text{ é uma produção não terminal da gramática } G\}$. G é *R-separável* se e somente se para cada α_1 e $\alpha_2 \in R(G)$ tal que $A \rightarrow \alpha_1$ e $B \rightarrow \alpha_2$, se $h(\alpha_1) = h(\alpha_2)$ então $A = B$.
- Um conjunto de produções é inversível se $A \rightarrow \alpha$ e $B \rightarrow \alpha$, então $A = B$.

Uma gramática G é dita tipo-inversível se ela for *R-separável* e se seu conjunto de produções terminais for inversível.

O algoritmo de inferência de gramáticas tipo-inversíveis além de fazer uso de autômatos a árvore também utiliza uma estrutura chamada *trie*.

Definição 21: Uma *trie* é uma dupla $T = (N, E)$ na qual:

- N é um conjunto finito de nós;
- E é um conjunto finito de arestas;
- uma aresta $e \in E$ é uma tripla (n_1, n_2, r) conectando os nós n_1 e n_2 , com $n_1, n_2 \in N$ e possui o rótulo r .

Foi utilizado em [15] uma *trie* na qual cada nó pode possuir mais de um objeto associado. Essa trie é construída para corresponder ao autômato a árvore criado utilizando as sequências de entrada. Os estados do autômato são os objetos associados aos nós da trie e os rótulos das arestas rotulam os estados e os terminais do autômato a árvore. Essa associação ocorre da seguinte forma: dado um estado s do autômato A que possui descendentes $d_1, d_2, \dots, d_k$, há um caminho p na trie T indo da raiz a um nó n em que:

- o estado s está associado ao nó n ;
- o caminho p é formado por k arestas rotuladas $d_1, d_2, \dots, d_k$, nesta ordem, da raiz ao nó n .

Isso faz com que um estado s esteja associado a um nó n da trie quando a sequência dos rótulos dos descendentes do estado s corresponder à sequência dos rótulos das arestas contidas no caminho que vai do nó raiz até o nó n .

Entretanto, no algoritmo implementado as arestas são rotuladas de forma diferente. Os rótulos dos estados são trocados pelo rótulo n e os rótulos dos terminais são trocados pelo rótulo t mediante utilização de homomorfismo h descrito anteriormente nesta seção. Não podem partir, de cada nó, duas arestas com o mesmo rótulo e quando isso acontece, estas devem ser agrupadas, bastando apenas agrupar os nós destinos das arestas. Podemos observar na Figura 3 o processo de criação de uma trie dado um autômato e sua nova rotulação.

Figura 3 – Um autômato a árvore, sua trie correspondente e a trie com a nova rotulação.

O algoritmo inicialmente constrói o autômato a árvore a partir das sequências de entrada e, em seguida, a partir do autômato, cria a *trie* que será utilizada. Esse processo ocorre da mesma forma que no algoritmo de Sakakibara, são definidas algumas propriedades sobre a trie de forma a torná-la correspondente a um autômato a árvore tipo-inversível. Essas propriedades são as de inversibilidade de produções terminais e de R-separabilidade. O algoritmo tem complexidade $O(n)$ no qual n é o número de parentisações contidas nas amostras de treinamento. Podemos observar a seguir uma descrição do algoritmo.

2.1.5 Estimação de parâmetros de GLCEs

O processo de inferir gramáticas livres de contexto estocásticas geralmente ocorre em duas fases. A primeira fase é composta pelo aprendizado das regras da gramática não estocástica, enquanto que na segunda fase ocorre a estimação das probabilidades de suas regras de produção.

Entrada: Uma amostra positiva não vazia S
Saída: Um autômato reversível A

- 1 Crie o autômato a árvore $A = (Q, V, \delta, F)$ formado a partir da amostra S
- 2 Crie uma partição trivial π_0 sobre Q
- 3 $q \leftarrow$ algum estado pertencente a F
- 4 $lista \leftarrow$ todos os pares (q, q') onde $q' \in F - \{q\}$
- 5 Crie uma trie t equivalente ao autômato a árvore A
- 6 %% Verifica propriedade de R-Separável
- 7 **para cada nó n de t fazer**
- 8 **se há dois ou mais estados associados ao nó n então**
- 9 insira pares desses estados em $lista$
- 10 $i \leftarrow 0$
- 11 **enquanto $lista \neq \emptyset$ faça**
- 12 remova o primeiro par (q_1, q_2) de $lista$
- 13 $B_1 \leftarrow B(q_1, \pi_0)$ e $B_2 \leftarrow B(q_2, \pi_0)$
- 14 **se $B_1 \neq B_2$ então**
- 15 $\pi_{i+1} \leftarrow \pi_i$ com B_1 e B_2 agrupados
- 16 $i \leftarrow i + 1$

Algoritmo 2: Algoritmo IgMakinen

Uma das formas de se realizar essa estimação é utilizando algoritmos EM (*Expectation-Maximization*).

2.1.5.1 Algoritmos EM

Algoritmos EM são algoritmos iterativos que alternam passos para cálculos de expectativa e de maximização, sendo muito úteis para resolver problemas de estimação de parâmetros, nos quais o modelo lida com dados e variáveis desconhecidas [16]. Esse tipo de algoritmo é bastante utilizado através da aplicação de HMM (*Hidden Markov Model*).

Este tipo de algoritmo considera dois conjuntos de variáveis: um conjunto composto pelas variáveis conhecidas, chamado de conjunto D e outro formado pelas variáveis escondidas, o conjunto H . O modelo possui inicialmente um conjunto de parâmetros ω , sendo que a cada iteração os passos **E** e **M** são executados de forma a obter uma nova estimação de ω . No passo **E** é calculada a distribuição de probabilidades sobre o espaço H , sempre levando em consideração os parâmetros calculados no passo anterior e as variáveis observáveis. No passo **M**, para maximizar $P(H, D|\omega)$, ω é recalculado, aumentando a probabilidade dos dados observados com a nova parametrização. O algoritmo vai realizando as iterações alternando fases **E** e **M** até a parametrização se tornar estável.

2.1.5.2 O algoritmo TGR

Foi proposto em [17] um algoritmo chamado *Tree Grammar Reestimator* que generaliza um algoritmo para estimar probabilidades de uma cadeia HMM, conhecido como *forward-backward* [16]. Esse algoritmo assume que as probabilidades das regras de produção da gramática dada já estão inicializadas para então fazer uso de um método EM para reestimar as probabilidades das produções da gramática inicial. O algoritmo possui complexidade $O(n^2)$, no qual n é o tamanho típico de uma sequência [18].

O conjunto de variáveis conhecidas (D) são as regras da gramática inicial, enquanto que ω representa as probabilidades dessas produções. Tudo o que ainda não é conhecido do modelo, como produções ainda não inseridas são as variáveis escondidas.

A amostra de treinamento é composta por cadeias que possui informações estruturais (*strings* parentisadas). Essas cadeias são trabalhadas em um formato de árvore, na qual os caracteres pertencentes à cadeia são os nós folhas da árvore e os nós internos, que são numerados, representam a estrutura descrita pelos parêntesis. Desta forma, podemos ver na Figura 4 a árvore que representa a sequência $((a(bc)d)(a(bc)d))$.

Figura 4 – Árvore que representa a cadeia $((a(bc)d)(a(bc)d))$

Considerando que t é uma árvore e n é um nó desta árvore, denotamos por t/n a subárvore de t com raiz em n . Denotamos também $t \setminus n$ a árvore que sobra de t após remover t/n mantendo o nó n . Podemos observar na Figura 5 à esquerda $t/3$ e à direita $t \setminus 3$.

Dada a seguinte gramática estocástica (as probabilidades das regras de produção estão

Figura 5 – Representação de $t/3$ (esquerda) e de direita $t/3$ (direita)

entre colchetes para evitar confusão com o lado direito da regra):

$$S \rightarrow AA \ [1]$$

$$A \rightarrow aBd \ [0.1]$$

$$A \rightarrow dBa \ [0.2]$$

$$A \rightarrow cBb \ [0.3]$$

$$A \rightarrow bBc \ [0.4]$$

$$B \rightarrow ad \ [0.1]$$

$$B \rightarrow da \ [0.2]$$

$$B \rightarrow cb \ [0.3]$$

$$B \rightarrow bc \ [0.4]$$

Utilizando a representação de árvore explicada anteriormente, pode-se associar ao símbolo A da gramática a sub-árvore $t/3$, pois o nó 3 tem descendentes ($a \ 5 \ d$) e é possível derivar de A os símbolos aBd , e de forma recursiva podemos associar $t/5$ ao símbolo B . Fazendo a associação dos valores de probabilidade das regras de produção com as sub-árvores temos que $t/3$ receberá um valor igual a $0.1 \times 0.4 = 0.04$ (valores obtidos das produções $A \rightarrow aBd$ e $B \rightarrow bc$, respectivamente). A probabilidade da sequência com a estrutura dada segundo a gramática utilizada é dada por $t/1$ que é 0.16%.

O valor $in[n, Z]$ representa o valor de t/n quando n é associado ao símbolo Z . A ideia é que se Z está associado a n , isso significa que cada símbolo do lado direito de Z também deve associar-se a cada descendente de n , seguindo a ordem da esquerda para a direita

tanto na produção quanto na árvore, respectivamente. Dessa forma, devemos considerar os valores dessas associações quando for efetuado o cálculo do valor da associação do símbolo do lado esquerdo Z ao nó pai n . Este valor é recursivamente calculado da seguinte forma:

- $in[n, X] = 1$ para todos os nós folhas n e terminais X no qual n coincida com X .
- $in[n, X] = 0$ caso n não coincida com X .
- para todos os não terminais Z , todos os nós internos m , e todos os nós filhos de m $n_1, \dots, n_k$:

$$in[m, Z] = \sum_{Z \rightarrow Y_1 \dots Y_k \in R_S} in[n_1, Y_1] * \dots * in[n_k, Y_k] * P(Z \rightarrow Y_1 \dots Y_k)$$

onde R_S representa o conjunto das regras de produção que possuem o símbolo Z como lado esquerdo sendo $P(Z \rightarrow Y_k)$ a probabilidade associada à regra $Z \rightarrow Y_k$.

Da mesma forma, pode-se fazer a associação de $t \setminus n$ a um nó não terminal. Ainda no exemplo acima, $t \setminus 3$ poderia ser associado ao símbolo A , uma vez que os descendentes do nó pai do nó 3 são $(2 \ 3)$, A aparece em $S \rightarrow AA$, $t/2$ pode ser associado a A (primeiro A do lado direito da produção) e $t \setminus 1$ pode ser associado ao símbolo Z .

O valor $out[n, Z]$ representa o valor de $t \setminus n$ quando n é associado ao símbolo Z . Semelhantemente ao cálculo de in , quando um símbolo S é associado a um nó n , os outros símbolos do lado direito da produção onde S aparece devem ser associados aos nós irmãos do nó n . Além disso, deve ser associado ao símbolo do lado esquerdo da produção que contém Z , Z' , o nó l , pai de n . Este valor é calculado recursivamente da seguinte forma:

- para o nó raiz n , $out[n, Z] = 1$ se Z for o símbolo inicial e $out[n, Z] = 0$, caso contrário.
- para todos os não terminais Z , todos os nós internos l e m , sendo que l é pai de m , e todos os nós irmãos de m $n_1, \dots, n_k$:

$$out[m, Z] = \sum_{Q \rightarrow Y_1 \dots Y_k \in W_Z, Q \in V_n} in[n_1, Y_1] * \dots * in[n_k, Y_k] * P(Q \rightarrow Y_1 \dots Z \dots Y_k) * out[l, Q]$$

sendo W_Z o conjunto de produções com o símbolo Z no lado direito.

As sequências que esse estimador recebe são representadas por árvores, sendo que para cada árvore é utilizada uma matriz in para armazenar os valores de $in[n, Z]$ e uma matriz out para armazenar os valores de $out[n, Z]$.

No passo **E** do algoritmo são calculadas as probabilidades de todas as cadeias contidas na amostra de treinamento e os valores das tabelas *in* e *out* são recalculados mediante utilização dos cálculos descritos acima.

No passo **M** as probabilidades de todas as regras de produções da gramática são recalculadas para que se ajustem aos novos valores das tabelas *in* e *out*. Dessa forma, para cada produção da gramática a sua nova probabilidade deve se basear nos valores *in* e *out* considerando as possíveis associações de S com todos os nós m de todas as sequências t . Além disso, deve-se considerar não só a probabilidade atual da produção, mas também a probabilidade das sequências. O valor computado é uma medida que mostra o quanto essa produção foi utilizada na análise de todas as sequências. Baseando-se em todos os nós m de todas as sequências t a nova probabilidade $Z \rightarrow Y_1 \dots Y_k$ pode ser calculada:

$$\left(\sum_t \left(\sum_m out[m, Z] * P(Z \rightarrow Y_1 \dots Y_k) * in[n_1, Y_1] * \dots * in[n_k, Y_k] / P(t|G) \right) \right) / norm$$

na qual $P(t|G) = in[n, S]$ sendo n o nó raiz da sequência t , S o símbolo inicial da gramática G e $norm$ um fator utilizado para normalização.

A seguir podemos observar o esquema geral descrito anteriormente. O algoritmo recebe como entrada uma amostra positiva que possui informações estruturais e uma gramática g com probabilidades iniciais, e gera como saída uma gramática estocástica.

2.2 Classificação bayesiana

A teoria de decisão bayesiana é uma abordagem estatística para o problema de tomada de decisão. O teorema de Bayes nos mostra que $P(B|A) = P(A|B)P(B)/P(A)$.

Considerando duas classes de imagens $c1$ e $c2$, e, considerando que dada uma imagem qualquer ela pode pertencer aleatoriamente à classe $c1$ ou $c2$, temos que $c1$ e $c2$ podem ser vistas como valores de uma variável aleatória c .

De modo geral, utilizando a fórmula de Bayes para classificar uma nova imagem x temos que $P(c_i|x) = P(x|c_i)P(c_i)/P(x)$ e $P(x) = \sum_j P(x, c_j) = \sum_j P(x)P(c_j)$, ou seja, **posteriori = versossimilhança*priori/evidência**. A classificação de uma nova imagem ocorreria da seguinte forma:

- c_1 se $P(c_1|x) > P(c_2|x)$
- c_2 caso contrário

Entrada: Uma amostra positiva com informações estruturais e uma gramática g com probabilidades iniciais

Saída: Uma gramática estocástica

```

1 enquanto estabilizou( $d$ ,  $k$ ) = falso faça
2   para cada produção  $r$  fazer
3      $novaP \leftarrow 0$    %nova probabilidade da produção  $r$ 
4     para cada sequencia  $s$  da amostra fazer
5        $ps \leftarrow \text{in}(\text{raiz}(s), \text{simboloInicial}(g))$  %probabilidade da sequência
6        $parcial \leftarrow 0$ 
7       para cada nó  $n$  de  $s$  fazer
8          $parcial \leftarrow parcial + \text{out}(n, r) * \text{probabilidade}(r) * \text{ins}(n, r)$ 
9        $novaP \leftarrow novaP + parcial/ps$ 
10    % divide os valores calculados pelo fator de normalização
11    normalizaProbabilidades()
12 ins( $n$ ,  $r$ )
13 %Calcula o produto de  $\text{in}(m, y)$ , onde  $m$  e  $y$  são, respectivamente, nó filho de  $n$  e
    símbolo do lado direito da produção  $r$ 
14  $parcial \leftarrow 1$ 
15 se  $\text{numeroDeDescendentes}(n) = \text{numeroSimbolosDoLadoDireito}(r)$  então
16   para cada  $i$  de 1 ate  $\text{numeroDeDescendentes}(n)$  fazer
17      $parcial \leftarrow parcial * \text{in}(\text{i-ésimoDescendente}(n), \text{i-ésimoSimbolo}(r))$ 
18 senão
19    $parcial \leftarrow 0$ 
20 retorna  $parcial$ 

```

Algoritmo 3: Algoritmo IgMakinen

Como $P(x)$ é uma constante em relação às classes, a regra se resume a:

- c_1 se $P(x|c_1)P(c_1) > P(x|c_2)P(c_2)$
- c_2 caso contrário

Se $P(c_1) = P(c_2)$ a decisão se resume à verossimilhança.

O classificador das imagens utilizado neste projeto é um classificador bayesiano. Cada analisador sintático i utilizado fornece o valor de $P(x|G_i)$. O classificador vai então calcular $P(G_i|x)$, e classificar a imagem x para a classe c de maior posteriori.

2.3 Processamento de Imagens

Esta seção tem como objetivo apresentar os conceitos básicos sobre processamento de imagens. Seu conteúdo e definições foram extraídos de [19] e [20].

2.3.1 Definições básicas

Pode-se definir uma imagem como sendo uma função matemática ou uma representação de um objeto que quando armazenada pode ser interpretada e manipulada. De forma matemática, uma imagem pode ser definida como uma função f da ‘intensidade luminosa’ que um objeto reflete. Como é considerado o espaço bidimensional para a maioria das imagens, temos $f(x,y)$, onde x e y são as coordenadas espaciais e f é o valor da intensidade, ou seja, o brilho da imagem no ponto (x, y) . Ainda, uma imagem depende da ‘quantidade de luz’ incidente, i , em uma cena e da ‘quantidade de luz’ refletida, r , pelos objetos que compõem a cena. É interessante observar que os termos referentes à luz podem estar relacionados com qualquer tipo de radiação. Por exemplo: em imagens de radiografia, esta radiação refere-se à incidência de raios-X e a imagem é formada considerando a porcentagem desses raios que cada tipo de tecido absorve.

Portanto, pode-se afirmar que: $f(x,y) = i(x,y) * r(x,y)$, onde:

- $i(x,y)$ depende da fonte de luz, $(0 < i(x,y) < \infty)$;
- $r(x,y)$ depende do tipo de material que compõe o objeto, $(0 \leq r(x,y) \leq 1)$.

Pode-se notar que $r(x,y)$ é 0 (zero) para a absorção total e 1 (um) para reflexão total.

A representação de imagens digitais pode ser feita por meio de um vetor de valores discretos. O domínio é finito e pode ser representado por uma matriz retangular. Cada uma das posições desse vetor assume um valor no intervalo $[O, M]$, sendo O a cor mais escura e M a cor mais clara. Na prática, pode-se entender uma imagem como uma função contínua, que pode ser representada por valores obtidos em intervalos regularmente espaçados. Em cada ponto medido, os valores são quantificados em um número que pertence a uma escala de diferentes cores. Uma representação matricial de uma imagem pode ser:

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & \cdots & f(0,n-1) \\ f(1,0) & f(1,1) & \cdots & f(1,n-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(m-1,0) & f(m-1,1) & \cdots & f(m-1,n-1) \end{pmatrix}$$

Essa representação matricial permite manipular a imagem no domínio espacial de tal forma que seja possível obter dela as informações desejadas. Dá-se o nome de processamento de imagem a todas as operações que podem ser realizadas sobre uma matriz que representa uma imagem.

Denomina-se *picture element* (pixel) cada ponto representado na matriz-imagem. Como cada ponto da imagem é um pixel, esta é a menor unidade sobre a qual é possível aplicar operações. Para cada pixel são definidas algumas relações básicas como por exemplo, vizinhança, adjacência e conectividade. As técnicas de processamento de imagens podem ser divididas em processamento de baixo nível, processamento de nível médio e processamento de alto nível.

O processamento de baixo nível tem como objetivo explorar a redundância presente na vizinhança dos pixels. O foco desta fase é recuperar algumas estruturas das imagens, tais como descontinuidade de superfície, profundidade, orientações etc. O processamento de baixo nível está relacionado ao processo de segmentação de uma imagem. Segmentar uma imagem consiste em dividi-la em suas regiões ou objetos. Os algoritmos de segmentação são baseados em uma das duas propriedades básicas dos valores de intensidade: descontinuidade e similaridade. Considera-se processamento de alto nível as tarefas de reconhecimento de padrões em imagens. As técnicas utilizadas podem ser divididas em métodos de teorias de decisão ou em métodos estruturais. No caso de teorias de decisão, os padrões são descritos de forma quantitativa, tais como tamanho, área, textura etc. Na segunda categoria, os padrões são melhores descritos de forma qualitativa, tais como descritores relacionais.

2.3.2 *Quadrees*

Esta seção apresenta a estrutura hierárquica chamada *quadtree* que pode ser utilizada para representar uma imagem. O conteúdo apresentado foi extraído de [21] e [22].

Uma região é um subconjunto de uma matriz que pode ser vista como sendo composta por quadrados unitários (pixels). Em uma matriz binária os pixels que representam uma região possuem valor 0 e os pixels que não representam uma região possuem valor 1. Uma região também pode ser vista como a união de blocos, no qual todos os pixels têm valor 0.

Uma *quadtree* é uma representação em que cada bloco máximo tem um padrão de tamanho e posição. A abordagem para representar a região é baseada em sucessivas subdivisões de uma matriz-imagem em quadrantes. Se a submatriz resultante de uma divisão não consistir unicamente de pixels com os mesmos valores, então esta submatriz sofre um processo de divisão em novos quadrantes até que se obtenham blocos nos quais os pixels possuam o mesmo valor. Esse processo é representado por uma árvore em que cada nó não-folha tem quatro filhos, sendo que o nó raiz representa a matriz inteira.

Os 4 filhos do nó raiz representam quadrantes rotulados da esquerda para a direita com *NW* (*northwest*), *NE* (*northeast*), *SW* (*southwest*) e *SE* (*southeast*) e os nós folhas correspondem aos blocos para os quais mais divisões não são necessárias. As folhas são ditas serem *pretas* ou *brancas* dependendo se esses nós representam ou não uma região da imagem, respectivamente. Todos os nós não folhas são chamados *cinzas*.

Quadtrees vêm sendo empregadas em vários domínios da computação gráfica, análises de cenas, *design* arquitetural e reconhecimento de padrões. O desenvolvimento de *quadtrees* foi motivado pela grande necessidade de economizar espaço, à medida que uma imagem representada por esta estrutura pode se tornar bem compacta. Muitos algoritmos para processamento de imagens em *quadtrees* foram desenvolvidos. Por exemplo, *finding neighbor*, *image rotation* e *component counting* foram todos desenvolvidos em *quadtrees* [22].

Podemos observar na Figura 6 uma imagem e sua respectiva *quadtree*.

Figura 6 – Quadtree. (a) Matriz binária de pixels de uma imagem. (b) Blocos cujos pixels tem a mesma cor. (c) Quadtree correspondente. Fonte [22]

2.4 Gramáticas Aplicadas a Imagens

Existem algumas diferenças importantes entre gramáticas de *strings* e gramáticas aplicadas às imagens. A primeira grande diferença é a perda da ordenação da esquerda para direita que existe nas *strings*. Em uma *string*, para toda regra de produção $A \rightarrow \beta$ é assumida uma ordem linear na sequência dos símbolos de β [23]. Em imagens, é necessária a criação de ligações mais complexas entre os nós que possuem algum relacionamento,

ou seja, é necessário criar ligações explícitas nas árvores construídas para representar adjacência, paralelismo, concentricidade etc.

Outra diferença importante é que em uma gramática de *strings* não existe a noção de escala, pois isto se trata de uma propriedade única das imagens. Desta forma, não se pode ler ou escrever uma sentença em português em múltiplas escalas, mas é possível ter uma imagem representada com várias resoluções. Isso significa que o analisador sintático deve ser capaz de terminar imediatamente em qualquer nó da árvore de derivação no qual não seja possível perceber mais detalhes [23].

A última grande diferença é que as imagens, além de possuírem uma grande estrutura de objetos que podem ser decompostos pelas regras de produção, também contêm vários padrões estocásticos, como por exemplo, texturas que são melhores representadas por modelos *Markov random field* [23].

O próximo capítulo detalha os objetivos deste projeto de mestrado.

Capítulo 3

Objetivos

3.1 Objetivo geral

O objetivo principal deste trabalho foi avaliar a utilização de gramáticas estocásticas inferidas a partir de amostras de treinamento que continham as estruturas de *quadrees* das imagens no problema de classificação de imagens.

3.2 Objetivos específicos

Os objetivos específicos estão divididos nos seguintes itens a seguir:

- realizar uma a revisão sistemática abrangente sobre a utilização de gramáticas na tarefa de classificação e reconhecimento de padrões em imagens;
- realizar pré-processamento de imagens de tal forma que elas pudessem ser representadas mediante a utilização da estrutura de *quadrees*;
- Adaptar um arcabouço de geração de classificadores baseados em gramáticas para que ele possa ser utilizado no reconhecimento de padrões em imagens;
- testar os classificadores gerados para a tarefa de classificação das imagens.

O próximo capítulo contém a revisão bibliográfica sistemática realizada com o intuito de identificar o estado da arte com relação à aplicação de gramáticas para reconhecimento de padrões.

Capítulo 4

Revisão Bibliográfica Sistemática

Revisões sistemáticas tiveram início na área de saúde, principalmente na medicina, no final dos anos 80, dada a quantidade de pesquisas desenvolvidas sem uma metodologia apropriada [24]. Trata-se de uma abordagem formal em que se utiliza um conjunto de passos planejados e bem definidos de acordo com um protocolo previamente estabelecido e documentado. Desta forma, uma revisão bibliográfica sistemática produz resultados mais confiáveis, pois os mesmos são passíveis de serem auditados, o que a torna independente dos revisores que a conduziram.

Para este trabalho de mestrado, realizou-se uma revisão bibliográfica sistemática com o intuito de levantar o estado da arte e as lacunas existentes na utilização de gramáticas para reconhecimento de padrões em imagens. Esta revisão sistemática possibilitou a confecção do artigo "*Using grammars for pattern recognition in images: a systematic review*" que foi aceito para publicação [25].

Uma descrição do planejamento e da condução, os resultados obtidos, um levantamento das técnicas envolvidas e dos trabalhos já desenvolvidos com essas técnicas e as discussões acerca de tais trabalhos podem ser verificados nas próximas seções.

4.1 Planejamento e Condução

A seguir são apresentados os tópicos relacionados à revisão sistemática realizada, bem como os protocolos de pesquisa desenvolvidos durante sua condução.

Objetivo: identificar o estado da arte na área de reconhecimento de padrões em imagens utilizando gramáticas estocásticas, tendo como foco encontrar materiais relevantes no que diz respeito à inferência de gramáticas a partir de imagens e trabalhos relacionados ao reconhecimento de imagens para uma dada gramática.

Perguntas da pesquisa:

- Quais são os métodos utilizados para fazer inferência de gramáticas a partir de imagens?

- Quais são as técnicas utilizadas para reconhecimento de imagens para uma dada gramática?

Critérios de seleção de fontes:

- Palavras-chaves: *image, grammar, grammatical, syntactic, linguistic, inference and estimation*
- *Listagem de fontes:*
 - IEEE Xplore - (<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>)
 - Portal ACM - (<http://portal.acm.org/dl.cfm>)
 - Periódicos Capes - (<http://www.periodicos.capes.gov.br/>)

Foram considerados todos os trabalhos encontrados referentes à área de pesquisa em questão, independentemente do ano de sua publicação. Entretanto, artigos sobre vídeo e detecção de eventos foram excluídos, pois o foco principal da revisão realizada era imagens estáticas.

Sumarização dos resultados: foram extraídos de cada trabalho as técnicas empregadas, as primitivas utilizadas, os resultados obtidos no reconhecimento de imagens utilizando gramáticas e as técnicas para inferências gramaticais empregadas. Ao término da revisão bibliográfica, foi realizada uma análise mostrando as vantagens e desvantagens de cada método. Também foram levantadas possíveis lacunas existentes no estado da arte e que podem ser exploradas.

Condução da Revisão: a condução da revisão sistemática ocorreu em um período de Maio de 2011 a Agosto de 2012. As buscas pelos materiais que abordavam reconhecimento de padrões em imagens utilizando gramáticas retornaram 242 artigos, sendo 58 na base de dados do IEEE Xplore, 107 no portal ACM e 77 do Portal Capes. Do total de artigos levantados, após serem aplicados os critérios de inclusão e exclusão, 31 foram selecionados e analisados na íntegra, sendo 20 da base de dados do IEEE Xplore, 2 do portal ACM e 9 do Portal Capes. Os critérios para incluir um artigo nesta revisão foram: (a) Trabalhos que apresentaram métodos de inferência gramatical a partir de imagens; (b) Trabalhos que aplicaram técnicas de reconhecimento de padrões em imagens para uma dada gramática e; (c) Artigos que utilizaram técnicas de reconhecimento de padrões em imagens tais como grafos AND-OR ou gramáticas de atributos.

Para enriquecer os resultados, em Julho de 2012 uma nova pesquisa foi realizada na base de dados IEEE Xplore com as seguintes palavras-chaves: *image, vision, grammar, grammatical, syntatic, linguistic, hierarchical model, compositional model, reconfigurable model, shock grammar, NOT(video)*. Esta nova busca recuperou 28 artigos, 5 dos quais já haviam sido analisados previamente. Dos 23 artigos restantes, 19 foram adicionados à esta revisão sistemática após terem sido aplicados os critérios de inclusão e exclusão. É possível observar na Tabela 13 (Apêndice A) uma síntese dos 50 artigos analisados, na qual são exibidas as técnicas e métodos empregados em cada um deles. Em seguida, é exibida uma análise global dos trabalhos (seção 3.2), considerando os tópicos relevantes para identificar o estado da arte. Cada artigo foi alocado em uma categoria considerando o tipo de gramática empregada e o propósito do artigo considerando as áreas da visão computacional. É exposto na Tabela 13 um sumário dos trabalhos analisados ordenados de acordo com a sequência que eles aparecem nas seções 3.3.1 a 3.3.7.

4.2 Análise Global

A pesquisa realizada mostrou que nos últimos anos, principalmente na última década, muito vem sendo pesquisado acerca da utilização de métodos sintáticos para reconhecimento de padrões em imagens. Devemos destacar que a maioria dos artigos apresentados utilizava uma gramática, criada manualmente, capaz de reconhecer uma determinada classe de imagens.

A evolução desta área de pesquisa com relação à quantidade de artigos publicados pode ser observada na Figura 7. Segundo o trabalho apresentado em [23], vários aspectos contribuem para a evolução recente desta área de pesquisa e que não estavam consolidados na década de 70, por exemplo: (i) um arcabouço matemático e estatístico consistente para integrar vários modelos de imagens; (ii) modelos de aparência mais realísticos para mapear primitivas aos pixels da imagem; (iii) algoritmos mais poderosos para classificação discriminativa e métodos generativos; (iv) grande quantidade de imagens digitais para treinamento e testes.

Em especial, pode-se perceber na Figura 7 que houve picos de publicações nos anos de 2005, 2007, 2009 e 2011. Em 2005, as principais técnicas utilizadas nos artigos desta revisão bibliográfica foram: grafos AND-OR, gramáticas de adjacência e *2D tabled matrix grammar*. Em 2007, as técnicas mais utilizadas foram *Spatial Random Tree Grammars*, gramáticas fuzzy para reconhecimento de texturas, grafos EDT e *L-System*. O ano de 2009

Figura 7 – Gráfico com a quantidade de artigo publicados ano a ano.

foi o que apresentou a maior quantidade de artigos publicados, sendo que grafos AND-OR apareceram em 5 destes artigos. Em 2011 as principais técnicas utilizadas foram grafos AND-OR, gramáticas estocásticas de atributos sensíveis ao contexto, lógica de predicado, modelos de incerteza, grafos gramaticais ETPL(k) e *shock graphs*. Os demais trabalharam apresentaram gramáticas livres de contexto para reconhecimento de padrões em imagens médicas, gramáticas fuzzy para reconhecimento de padrões em textura e gramáticas *L-System* para construções de objetos.

Analisando a Tabela 13 podemos perceber que vários tipos diferentes de gramáticas foram utilizadas e que elas são, geralmente, combinadas com várias outras técnicas para reconhecimento de padrões em imagens. Deve-se observar que embora todos os artigos analisados utilizassem gramáticas, alguns as utilizavam para reconhecimento de objetos em imagens [26], outros tinham foco no reconhecimento de padrões em texturas [27] e outros ainda utilizavam métodos sintáticos para criar objetos novos, geralmente árvores e flores [28]. Em [29] foram utilizadas gramáticas na tarefa de segmentação de imagens, em [30] foi apresentado um método para trabalhar com escalas das imagens e em [31] são apresentadas técnicas para reconhecimento de *layout* de documentos. Na Figura 8 é exibido um gráfico no qual pode perceber que métodos sintáticos são mais utilizados para reconhecimento de objetos (66%), em detrimento ao reconhecimento de padrões em textura (6%), criação de novos objetos (14%), segmentação de imagem (4%), mudanças de escalas (2%) e reconhecimento de *layout* de documentos (8%).

A Figura 9 representa uma visão das principais técnicas encontradas nos artigos da revisão realizada. Nela podemos notar que 34% dos artigos utilizam gramáticas estocásticas para reconhecimento de padrões em imagens. Grafos AND-OR e o algoritmo

Figura 8 – Gráfico com a tendência da utilização de métodos sintáticos.

top-down/bottom-up estão presentes em 16% dos trabalhos. Gramáticas e lógica *fuzzy* aparecem relacionadas a reconhecimento de padrões em textura (6%). Artigos que apresentavam gramáticas do tipo *L-System* trabalhavam com construção de objetos, notadamente modelagem de árvores, flores e fungos e representam um total de 10%. Outra técnica encontrada foi a *Spatial Random Tree* que faz uso do algoritmo *center-surround* para classificação de objetos (10%). Artigos que utilizaram *shock grammars* correspondem a 8%. Artigos que não apresentam nenhuma das técnicas mencionadas anteriormente representam 42% dos trabalhos analisados durante esta revisão sistemática.

Ainda analisando a Tabela 13 podemos notar a forte presença de métodos estatísticos utilizados na tarefa de reconhecimento de padrões. Os métodos estocásticos mais comuns encontrados na revisão sistemática realizada foram modelos de árvores escondidas, mistura de modelos escondidos de Markov, cadeias de Markov Monte Carlo, entre outros.

É comum encontrar avaliações tanto qualitativas quanto quantitativas nos trabalhos analisados. Para as análises qualitativas sempre são exibidas as imagens obtidas com a aplicação da técnica ou do modelo proposto, enquanto que para as análises quantitativas, geralmente são exibidas taxas de acerto e erro na classificação de um objeto, bem como curvas ROC (*Receiver Operating Characteristic*).

Podemos notar na Figura 10 que os Estados Unidos é o país com o maior número de publicações (26), seguidos pela China (8), Polônia (5), Espanha (4) e Portugal (4).

Figura 9 – Gráfico com as principais técnicas utilizadas.

Figura 10 – Gráfico com o número de publicações para cada País.

4.3 Gramáticas de Imagens para Reconhecimento de Padrões

Tendo como base a revisão sistemática conduzida, foi possível encontrar na literatura várias técnicas com abordagens sintáticas para reconhecimento de padrões em imagens ou modelagem de objetos, as quais serão apresentadas a seguir.

4.3.1 Reconhecimento de Objetos

4.3.1.1 Gramáticas Estocásticas

O trabalho [32] apresenta uma abordagem probabilística para decodificação de imagem. A abordagem procura maximizar *a posteriori* de uma função objetiva que representa um documento e utiliza um modelo de tarefa específica para decodificar o documento. O modelo utiliza uma gramática estocástica convencional acrescida de uma coleção de atributos para cada regra. A associação dos atributos com as regras é feita utilizando funções que computam os valores dos atributos do lado esquerdo da regra, dados os valores dos atributos do lado direito. Os atributos do lado esquerdo são parâmetros para a instanciação de um objeto, enquanto os atributos do lado direito são parâmetros dos componentes do objeto. O artigo não apresenta resultados e conclusões acerca da abordagem utilizada.

Spatial Random Tree Grammars

Spatial Random Trees (SRT) [33] são modelos estocásticos de árvores escondidas nos quais os nós folhas representam partes de uma imagem. Neste modelo, tanto os estados dos nós da árvore são variáveis aleatórias, como toda sua estrutura é gerada de forma aleatória por uma gramática probabilística.

Multitree dictionaries são utilizados para o desenvolvimento do modelo SRT e são definidos por meio do formalismo de gramáticas. Os símbolos desta gramática podem representar regiões diferentes de uma imagem. As regras de produção podem dividir essas regiões em regiões menores por meio de suas aplicações recursivas. É possível observar na Figura 11 a aplicação das regras de produção em uma região retangular sobre o processo de divisão.

Figura 11 – (a) Ilustração das áreas retangulares de uma imagem. (b) árvore gerada correspondente. Fonte: [33]

Para a construção de uma SRT, é necessária uma imagem que represente um domínio Q . Também é necessária uma gramática G que descreva a segmentação hierárquica do

domínio Q . Nessa gramática, o símbolo inicial aparece apenas na raiz da árvore. Os símbolos não terminais aparecem apenas como nós internos e os símbolos terminais aparecem apenas nas folhas da árvore.

- o símbolo inicial é dado por $a = (0, Q)$;
- os símbolos não terminais têm a forma $a = (j, R)$, na qual R é um subconjunto retangular de Q (obtido por meio da subdivisão de Q) tal que $|R| > 1$ e j é um inteiro de um conjunto fixo $\{1, \dots, J\}$ que pode representar, por exemplo, a classificação dos pixels da imagem pertencentes a R em uma das J classes;
- os símbolos terminais têm a forma $a = \{j, \{n\}\}$ na qual $\{n\}$ é um retângulo 1×1 , um pixel, ($n \in Q$) e j é definido como anteriormente.

Seja $\mathfrak{R}$ o conjunto de todas as regiões válidas, ou seja, o conjunto de todos os sub-retângulos de Q , incluindo os retângulos 1×1 . As regras de produção são definidas como:

- a produção raiz é da forma $(0, Q) \rightarrow \{(j, Q)\}$ para $j = 1, \dots, J$;
- as demais regras são definidas para todos os símbolos não terminais $a = (j, R)$. Essas regras são da forma $(j, R) \rightarrow \{(j', R'), (j'', R'')\}$, para $j, j', j'' = 1, \dots, J$ e para todas as partições de R em dois retângulos congruentes R' e R'' .

Para cada símbolo raiz ou não terminal presente na árvore é definida uma distribuição de probabilidade. É possível observar na Figura 12 um exemplo das regras de produção para os símbolos não terminais (subdivisões horizontais e verticais).

Figura 12 – Produções dos símbolos não terminais para subdivisões das regiões retangulares. Fonte: [33]

Para realizar a divisão da imagem a gramática pode ser generalizada de forma que cada região pode considerar uma forma arbitrária, ou seja, cada região não precisa, necessariamente, ser um retângulo.

Foi utilizado em [33] e [34] o modelo probabilístico hierárquico SRT. A principal contribuição desses trabalhos é a introdução de classes de distribuição de probabilidades sobre as árvores para cálculos exatos da verossimilhança em tempo polinomial. Outra contribuição importante é o desenvolvimento dos algoritmos que podem ser utilizados nas tarefas de segmentação e de classificação. Os símbolos utilizados na gramática podem, por exemplo, representar diferentes regiões de uma imagem, como pode ser visto na Figura 11. Uma árvore começa com um elemento único A (*raiz*) e é construída por meio da aplicação das regras de produção, na qual cada produção $a \rightarrow \alpha$ é retratada como um nó rotulado e cujos filhos são rotulados com elementos de α . A taxa de acerto do modelo proposto foi de aproximadamente 97%, quando analisadas imagens de casas, prédios e fachadas de lojas.

Este arcabouço foi evoluído para uma versão mais robusta, descrito em [35], para ser aplicado não apenas em imagens, mas também para qualquer dado multidimensional. Em sua nova formulação, um símbolo terminal é um vetor de características, que é extraído de uma região da imagem por meio de um algoritmo de segmentação.

Grafos AND-OR

Um grafo AND-OR é uma sêxtupla que representa uma gramática de imagem $G = \langle S, V_N, V_T, R, \Sigma, P \rangle$. Na qual S é o símbolo inicial, representando a cena ou a categoria de um objeto, V_N representa o conjunto de nós não terminais, V_T representa os nós terminais (primitivas, partes e objetos), R é o conjunto de relações entre os nós, Σ é o conjunto de todas as configurações válidas¹. que podem ser derivadas de G e P representa o modelo de probabilidade definido no grafo AND-OR. Os nós AND de um grafo AND-OR têm como descendentes símbolos do lado direito de uma mesma produção, enquanto nós OR têm como descendentes nós que representam produções com o mesmo lado esquerdo. Por exemplo, as seguintes regras de produção de uma gramática probabilística $A \rightarrow ab \mid cc$ com probabilidade $p|(1 - p)$ pode ser representada por um grafo AND-OR, conforme pode ser visto na Figura 13.

Para representar informações de contexto, a árvore AND-OR é transformada em um grafo AND-OR adicionando arestas horizontais ligando os nós que possuem algum tipo de relação (nós OR filhos de um mesmo nó AND). A adição destas arestas produz um grafo AND-OR que representa uma gramática de imagem [23].

A Figura 14 (a) exibe um grafo AND-OR para uma gramática com as seguintes regras

¹Uma configuração válida é uma sequência de símbolos que pode ser obtida através de derivações utilizando R

Figura 13 – Grafo AND-OR que representa uma regra de produção. Fonte: [23].

de produção:

- $A \rightarrow B \ C \ D$
- $B \rightarrow E \mid F$
- $E \rightarrow 1$
- $F \rightarrow L \ M$
- $L \rightarrow P \mid Q$
- $P \rightarrow 2$
- $Q \rightarrow 3$
- $M \rightarrow Q \mid R$
- $R \rightarrow 4$
- $C \rightarrow G \mid H \mid I$
- $G \rightarrow M \ 5$
- $H \rightarrow N \ O$
- $N \rightarrow S$
- $S \rightarrow 6$
- $O \rightarrow T \mid U$
- $T \rightarrow 7$
- $U \rightarrow 8$

- $I \rightarrow 9$
- $D \rightarrow I \mid J \mid K$
- $J \rightarrow 10$
- $K \rightarrow 11$

Figura 14 – Exemplo de grafo AND-OR: (a) grafo no qual os nós AND são representados por círculos com bordas sólidas, os nós OR por círculos com bordas tracejadas e os nós terminais por quadrados; (b - c) grafos sintáticos que foram extraídos de (a); (d - e) configurações obtidas por meio dos grafos sintáticos (b) e (c). Fonte: [23].

Nesta figura, os círculos com bordas sólidas representam os nós AND, os círculos com bordas tracejadas representam os nós OR e os nós terminais são representados por quadrados. Os nós OR possuem rótulos para classificação em vários níveis, como categoria de cena, classes de objetos, partes etc. Devido à sua natureza recursiva, objetos podem ser adicionados a um grafo AND-OR, construindo um grande grafo que, teoricamente, pode representar qualquer imagem.

A Figura 15 exibe um grafo AND-OR para a categoria de objetos relógio. Os grafos sintáticos representam instâncias deste grafo AND-OR e estão representados por arestas em negrito.

O modelo estocástico de um grafo AND-OR deve definir probabilidades para cada grafo sintático $\mathbf{pg}$. Como cada um destes grafos produz uma configuração C , temos que $p(\mathbf{pg};\theta)$ produz a probabilidade marginal da configuração, sendo θ o parâmetro da

Figura 15 – Exemplo de um grafo AND-OR para reconhecimento de relógios e dois possíveis grafos de derivação. Fonte: [36].

distribuição de probabilidade. Por definição, temos que um grafo sintático **pg** é uma árvore **pt** com o incremento das relações E , ou seja $\mathbf{pg} = (\mathbf{pt}, E)$ [23].

O algoritmo *top-down/bottom-up* utilizado é um algoritmo guloso para maximização da probabilidade Bayesiana *a posteriori* que é processado em três fases. A primeira fase é a detecção *bottom-up*, na qual hipóteses são criadas com um determinado peso. Na segunda fase, o algoritmo escolhe a hipótese com o maior peso entre todos os candidatos e a aceita se ela aumenta a probabilidade Bayesiana. Na terceira fase ocorre a integração *top-down/bottom-up*. Cada hipótese presente no grafo sintático atual casa com uma regra de produção com atributos passados para o nó não terminal. Esses nós não terminais também casam com outras regras de produção, que então geram propostas *top-down* para predição. Os pesos das propostas *top-down* são calculados com base nas probabilidades posteriores. Dessa forma, sempre que um novo nó é adicionado ao grafo é criada uma nova subconfiguração que ativa as seguintes ações: criação de potenciais novas propostas *top-down*, inserindo-as em uma lista; redefinição dos pesos de algumas propostas candidatas; transmissão dos atributos entre os nós e seus parentes por meio de equações de restrições associadas com a regra de produção. Ao término do algoritmo obtém-se uma configuração que representa uma imagem.

No trabalho [3] foram utilizados grafos AND-OR para reconhecimento de padrões em imagens. Entretanto, antes de apresentar este trabalho devemos entender o que são os *oriented triplets*, pois esta estrutura foi utilizada como primitivas.

Uma imagem pode ser representada por alguns pontos de interesse. Esses pontos são

escolhidos por meio da aplicação do operador K_b (Kadir-Brady) [37], sendo que cada um desses pontos é chamado de *attributed feature* (AF). Cada AF presente em uma imagem é representado por sua localização, orientação e um vetor de aparência. A combinação desses AFs, em particular uma trinca, é utilizada para formar os *oriented triplets*. Esses triângulos possuem propriedades geométricas que os tornam invariantes a operações de escala e de rotação [3]. Estas propriedades incluem os ângulos entre os vértices e seus ângulos relativos, como podem ser observados na Figura 16. Os AFs e os triângulos formados podem ser vistos na Figura 17.

Figura 16 – *Oriented triplets*. β - ângulos internos, θ - orientações dos vértices e α - ângulos relativos. Fonte [3]

Figura 17 – AFs e os oriented triplets. Fonte [3]

Para criar um arcabouço teórico para aprendizado não supervisionado de modelos probabilísticos para geração e interpretação de imagens naturais, o trabalho [3] apresentou a utilização do modelo chamado *Probabilistic Grammar Markov Model* que combina *Markov Random Fields* e gramáticas livres de contexto probabilísticas. Para a construção do modelo probabilístico foram utilizados triângulos orientados de AFs como primitivas. A tarefa de aprendizagem do modelo pode ser dividida em duas partes: aprendizagem da estrutura do modelo; e aprendizagem dos parâmetros do modelo. Para a primeira tarefa,

a estratégia adotada foi construir um grafo AND-OR de forma iterativa à medida que mais triângulos vão sendo adicionados ao grafo. Para a segunda tarefa, foram utilizadas técnicas de agrupamento para propor adição de novos triângulos ao grafo e para validar ou rejeitar estas propostas. O método apresentado é independente de posição, orientação e escala dos objetos. Os testes foram realizados com imagens obtidas da *Caltech*² (13 categorias de objetos) e o modelo apresentou uma taxa de 90% de resultados verdadeiros positivos e uma taxa de 10% de resultados falsos positivos. Estes resultados são para classificações entre imagens que continham um objeto procurado e de imagens que possuíam apenas o fundo.

Com o intuito de localizar faces humanas em imagens, foi proposto em [38] um método que possui uma arquitetura que trabalha em duas camadas. Na primeira extraem-se características dos componentes faciais utilizando fator de certeza baseado em configurações geométricas, enquanto na segunda camada utilizam-se algoritmos *top-down/bottom-up* para percorrer um grafo AND-OR manualmente construído com a finalidade de localizar a face. Os testes foram realizados com 3022 imagens e a taxa de detecção de faces obtida com o método foi de 93.2%.

Foi apresentado em [39] e em [26] uma gramática de atributos capaz de representar uma imagem. Além disso, os estudos apresentam o efeito de um algoritmo *top-down/bottom-up* utilizado para análise sintática no processo de maximizar a probabilidade *a posteriori*. A gramática criada possui um nó raiz para a cena, um nó não terminal recursivo para objetos e superfícies, uma classe de primitivas e seus nós terminais (retângulos planares projetados nas imagens). Todos os nós terminais e não terminais são descritos por atributos que representam suas propriedades geométricas e sua aparência. Essa gramática possui seis regras de produção. Uma regra expande o nó raiz (cena) em m objetos independentes, outra regra instancia um nó não terminal em uma primitiva, as quatro regras restantes arranjam os objetos ou superfícies recursivamente em quatro possibilidades: alinham os m objetos; aninham um objeto dentro de outro; alinham três retângulos para formar um cubo; arranjam objetos no formato $m \times n$ em uma malha. Dada uma imagem de entrada, o objetivo é calcular, em tempo de execução, a árvore sintática na qual cada nó não terminal corresponda a uma regra de produção. A gramática utilizada pode ser vista na Figura 18 e alguns dos resultados obtidos podem ser vistos na Figura 19.

Em [40] foi apresentado um grafo gramatical AND-OR para representar o corpo hu-

²<http://www.vision.caltech.edu/html-files/archive.html>

Figura 18 – Regras de produções utilizadas na gramática para reconhecimento de retângulos. Fonte [26].

Figura 19 – Resultados de alguns experimentos. Os retângulos reconhecidos são apresentados na última coluna. Fonte [26].

mano como uma composição articulada de partes reconfiguráveis. O fato de este modelo ser reconfigurável permite a substituição de partes compatíveis com diferentes atributos tais como aparência de roupas. Cada regra de produção define um modelo de aparência para a raiz, um conjunto de partes filhas e um conjunto de restrições geométricas e atributos para as partes filhas. Um modelo probabilístico é definido sobre os analisadores sintáticos e a inferência exata é calculada para certas restrições. Os resultados obtidos foram competitivos com o estado da arte: 88.4% (método apresentado) contra 86.9% (estado da arte) para a estimação das partes corretas na média.

É apresentado em [41] a utilização de grafos AND-OR para guiar a anotação imagens com o intuito de criar uma base de dados de imagens. As imagens são analisadas e sua

descrição é transformada em grafos AND-OR. De acordo com os autores, este tipo de grafo permite uma melhor extração dos dados. A base de dados apresentada possuía duas camadas: a primeira gerencia as relações visuais do modelo com o grafo AND-OR e a segunda camada é utilizada para análise dos dados com base em objetos físicos no banco de dados. Para indexar o grafo AND-OR na primeira camada foi utilizada a categoria do objeto. Os dados exportados desta base de dados podem ser vistos como imagens brutas, contornos de objetos, partes de interesses etc.

Em [42] foi apresentada uma meta-gramática utilizada para escrever regras para detecção e reconhecimento de objetos. As regras para descrever os objetos seguem a estrutura de um grafo AND-OR. Na gramática proposta, o conjunto de símbolos terminais é formado por características da imagem de baixo nível. A representação do objeto é restringida para expressar, de maneira hierárquica, a acomodação da diversidade durante a construção. Para fundamentar a gramática proposta a um caso específico de aplicação, foi implementado um compilador utilizado para realizar a análise sintática e traduzir a gramática de um objeto em uma implementação em prolog. A técnica foi utilizada para detectar componentes em uma interface gráfica (GUI) com o objetivo de automatizar a detecção de componentes para o processo de teste de software. Foram utilizados 6 conjuntos de imagens com um total de 20 a 40 objetos por categoria (*pushbutton*, *menulist*, *radiobutton* e *checkbox*). As taxas de detecções obtidas foram: 90% para *pushbutton*, 100% para *menulist*, 90% para *radiobutton* e 67% para *checkbox*, sendo que os números de alarmes falso foram de 0.17, 0, 0.33 e 2.67, respectivamente.

Gramáticas Estocásticas Sensíveis ao Contexto

Foi proposto em [43] um arcabouço matemático para construir um modelo hierárquico probabilístico de imagens que podem utilizar relacionamentos contextuais arbitrário. Diferente de uma gramática formal tradicional, a arquitetura proposta permite o compartilhamento de subpartes entre várias entidades. Além disso, o modelo não limita interpretações a árvores únicas, ou seja, zero ou mais instâncias do padrão modelado podem ser detectadas. O método proposto foi testado em um conjunto de 385 imagens contendo placas de carros e a taxa de detecção correta foi superior a 98%.

Em [44] foi utilizada uma gramática estocástica sensível ao contexto para reconhecer estruturas regulares que possuíam simetrias. Foi utilizado o conceito de escopo, no qual a configuração de um objeto é colocada dentro de um “*container*” (um retângulo, por exemplo). Esses escopos podem representar, por exemplo, fileiras de janelas. Além disso, foram introduzidas exceções espaciais e estruturais (presença de buracos) nessas fileiras

de elementos. A ideia consiste em descrever um vetor semiregular de elementos. Uma *lattice* é um *layout* espacial de elementos terminais com possíveis derivações espaciais de localidade. Foi representada em [44] essa estrutura utilizando uma gramática estocástica sensível ao contexto que descreve como uma imagem contém os escopos.

Foi apresentado em [45] um arcabouço que utiliza grafos gramaticais sensíveis ao contexto para extrair telhados de construções presentes em imagens aéreas. A gramática proposta é formada por cinco regras de produção e três tipos de primitivas compartilhadas (quadriláteros). Cada regra de produção contém equações para restringir atributos de um nó pai com relação aos seus filhos. Além disso, ligações entre os nós no mesmo nível foram criadas para representar seus relacionamentos espaciais e de aparência. Para extrair o telhado foi utilizado um algoritmo *bottom-up* que gera hipóteses (quadriláteros) por meio do agrupamento de estruturas presentes na imagens, tais como linhas retas, linhas paralelas etc. Os autores não apresentaram resultados quantitativos.

4.3.1.2 Modelos de Markov e Lógica *Fuzzy*

Foi apresentado em [46] uma abordagem que faz uso de um descritor estrutural de contexto para extração de objetos em imagens, sendo que este descritor pode encontrar partes de objetos com estruturas similares. A abordagem também utiliza triângulos orientados como uma conjunção de partes e treina o modelo utilizando *Mixture Hidden Markov Model* (MHMM) e *Adaptive Neuro-Fuzzy Inference System* (ANFIS) para a etapa de categorização do objeto. Para gerar a gramática a partir das partes de objetos, são utilizados blocos desses triângulos orientados em que cada bloco é construído a partir de suas partes centrais. Esta estrutura cresce baseada no número de partes detectadas por meio da adição de novos triângulos. Para testar a abordagem proposta foram utilizados modelos de quatro classes de imagens: motos, faces, aviões e, carros. Utilizando ANFIS, as taxas de reconhecimento foram de 75.3%, 88%, 85% e 70.2% para as partes de cada uma das classes, respectivamente. Utilizando MHMM, as taxas de reconhecimento foram de 96.2%, 98.3%, 95.2% e 82.7%, respectivamente.

4.3.1.3 Expressões Regulares Aumentadas

Augmented Regular Expressions (ARE) é um formalismo utilizado para descrever, reconhecer e aprender uma classe não trivial de linguagens sensíveis ao contexto. Essas expressões aumentam o poder descritivo de expressões regulares incluindo um conjunto

de restrições que envolvem o número de instâncias dos operadores da operação estrela³ em cada *string* da linguagem. É apresentado em [47] um método de inferência gramatical para aprender o modelo de gramática a partir de um conjunto de amostras positivas e negativas. Cada modelo é representado por um *pseudo-bidimensional ARE*, no qual cada linha é representada por um ARE e todas as colunas juntas são representadas por outro ARE. O trabalho descreve o processo de aprendizagem e os resultados da aplicação deste método para aprender sinais de trânsito. Durante essa aprendizagem, as linhas do modelo da imagem são representadas por meio de um autômato de estados finitos (FSA - *Finite-State Automata*). A forma utilizada para aprender o FSA por meio de indução é chamada *Active Grammatical Inference*. O processo utilizado nessa metodologia é concebido como uma sequência de ciclos de aprendizado, em que cada ciclo inclui uma combinação de técnicas neurais e simbólicas, no qual o controle do próximo treinamento neural é dinamicamente modificado pela aquisição de informação ou pela imposição de informação externa.

4.3.1.4 Lógica de Predicado

[48] apresentam uma abordagem baseada em lógica de predicados para reconhecer e detectar padrões de objetos em imagens. São exploradas duas classes de objetos: pedestres, observados por câmeras de seguranças e estruturas complexas criadas pelo homem e que podem ser vistas em imagens de satélites (*surface-to-air missile* - SAM). O método apresentado tenta analisar um padrão de objetos por meio de modelagem e especificação de gramáticas. Essas gramáticas são codificadas como regras de primeira ordem e a análise do objeto consiste em procurar através do espaço de características a melhor solução que satisfaz às restrições lógicas. No caso de detectar partes do corpo humano, foram treinados vários classificadores de máquinas de vetores de suporte (SVM - *Support Vector Machine*) nos histogramas de orientações de gradiente com uso da função *sigmoid* para a saída de cada SVM. As regras para validar ou rejeitar uma hipótese de ser humano são baseadas em informações geométricas e informações sobre as cenas que são inseridas *a priori* no sistema. Essas informações são acerca do tamanho das pessoas e regiões nas quais espera-se encontrar os pés das pessoas.

³Em expressão regular, se V é uma linguagem, então V -estrela pode ser descrito como um conjunto de todos de todos os elementos que podem ser formados por meio da concatenação de zero ou mais elementos de V

4.3.1.5 Gramáticas de Adjacência

Uma gramática de adjacência é uma quintupla $G = (V_T, V_N, S, P, C)$ na qual V_T representa o alfabeto de símbolos terminais, V_N denota o alfabeto de símbolos não terminais, $S \in V_N$ é o símbolo inicial da gramática, C é um conjunto de restrições aplicado aos elementos da gramática, P representa o conjunto de produções definido como $\alpha \rightarrow \beta_1, \dots, \beta_n$ se $\Gamma(\beta_1, \dots, \beta_n)$ na qual $\alpha \in V_N$ e todos os $\beta_j \in V_T \cup V_N$ e Γ é uma restrição de adjacência definida sobre os atributos de β_j .

Os símbolos β_j podem aparecer em qualquer ordem. Por exemplo, para a regra de produção $\alpha \rightarrow \mu, \nu, \sigma \in P$ deve-se considerar todas as seis possibilidades de permutação dos símbolos μ, ν e σ como uma substituição igualmente válida para o símbolo α .

Um exemplo de desenho pode ser visto na Figura 20 (a), já os traços que formam o desenho podem ser vistos na Figura 20 (b), enquanto que a Figura 20 (c) exhibe a regra de produção que combina os traços para formar o desenho. Observe que esse desenho possui dois traços do tipo segmento e uma restrição de adjacência entre os traços do tipo intersecção [49].

Figura 20 – Elementos de um símbolo. Baseado em [49].

Estudos para reconhecimento de desenhos gráficos, com foco principal em reconhecimento de desenhos feitos a mão e gestos, foram apresentados em [49] e em [50]. O processo de reconhecimento apresentado consiste de três fases. A primeira fase, extração de primitivas, consiste em aproximações dos traços desenhados por primitivas (arcos e segmentos de retas). A segunda fase é um estágio sintático para reconhecimento de objetos compostos em que os símbolos de uma determinada classe são reconhecidos por uma gramática de adjacência. A última fase é responsável por aplicar regras semânticas às instâncias dos símbolos reconhecidos no desenho. A abordagem criada foi testada em mais de 700 símbolos e obteve uma taxa de reconhecimento acima de 87.7%.

4.3.1.6 Grafos Gramaticais ETPL(k)

Em [51] e em [52] é apresentada uma abordagem para interpretação de imagens médicas. Em particular, o trabalho foca em reconhecimento de doenças cardíacas (es-

tenoses) que podem ser detectadas a partir de imagens 3D (tridimensional) de vasos coronários. Inicialmente a imagem sofre um processo de *esqueletização* que representa sua estrutura morfológica. A imagem *esqueletizada* é utilizada para formar um grafo no qual cada vértice é representado por um ponto de ramificação da artéria principal, sendo que este grafo representa relações espaciais entre as artérias. Esta representação espacial pode ser formalizada pelo uso de grafos gramaticais chamados *Embedding Transformation-preserved Production-ordered K-Left nodes unambiguous* (ETPL(k)). O sistema criado é capaz de reconhecer com uma taxa de acerto de 85% os locais das estenoses, as quantidades e os tipos (concêntricas e excêntricas).

4.3.1.7 Gramáticas Livres de Contexto

Para reconhecimento de padrões em radiografias de ossos humanos, mais especificamente fraturas e fissuras de ossos dos braços e das pernas, foi criada a abordagem apresentada em [53]. A gramática livre de contexto criada pode reconhecer os seguintes tipos de fraturas: fissura, espiral, longitudinal, fratura deslocada, união retardada, transversal e adesão. Nesta gramática, os nós terminais (a, b, c, d, e, f, g, h) representam intervalos de ângulos. Por exemplo, o nó terminal *a* representa o intervalo entre -10 e 10 graus. A gramática proposta foi capaz de reconhecer a angulação entre as partes que constituíam os ossos humanos de tal forma que as fraturas pudessem ser identificadas. O sistema criado obteve um desempenho superior a 90% na interpretação de fraturas de ossos de pernas humanas.

Um método para indexação e recuperação de imagem foi apresentado em [54]. As gramáticas livres de contexto apresentadas são capazes de reconhecer a topologia dos órgãos e partes do corpo humano (artérias, ossos e pâncreas) de tal forma que seja possível identificar lesões visíveis e doenças nesses órgãos e estruturas. Uma das técnicas utilizadas cria uma gramática baseada em grafos sintáticos a partir de imagens (obtidas por meio de exames de radiografia) segmentadas de ossos humanos, conforme pode ser visto na Figura 21. Nesta técnica, os símbolos terminais são representados pelos centros de gravidades de cada osso da mão. Outra técnica apresentada utiliza uma gramática baseada na topologia de artérias e de estrutura do pâncreas, na qual é possível modelar estenoses arteriais e ajudar no diagnóstico de câncer. A aplicação das técnicas propostas alcançou uma taxa de reconhecimento de aproximadamente 93%, quando utilizado para analisar e interpretar lesões.

Foi desenvolvido em [55] um método para reconhecimento de objetos que utiliza as

Figura 21 – Geração de grafos gramaticais utilizados para construção de uma gramática. (a) Descrição das estruturas da palma. (b) Representação do grafo em uma imagem de radiografia. (c) Apenas a representação em forma de grafo. (d) Grafo invertido para exemplificar as regras de produção. Fonte [54].

vantagens das metodologias *bottom-up/top-down*. A abordagem explora aspectos conceituais de gramáticas livres de contexto, estatísticas não paramétricas e campos poligonais. A abordagem é formada não apenas por modelos sintáticos hierárquicos (HSM - *Hierarchical and Syntactic Models*) utilizados para lidar com restrições contextuais, como decomposição de objetos em partes e subpartes, mas também por modelos de dados utilizados para descrever funções de verossimilhança de dados observados considerando cada instância da representação do objeto. Os modelos de dados, descritos pela estatística não paramétrica, são invariantes sob certas condições, por exemplo, contraste. Em um HSM, os tipos dos objetos são vistos como junções articuladas e respectivas partes; por exemplo, carros têm portas com janelas, mas também têm faróis, placas etc., sendo que um objeto é, geralmente, a concatenação de primitivas. Os testes foram realizados em um banco de dados de ferramentas, como martelos, pás, tesouras etc.

Foi apresentado em [56] um arcabouço para detecção de prédios. No arcabouço proposto, um conjunto de postos desorganizados é utilizado como entrada e uma árvore sintática cujos nós são superfícies e partes volumétricas é obtida como resultado. Para

aumentar a robustez à oclusão, foram consideradas as partes volumétricas que podiam ser inferidas a partir de subconjuntos das superfícies delimitadoras. A ideia é ver um prédio como uma árvore, cujos nós representam partes volumétricas que estão próximas umas das outras e cobertos por telhados (*patches* planares) que são considerados filhos dos volumes. Para realizar esta tarefa, foi introduzida uma gramática simples utilizada para capturar propriedades geométricas entre os volumes e os *patches* planares. A gramática proposta tem dois super-nós, “building” e “non-building”, usados como ascendentes de todos os outros nós não raiz. A existência desses super-nós proporciona realizar a detecção enquanto a análise sintática é executada, na qual todas as árvores sintáticas cuja raiz é o nó “building” são consideradas representações dos prédios. A simplicidade da gramática proposta permite uma forma eficiente de análise sintática utilizada para inferir árvores sintáticas de prédios. Os testes foram executados utilizando a base de dados *Wright State 100*, contendo aproximadamente 1 bilhão de pontos coletados de forma aérea e terrestre por *scanners*. A precisão obtida foi de 89.3%, contra 87.9% obtida pelo método de máquinas de vetores de suporte.

4.3.1.8 Grafos EDT

Uma proposta que consiste em derivar símbolos terminais para alguma classe de objetos desconhecida, de tal forma a representar uma imagem como uma sequência desses símbolos foi apresentada em [57]. Para realizar esta tarefa, a técnica *sparse kernel feature analysis* é aplicada a um conjunto de fragmentos aleatórios de imagens de um conjunto de treinamento, gerando uma sequência de componentes esparsos que representam as características da imagem. Para cada uma das imagens analisadas, as características extraídas são utilizadas para derivar as localizações dos símbolos terminais. Em seguida, um algoritmo é aplicado para codificar as posições dos símbolos terminais para formar um grafo chamado *Extended Dominator Tree* (EDT), levando em consideração a frequência da ocorrência do símbolo e de suas configurações no conjunto de imagens de treinamento. O algoritmo gera como saída uma sequência de símbolos na forma de grafo que pode ser utilizado para métodos de inferência gramatical. Uma possível limitação da técnica é que ela pode não ser capaz de processar objetos complexos, na qual as relações espaciais podem não ser suficientemente descritivas. Os testes foram realizados utilizando imagens de carros, sendo que para cada uma das 200 primeiras imagens do conjunto de treinamento foram utilizados 5 fragmentos de tamanho 20 x 20 pixels.

4.3.1.9 Gramáticas Regulares

Um método para interpretar e descrever como a trajetória de um objeto pode ser modelada por meio de características qualitativas foi apresentado em [58]. A evolução de um objeto em uma cena pode ser descrita por regras de uma gramática regular que devem ser baseadas em características observáveis, descrever objetos individuais, descrever relações entre os objetos e incluir composições de ações. Exemplos de características incluem reconhecimento de objetos (copo, pires), relações geométricas (alinhado, paralelo), descontinuidades temporais (entrar em um campo de visão, estática) etc. Quando uma dessas características é encontrada em um determinado estado, é acionada uma determinada regra de produção que representará uma transição de estado. Além disso, a cada transição é associada uma ação específica (manipulação, inicialização, término etc.). O sistema foi testado com uma sequência de imagens que descreviam uma mesa para o café da manhã. A descrição da cena foi realizada por um analisador sintático genérico que possuía 42 regras de produção. Como resultado da análise, é apresentada a seguinte descrição textual da cena:

- Um pires foi colocado sobre a mesa;
- Uma xícara foi colocada sobre a mesa;
- Uma xícara com um pires foi colocada sobre a mesa;
- Uma colher foi colocada sobre a mesa;
- O pote de chá agora está sobre a mesa;
- A jarra de leite está sobre a mesa;
- A mesa está posta, ‘O Chá está servido!’;

4.3.1.10 Gramática Plex Tridimensional

A ideia básica de uma gramática *plex* é considerar cada símbolo não terminal ou terminal como uma primitiva ou uma superfície composta possuindo um número arbitrário n de curvas (*attaching curves*) para conexões com outras curvas. Esta estrutura é chamada de n *attaching-curve entity* (NACE) e estruturas formadas pela interconexão destas entidades são chamadas estruturas *plex* [59].

Uma gramática *plex* tridimensional pode ser representada por uma sêxtupla $G_p = (N, \sum, P, S, I, i_0)$, na qual N é um conjunto finito não vazio de elementos não terminais NACEs; $\sum$ é um conjunto finito não vazio de elementos terminais NACEs; $N \cap \sum = \emptyset$; P é o conjunto de regras de produção; $S \in N$ é o símbolo inicial; I é um conjunto de símbolos chamados identificadores; $I \cap (N \cup \sum) = \emptyset$; $i_0 \in I$ é o identificador especial nulo.

Os símbolos I são utilizados para identificar a junção das curvas (*attaching curves*) de NACE. Cada uma dessas curvas tem um identificador em I , sendo que não existem duas dessas curvas do mesmo NACE que tenham o mesmo identificador. O identificador nulo i_0 funciona como um indicador de local e não pode ser associado com nenhuma dessas curvas.

Uma gramática livre de contexto tridimensional *plex* tem produções da forma $A\Delta_A \rightarrow \chi\Gamma_\chi\Delta_\chi$, na qual A é um único NACE ou componente, Δ_A é a lista de curvas do lado esquerdo (*left-side tie-curve list*), χ é a lista de componentes do lado direito (*right-side component list*), Γ_χ é a lista de interconexões do lado direito (*right-side interconnection list*) e Δ_χ é a lista de curvas do lado direito (*right-side tie-curve list*). Assume-se que χ é não nulo e que um único NACE A aparecendo em qualquer contexto pode ser substituído pelo *subplex* dado por $\chi\Gamma_\chi$.

A lista de componentes do lado direito é uma *string* da forma $\chi = a_1a_2\dots a_j\dots a_n$, na qual a_i é um único NACE chamado componente. Γ_χ especifica como e quais NACEs estão interconectados. O componente $\chi\Gamma_\chi$, quando considerado em pares, define a estrutura envolvida em uma regra de produção.

Uma gramática *plex* é um caso especial de gramáticas de atributos, que por sua vez, permite atribuir atributos às regras de produção, por exemplo, probabilidades. Na regra de produção $A\Delta_A \rightarrow \chi\Gamma_\chi\Delta_\chi$, a parte $A \rightarrow \chi$ é a mesma de uma gramática de *strings* convencional, enquanto Δ_A , Γ_χ e Δ_χ são partes semânticas. A linguagem gerada pela parte sintática de uma gramática *plex* tridimensional ainda é livre de contexto.

Para a imagem da Figura 22, um exemplo de gramática *plex* tridimensional que poderia ser utilizada para descrevê-la é:

- $G_p = (N, \sum, P, S, I, i_0)$
- $N = \{S, A, B, C\}$
- $\sum = \{a, b, c, d\}$

- $S = S$
- $I = \{0, 1, 2, 3, 4\}$
- $I_0 = 0$
- P é o conjunto das regras de produção:
 1. $S() \rightarrow aA(12)()$
 2. $A(3) \rightarrow dB(21, 32, 43)(21, 32, 43)$
 3. $A(3) \rightarrow d()(2, 3, 4)$
 4. $B(3) \rightarrow cbcC(3400, 0230, 2003, 0302, 0021)(0040, 0100, 400)$
 5. $C(3) \rightarrow aA(12)(01, 12, 03)$

Figura 22 – (a) Objeto geométrico. (b) Decomposição do objeto em suas partes. Fonte: [59].

A quarta regra de produção, por exemplo, indica que B pode ser construído por meio da junção de c , b , c e C . Neste caso, $\chi = cbcC$, $\Delta_B = (1, 2, 3)$, $\Delta_\chi = (0040, 0100, 4000)$ e $\Gamma_\chi = (3400, 0230, 2003, 0302, 0021)$. Δ_B indica que B está conectado ao resto do *plex* pelos *tie-curves* 1, 2 e 3. O primeiro campo de Δ_χ , 0040, indica que o *tie-curve* 4 do segundo c em χ se conecta ao resto do *plex*, enquanto que os outros componentes em χ não estão envolvidos. A conexão é feita na curva correspondente ao *tie-curve* 1 de B , como indicado pela primeira entrada em Δ_B . Isto é, o *tie-curve* 1 de B corresponde ao *tie-curve* 4 do segundo c em χ . Os outros campos de Δ_χ podem ser interpretados

de forma similar. O primeiro campo de Γ_χ , 3400, indica que o *tie-curve* 3 do primeiro componente em χ , primeiro c , se conecta ao *tie-curve* 4 do segundo componente em χ , ou seja, a b . Os outros campos de Γ_χ podem ser interpretados da mesma forma [59].

Como proposta para reconhecimento de objetos tridimensionais um novo sistema foi apresentado em [59]. O sistema criado é formado por duas partes principais: análise e reconhecimento. A análise consiste da seleção de primitivas (fragmentos de superfícies) e de uma gramática *plex* tridimensional. A parte do reconhecimento é responsável pelo pré-processamento da imagem, segmentação, identificação das primitivas e uma análise estrutural. Durante o reconhecimento, uma imagem é capturada e armazenada no computador. O próximo passo é realizar a segmentação desta imagem, para que o sistema possa identificar suas partes e gerar uma matriz de transformação. Esta informação é então, enviada para um analisador de estrutura que decidirá se o padrão encontrado na imagem pertence às classes de padrões que são projeções do objeto modelo. Enquanto realiza a análise estrutural, o analisador produz uma descrição completa da estrutura em termos de análise sintática mais à esquerda para um reconhecimento com sucesso, caso contrário, o padrão é rejeitado.

4.3.1.11 Abordagem Sintático-Estrutural

[60] apresentam uma proposta baseada na combinação de métodos estatísticos e estruturais para reconhecimento de caracteres chineses. Na abordagem estrutural utilizada, um caractere chinês é representado por meio de uma estrutura hierárquica, na qual as primitivas são formadas pelos traços utilizados nos desenhos. A parte estatística do arcabouço proposto é responsável pelas comparações entre os traços da imagem que está sendo analisada no momento com as informações obtidas dos exemplos de treinamentos. O protótipo construído foi treinado com 7920 amostras de caracteres chineses escritos a mão e exibe como resposta uma pontuação entre o intervalo $[0,1]$ para a imagem analisada contra todos os exemplos do banco de dados.

4.3.1.12 Shock Grammar

É chamado de esqueleto a representação 2D (bidimensional) de um objeto que preserva o contorno de um objeto. Esses esqueletos podem ser convertidos em *shock graph*, que é uma abstração da forma utilizada para agrupar pontos do esqueleto de acordo com a variação de uma função do raio. Estes grupos são chamados *shock groups* e são rotulados de 1 a 4. Cada agrupamento provê a decomposição do esqueleto em partes e a inter-

relação é definida por uma gramática bem definida. Um *shock graph* é obtido a partir dos *shock groups* e são grafos direcionados e acíclicos [61].

Shock graphs, um método baseado em esqueletos, foi utilizado em [62] para reconhecimento de objetos. O método utiliza pontos de ramificações (que têm 3 ou mais pontos vizinhos na vizinhança de 8 pixels) em *shock graphs* e *shock grammars* para melhorar a taxa de reconhecimento e a eficiência no reconhecimento de objetos. Para obter o esqueleto de uma imagem foi utilizado o eixo medial. *Medial axis*, que é o ponto central do maior círculo contido dentro de uma curva que toca as fronteiras da forma. Os pontos de ramificação são os pontos básicos do esqueleto de uma forma. Geralmente, esses pontos são partes internas do esqueleto e são robustos quando consideramos deformações. Além disso, esses pontos são, geralmente, invariantes na presença de partes ocultas. O *matching* de árvores utilizado em [62] foi baseado tanto em sua topologia, quanto em informações geométricas para cada vértice e seus correspondentes pontos no esqueleto. Os testes foram executados para determinar a similaridade entre duas formas dadas. A similaridade é normalizada com valores entre $[0, 1]$, considerando que a maior similaridade produz valores próximos de zero.

No método proposto em [61] todo objeto na base de dados é convertido em uma imagem binária e seus esqueletos são obtidos e convertidos em *shock graphs*. O *shock graph* de uma imagem é comparado aos *shock graphs* das imagens no banco de dados. O algoritmo para comparar os *shock graphs* de duas imagens é o seguinte: (i) dividir o grafo de todos os objetos em subgrafos; (ii) comparar os subgrafos dos objetos com o subgrafo da imagem utilizada para realizar a busca; (iii) quando comparar as subárvores, compare o tipo do *shock* e se o tipo não for correspondente, considerar isso uma incompatibilidade. A pontuação obtida é a soma de todos os *matchings*.

Foi desenvolvida em [63] uma teoria para representação genérica de formas bidimensionais, na qual a descrição da estrutura é derivada a partir de *shocks* (pontos que colidem em um eixo) do processo de evolução de curvas agindo nos contornos. A abordagem utilizada para classificar um *shock* é baseada em propriedades diferenciais de uma *embedding surface*. A ideia era desenvolver uma implementação para obter estimativas geométricas precisas na vizinhança das descontinuidades para localizar os *shocks*. As restrições utilizadas para não permitir algumas configurações de *shock* incoerentes foram definidas por uma *shock grammar*. Como a gramática definida não descreve as restrições topológicas e geométricas, a gramática é incorporada em um grafo.

Em [64] os pesquisadores abordaram o problema do *matching* de formas organizando

os *shocks* em um *shock graph* direto e acíclico. O espaço dos grafos é caracterizado por regras de uma *shock graph grammar*, que permite a redução de um *shock graph* a uma única *shock tree*. Assim, os autores utilizaram um algoritmo de *matching* de árvores que encontra o melhor conjunto de nós correspondentes entre duas *shock trees* em tempo polinomial. Os pesquisadores demonstraram a performance do sistema desenvolvido utilizando uma base de dados com uma larga diversidade de formas considerando articulação, oclusão e mudanças de ponto de vista.

4.3.2 Reconhecimento de Texturas

Os trabalhos analisados nesta revisão que tiveram como foco o reconhecimento de padrões em texturas pertencem ao mesmo grupo de pesquisadores e fizeram uso de gramáticas *fuzzy*. Desta forma, antes de apresentá-los é necessário fazer uma definição formal deste tipo de gramática.

Uma gramática *fuzzy* é um modelo de classificador sintático que pode ser utilizado para representar os padrões de relações estruturais e descrever a sintaxe de uma linguagem *fuzzy*. Esta gramática é uma quintupla $GF = (V_N, V_T, P, S_0, \mu)$, em que V_N e V_T representam os símbolos não terminais e terminais, respectivamente; P é o conjunto de regras de produção do tipo $\alpha \rightarrow \beta$, em que $\alpha \in V_N$ e $\beta \in V^*$ ($V = V_N \cup V_T$); $S_0 \in V_N$ é o símbolo inicial e μ é um mapeamento de $P \rightarrow [0,1]$, tal que $\mu(p)$ denota a possibilidade da cadeia corrente $p \in P$ [27].

Foi apresentado em [65], em [66] e em [27] um sistema para monitoramento de peças de cortiça que utiliza uma abordagem de segmentação de textura baseada na transformada *wavelet* e uma gramática *fuzzy* como classificador. As características são extraídas das imagens processadas com a transformada *wavelet*, sendo esta técnica aplicada a cada componente da cor, de forma a ser realizada uma análise de textura baseada em cores. A abordagem desenvolvida é formada pelas fases de aprendizagem e de monitoramento. Na fase de aprendizagem, a textura é escolhida manualmente por meio da especificação de uma região de interesse na imagem. Um vetor de características é extraído e uma regra *fuzzy* para caracterizar a textura é utilizada para gerar a gramática *fuzzy*. Na fase de monitoramento, um vetor de características é extraído da imagem que está sendo analisada. Em seguida, é utilizada a gramática *fuzzy*, gerada na etapa de aprendizagem, para avaliar este novo vetor de características e realizar a classificação. Uma vantagem desta abordagem é que o sistema pode ser treinado com apenas uma amostra da textura. Além disso, o sistema pode ser utilizado em ambientes com diversos tipos de texturas,

mesmo que as condições de iluminação não sejam estáveis. A viabilidade e a eficiência da abordagem foram comprovadas em experimentos com mais de 30 tipos de texturas extraídas de imagens com 640x480 pixels. O sistema fornece como resposta um valor maior que 0.85 para a regra que descreve a textura utilizada como entrada e uma resposta inferior a 0.3 para regras que correspondem a outras texturas. Quanto maior esse valor, maior a possibilidade da textura ser descrita pela regra utilizada.

4.3.3 Construção de Objetos

4.3.3.1 Gramática de Atributos

O trabalho [67] apresenta um estudo para percepção de objetos tridimensionais (3D) a partir de esboços bidimensionais (2D) manualmente anotados de uma imagem. A gramática de atributos utilizada é definida como uma quádrupla $G = (V_N, V_T, P_R, R)$. V_N e V_T são os conjuntos de símbolos não-terminais (aspectos 2D/partes 3D) e nós terminais (elementos 2D), respectivamente. P_R é o conjunto de regras de produção utilizadas para formar pirâmides, cubos, prismas etc. R é o conjunto de relações existentes entre cada par de símbolos não terminais. A abordagem utilizada é composta de duas camadas, uma 2D e outra 3D, que formam um modelo gramatical hierárquico para representar objetos criados pelo homem. Na camada 2D, dada uma imagem de entrada (esboço 2D manualmente anotado), o modelo agrupa os elementos geométricos (triângulos, retângulos etc.) formando vários aspectos 2D por meio de um algoritmo *top-down/bottom-up*, sendo que cada um desses aspectos é utilizado para fazer inferência sobre estruturas ocultas da parte 3D. Essas duas camadas são calculadas recursivamente em um arcabouço bayesiano utilizando *data-driven Markov Chain Monte Carlo*. Os testes foram realizados com oito tipos de objetos diferentes com 20 imagens para cada categoria. Para cada categoria de objeto modelado pelo sistema foi criado outro objeto em um *Computer Aided Design* 3D. Todos os modelos foram misturados e em seguida foi pedido para que as pessoas apontassem qual objeto havia sido criado pelo método proposto. Esperava-se uma taxa de acerto de 50%, pois a ideia é que as pessoas não conseguissem diferenciar um objeto criado com a técnica apresentada de um objeto criado com um CAD, sendo que a apurada foi de 52,85% - muito próxima da esperada.

4.3.3.2 Gramáticas Matriciais

[68] apresentam um sistema para gerar figuras de matrizes retangulares utilizando um sistema chamado *Splicing Array Grammar Systems* (SAGS). Neste trabalho, encontram-se as especificações formais de uma *2D tabled matrix grammar*, das operações que podem ser realizadas com essas matrizes (operações sobre as linhas e as colunas) e do próprio sistema SAGS. É exibida na Figura 23 duas imagens geradas pelo modelo em questão.

Figura 23 – Figuras geradas pelo sistema SAGS. Fonte: [68].

4.3.3.3 L-System

Primeiramente introduzido como uma teoria matemática para explicar o crescimento e desenvolvimento de organismos multicelular por Aristid Lindenmayer, L-systems desde então proporcionou um grande impacto nas áreas de simulação e modelagem [8].

Para apresentação dos próximos trabalhos desta seção, deve-se, primeiramente, introduzir a noção de gramáticas paralelas e de *L-System*.

Gramáticas Paralelas

Uma determinada gramática $G = (V_N, T_N, R, S)$ aceita uma *string* x se existe uma sequência de passos de derivação $S \rightarrow w_0 \rightarrow w_1 \rightarrow \dots \rightarrow w_k = x$, na qual cada passo de derivação ($\rightarrow$) denota a aplicação de uma regra de produção. No caso de gramáticas paralelas, esta restrição é modificada. Dada uma gramática paralela $G|| = (V_N, T_N, R, S)$, dizemos que esta gramática aceita a string x , se existe uma sequência de derivação paralela $S \Rightarrow w_0 \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_k = x$, na qual cada passo de derivação ($\Rightarrow$) denota a aplicação de todas as possíveis regras para uma *string* intermediária.

As linguagens aceitas pela gramática $G||$ e pela correspondente G não é a mesma, exceto para casos simples. Por exemplo:

$$L(G) = \{a^n | n \geq 1\}$$

$$L(G) = \{a^{2^n} | n \geq 1\}$$

Este tipo de gramática tem sido muito utilizado em aplicações de computação gráfica na forma de *L-system* [8].

L-System

De acordo com [8], este tipo de sistema nasceu de uma teoria matemática que explicava o crescimento e desenvolvimento de organismos multicelulares e teve grande impacto em áreas de simulação e modelagem.

Trata-se de gramáticas formais que podem gerar *strings* e que são, geralmente, gramáticas sensíveis ao contexto e paralelas. As regras de produção e os símbolos têm sua própria semântica e interpretações visuais baseadas nos problemas sendo modelados.

A Figura 24 ilustra a aplicação de *L-System* para modelagem de plantas e de construções.

Figura 24 – Aplicação de *L-System* para modelagem de plantas e de construções. [8].

Um método para fazer inferência sobre a estrutura biológica a partir de imagens obtidas por microscópios utilizando modelos baseados em gramáticas que representam o crescimento biológico foi apresentado em [28]. Como muitas estruturas biológicas podem ser descritas de forma recursiva, é possível descrever este crescimento por meio de regras de produções gramaticais similares às do *L-System*. Como fazer a inferência de tal modelo é difícil devido à quantidade e à independência de parâmetros, foi criado um verificador de amostras baseado em *Markov Chain Monte Carlo* para explorar o espaço dos parâmetros em busca de um conjunto de parâmetros que provavelmente gerou os dados. Para testar o método proposto foram utilizadas 184 imagens do fungo *Alternaria* de dois conjuntos de dados. O verificador de amostras foi executado a partir de 10 estados iniciais aleatórios nos dois conjuntos, cada um com 20.000 iterações. Alguns resultados podem ser vistos na Figura 25.

Gramáticas do tipo *L-System* também foram utilizadas em [69]. Este trabalho apre-

Figura 25 – Resultados da aplicação do modelo ao conjunto $A1$ e $A2$. (a) Um elemento de $A1$, (b)-(e) modelos inferidos. (f) Um elemento de $A2$, (g)-(j) modelos inferidos. Fonte [28].

senta um método de modelagem para simulação do desenvolvimento e crescimento de plantas. A abordagem utilizada possui duas características principais. A primeira é enfatizar o relacionamento tempo-espaco entre as partes que constituem uma planta. Por exemplo, em várias espécies, alguns estágios do desenvolvimento podem ocorrer simultaneamente. A segunda característica é a capacidade de simular o crescimento da planta, pois a abordagem proposta é capaz de representar uma planta com diferentes idades. Para conseguir reproduzir estas duas características, o método proposto fez uso de gramáticas do tipo *L-Systems*, pois estas possuem regras de produção recursivas são capazes de simular o desenvolvimento de plantas. Alguns dos modelos de plantas gerados podem ser vistos na Figura 26.

Figura 26 – Resultados da aplicação do modelo para simular o desenvolvimento de plantas. Fonte [69].

Em [70] é apresentado um método que combina técnicas baseadas em regras e em imagens para criar modelos 3D de árvores leves (fáceis de serem renderizadas) para serem utilizadas na web. O modelo proposto pode ser dividido em 4 passos: 1) recuperação da estrutura de um tronco de árvore a partir de uma imagem 2D; 2) reconstrução 3D do esqueleto do tronco utilizando método de visão binocular; 3) Extração do axioma e das regras de produção a partir do esqueleto 3D utilizando gramáticas *L-Systems*; 4) utilização

de um algoritmo de interpretação para criar modelos que podem ser renderizados na máquina do cliente via web. Exemplos podem ser vistos na Figura 27.

Figura 27 – Exemplos de árvores criados com o modelo e seus tamanhos: (a) 1.56K; (b) 1.11K; (c) 1.18K; (d) 0.87K. Fonte [70].

O trabalho de [71] apresenta uma extensão da gramática utilizada no sistema Genr8 que é utilizado por arquitetos para elaborar *design* de superfícies. Uma superfície no Genr8 começa com um polígono fechado e cresce, de forma generativa, por meio de aplicações simultâneas das regras de produção. Durante o processo de crescimento, a superfície sofre influência de fatores ‘externos’, tais como repelentes, atratores e fronteiras. O Genr8 utiliza uma gramática chamada *Hemberg Extended Map L-System* que é uma forma mais complexa da gramática *L-System*. É possível observar na Figura 28 um exemplo de superfície criada pelo sistema desenvolvido.

Figura 28 – Modelo físico baseado na superfície Genr8. Fonte [71].

Foi apresentado em [72] um método de aprendizado de regras gramaticais visando à criação de uma gramática *L-System* para modelar plantas. No passo inicial, técnicas de processamento de imagens e reconhecimento de padrões foram utilizadas para recuperar as estruturas geométricas e morfológicas da planta. Em seguida, os dados obtidos no passo anterior são analisados por uma *hidden Markov tree* e uma *semi-Markov chain* para compor os parâmetros para o autômato hierárquico bidimensional (BHA - *Bidimensional*

Hierarchical Automata) que descreve a estrutura da ramificação da planta. O último passo é criar uma gramática *L-System* a partir da transformação do BHA. Para testar o método proposto dois experimentos foram conduzidos utilizando 35 imagens de árvores sem a folhagem com uma cobertura de 360 graus ao redor da planta. Uma das limitações da técnica é o alto custo para capturar a estrutura da árvore a partir de várias imagens. Além disso, não é possível capturar a estrutura de árvores com folhas devido à oclusão dos ramos.

4.3.4 Segmentação de Imagens

O trabalho de [29] apresenta um método estocástico para segmentação de imagens. A abordagem utilizada decompõe uma imagem em seus componentes visuais e retorna como saída uma representação hierárquica em forma de grafo AND-OR. Inicialmente é formulado o problema de inferência bayesiana, e o espaço de soluções é decomposto em união de vários subespaços de dimensões variadas, sendo que o objetivo é ‘otimizar’ a probabilidade *a posteriori*. Em seguida, uma abordagem *top-down* é utilizada para descrever como os objetos e os modelos de regiões (textura, sombreamento etc) geram a intensidade da imagem. Por último, para calcular as estimativas dos parâmetros são utilizadas propostas *bottom-up* como guia de busca através do espaço de parâmetros. Os resultados obtidos podem ser vistos na Figura 29.

Figura 29 – Resultados da segmentação de imagens. Fonte: [29].

Foi apresentado em [73] um método para segmentar uma dada imagem e estimar a área utilizando gramáticas regulares. A imagem é dividida em pequenos blocos, cada um tendo uma estrutura pré-definida que representa o nó terminal de uma gramática. Em seguida, cada bloco é associado a um número que faz o mapeamento da estrutura

representada de tal forma a criar uma matriz numérica que representa a imagem. O próximo passo é criar uma expressão regular para representar um contorno fechado. O autômato, equivalente a uma expressão regular, é utilizado para analisar a matriz em buscas por submatrizes com os contornos definidos realizando a segmentação da imagem. Os testes foram conduzidos em imagens cardíacas e foi obtida uma precisão de 93.22% na estimação das áreas segmentadas.

4.3.5 Mudança de Escalas

É apresentado em [30] um estudo sobre percepção de escala de espaço por meio do aumento de escala de uma imagem tradicional. A abordagem utilizada faz uso da representação de esboços de primitivas que dividem a imagem em partes estruturais (*sketchable*) para delimitar a fronteira dos objetos e partes de texturas (*non-sketch*). A abordagem também utiliza um tipo de grafo gramatical chamado *sketch pyramid* para representar mudanças topológicas. Para a construção desses grafos, o método proposto faz uso de um arcabouço bayesiano e utiliza *Markov chain Monte Carlo reversible jumps*. A técnica criada foi utilizada com duas propostas diferentes. A primeira, *multi-scale object tracking*, é utilizada para fazer a monitoração de objetos em uma cena, como pode ser visto na Figura 30. A segunda proposta é chamada *adaptive image display* e ataca o problema de exibir imagens de alta resolução em displays de baixa resolução (celulares, PDAs etc.) e faz uso de *quadtrees*, como pode ser visto na Figura 31.

Figura 30 – Sequência de rastreamento de um carro. (a) imagens originais. (b) esboços iniciais obtidos pelo algoritmo bottom-up. (c) esboços finais obtidos a partir de b. Os traços do carro estão em preto, enquanto que o do fundo está em cinza. Fonte [30].

Figura 31 – Resultado da técnica *adaptive image display*. Fonte [30].

4.3.6 Reconhecimento de *Layout*

4.3.6.1 Gramáticas Regulares

Foi utilizado em [31] gramáticas estocásticas regulares para representar o *layout* de documentos como uma estrutura hierárquica, sendo que cada nó desta hierarquia descrevia uma região do documento. A hierarquia e a linguagem são definidas pelo usuário, enquanto que os parâmetros de probabilidade são estimados a partir de amostras de treinamento. A estrutura física do *layout* do documento é reconhecida por meio de um algoritmo de segmentação baseado em análise sintática.

4.3.6.2 Gramáticas Livres de Contexto

Uma abordagem para segmentação hierárquica e reconhecimento de *layout* da estrutura de documentos utilizando gramáticas estocásticas livres de contexto foi apresentado em [74]. Aprendizado de máquina é utilizado para selecionar as características e todos os parâmetros utilizados no processo de análise sintática. Programação dinâmica também foi utilizada para encontrar a melhor árvore de derivação para uma página. Para testar a abordagem dois estudos de caso foram utilizados. No primeiro, o arcabouço gerado utiliza uma gramática criada manualmente para analisar o *layout* do documento, sendo obtida a *F-measure*⁴ de 85%. No segundo caso, o arcabouço foi utilizado para inferir a gramática para reconhecimento de equações matemáticas encontradas no documento e a *F-measure* encontrada foi de 94%.

⁴F-measure é uma medida da média harmônica da precisão.

4.3.6.3 Gramáticas Sensíveis ao Contexto

Foram utilizados grafos gramaticais espaciais sensíveis ao contexto em [75] para recuperar a estrutura semântica presente em uma interface web. A gramática executa o agrupamento semântico e a interpretação de objetos segmentados a partir do *screenshot* de uma tela. Um grafo espacial é construído no qual cada nó representa um objeto reconhecido e as arestas representam as relações espaciais entre os objetos (textos, botões etc). No próximo passo, uma análise sintática é executada para descobrir a estrutura hierárquica envolvendo estes objetos. A abordagem foi testada em muitas páginas web e o reconhecimento do objeto ocorreu de forma satisfatória em menos de 25 milissegundos. A abordagem é menos complexa, considerando a quantidade de nós gerados, que uma análise baseada no código fonte HTML.

É apresentado em [76] uma técnica que visa a detectar títulos de artigos técnicos. Para tanto, foram utilizados *semi-Markov models* e uma gramática estocástica (*stochastic attributed K-d tree grammar*). Os autores consideraram a análise do documento como uma análise sintática, pois as ordens dos componentes em um documento e suas relações podem ser modeladas com uma gramática que descreva a página nos níveis dos componentes em termos de blocos e regiões. O método proposto foi dividido em duas partes: (i) para descrever o agrupamento das regiões das páginas em blocos retangulares foi utilizado *semi-Markov model*; (ii) para descrever a decomposição hierárquica da página foi utilizado a *K-d tree grammar*. Inicialmente, a imagem do documento é dividida em faixas e então o número de pixels pretos em cada faixa é contado. Em seguida, a sequência de pixels contados é utilizada no *hidden semi-Markov model*. As fronteiras entre diferentes grupos de faixas são indicadas pelas mudanças de estado do modelo, que rotula os grupos de faixas. O vocabulário de símbolos da gramática proposta é formado por esses rótulos e é utilizado para encontrar o *layout* físico da página.

4.4 Discussão

Conforme pode ser visto na Figura 9 os métodos/técnicas mais utilizados nesta área de pesquisa são gramáticas estocásticas, grafos AND-OR, Gramática e lógica *fuzzy*, *shock grammars* e *L-System*.

Gramáticas estocásticas foram bastante utilizadas, sejam elas as tradicionais livres de contexto ou novas formulações como *Spatial Random Tree Grammars*, *L-systems*, ou baseadas em grafos AND-OR. Grafos AND-OR foram utilizados por vários grupos de pesquisa

e ele foi útil tanto em tarefas de reconhecimento de objetos [3], como para segmentação de imagens, embora apenas um único artigo tenha feito uso dessa estrutura para este último propósito [29]. *Spatial Random Tree* também foi utilizada para reconhecimento de padrões em imagens. Embora a técnica tenha sido utilizada por apenas um grupo de pesquisa ela alcançou uma taxa de reconhecimento de imagens de aproximadamente 97% [35]. *Shock grammars* e *shock graphs* apareceram em quatro estudos. Essas técnicas foram utilizadas para reconhecer formas de objetos presentes em imagens binárias [64].

Gramática e lógica *fuzzy* apareceram em 4 trabalhos. Um desses trabalhos [46] era para reconhecimento de objetos em imagens, os outros eram todos do mesmo grupo de pesquisa e lidaram com problemas de reconhecimento de texturas, [27]. É importante notar que todos os trabalhos de reconhecimento de textura utilizaram gramáticas e lógica *fuzzy*.

Para construção de objetos, notadamente objetos que podem ser construídos por meio de regras recursivas, a principal gramática utilizada foi do tipo *L-System*, por exemplo, [69].

É possível notar que parece haver uma distribuição desigual dos artigos com relação à sua aplicação no campo de visão computacional. A maioria das aplicações está concentrada em problemas de reconhecimento de objetos (33 artigos dos 50), construção de objetos (7 artigos), reconhecimento de layout (4 artigos) e segmentação de imagens (2 artigos). Para cada uma das seguintes aplicações presentes nesta revisão (reconhecimento de textura e mudança de escala), apenas um estudo foi desenvolvido⁵.

Podemos notar que a maioria dos trabalhos não faz inferência da gramática propriamente dita, mas sim inferem as distribuições de probabilidades que são atribuídas a cada regra de produção quando a gramática é estocástica, por exemplo [33]. Os principais trabalhos que fazem inferência da gramática são [3] que apresenta a construção de grafos AND-OR, [47] que cria expressões regulares aumentadas com o auxílio da técnica *Active Grammatical Inference*, [65], [66] e [27] que constrói uma gramática *fuzzy* para reconhecimento de padrões em texturas, [70] e [72] que inferem uma gramática do tipo *L-System* para modelagem de árvores, [74] para reconhecimento do *layout* de documentos e [46] que fez uso de MHMM e ANFIS para aprender uma gramática.

É possível observar na Tabela 1 os principais trabalhos analisados durante a revisão bibliográfica.

⁵Os três artigos que lidaram com reconhecimento de texturas utilizaram a mesma abordagem e foram desenvolvidos pelo mesmo grupo de pesquisa

Tabela 1 – Principais trabalhos levantados durante a revisão sistemática

Artigos	Gramáticas es- tocásticas	Grafos AND-OR	Spatial Random Tree Grammar	L-System	Lógica Fuzzy	Gramáticas Livres de Contexto	Shock Grammars
[35]	X		X				
[3]	X	X					
[54]						X	
[69]				X			
[27]					X		
[39]	X	X					
[64]							X

Embora vários trabalhos tenham utilizado métodos semelhantes, os que constam na Tabela 1 foram selecionados por apresentarem mais detalhes sobre as técnicas utilizadas e que serviram de inspiração para este trabalho de mestrado. Além disso, todos esses trabalhos apresentam os resultados da técnica desenvolvida, seja de forma quantitativa ou qualitativa.

A partir da análise conduzida podemos observar as vantagens e desvantagens da utilização de gramáticas em Visão Computacional. Desta forma, algumas lacunas foram percebidas na literatura analisada que podem mostrar algumas direções e oportunidades para novas linhas de pesquisa.

4.4.1 Vantagens e Aplicações de Gramáticas em Visão Computacional

A primeira vantagem é uma representação mais concisa que as gramáticas em geral podem prover em comparação com a representação por conjunto inerente de pixels das imagens. A abordagem sintática pode representar padrões estruturais de forma eficiente e, consequentemente, facilitar a recuperação da imagem pela sua estrutura. Desde 1986 foi observado em [59] que as abordagens sintáticas para classificação de padrões e análise de cenas mereciam a atenção dos pesquisadores devido a sua habilidade em lidar com padrões de estruturas e seus relacionamentos. A abordagem sintática é capaz de descrever um largo conjunto de padrões complexos utilizando pequenos conjuntos de primitivas simples e regras gramaticais.

Quando as imagens são percebidas como cenas estruturadas como uma ordem hierárquica entre seus objetos, sua representação por gramáticas é ainda mais intuitiva. Isso acontece devido ao fato de que gramáticas podem, naturalmente, descrever estruturas hierárquicas por meio de seus símbolos terminais e não terminais. Esta representação natural foi enfatizada em [58], que adicionalmente percebeu que o uso de uma abordagem gramatical em imagens (reconhecimento de objetos) incluía facilidades para recuperação de erros.

Além de outros métodos para reconhecimento de padrões, o pesquisador interessado em utilizar abordagens gramaticais tem muitas opções de modelos a escolher, cada um com seu poder de representação diferenciado, mas também com uma complexidade diferente. Gramáticas, considerando os níveis da hierarquia de Chomsky, podem ser escolhidas para obter os resultados esperados da maneira mais eficiente.

A maioria das aplicações em Visão Computacional requer gramáticas pelo menos livres de contexto. Entretanto, dependendo do objetivo da aplicação, a maior eficiência de gramáticas regulares pode ser utilizada (tempo linear na análise sintática). Por exemplo, em [31] foi utilizada uma gramática regular estocástica para descrever o *layout* estrutural de documentos e em [73] foram utilizadas gramáticas regulares para segmentação de imagens. Outra vantagem da utilização deste tipo de gramática é a disponibilidade de algoritmos eficientes para as regras gramaticais, permitindo o aprendizado supervisionado dos modelos utilizados, como apresentado em [31].

Gramáticas livres de contexto demandam tempo polinomial para os algoritmos de análise sintática, mas são capazes de caracterizar as dependências entre os componentes arbitrariamente distantes em estruturas aninhadas ou ramificadas. Tal característica

permite a este tipo de gramática representar padrões estruturais mais ricos e consequentemente de ser utilizado em uma variedade mais ampla de aplicações em Visão Computacional. Por exemplo, gramáticas livres de contexto podem representar diferentes lesões em imagens médicas por meio de regras que utilizam intervalos dos ângulos entre os ossos, veias, etc ([53], [54]). Alguns algoritmos de aprendizado de gramáticas livres de contexto são apresentados em ([13], [15], [14]), permitindo a criação de modelos a partir de amostras de treinamento.

Embora gramáticas regulares e livres de contexto possam ser utilizadas, o poder de representações adicionais pode ser necessário para melhor caracterizar padrões visuais complexos. Particularmente, algumas informações contextuais são importantes para representar relacionamento espacial entre partes da imagem. Desta forma, a maior parte dos trabalhos reportou sucesso em reconhecimento de objetos complexos utilizando essas características. *Spatial Random Tree Grammars* ampliou gramáticas estocásticas livres de contexto incorporando informações espaciais como parte da gramática ([77], [33], [35]). Alternativamente, gramáticas estocásticas livres de contexto podem ser combinadas com *Markov Random Field* para representar a variabilidade da configuração de um objeto e o último representando o relacionamento espacial entre as partes ([38],[3]). O relacionamento espacial também pode ser representado utilizando grafos gramaticais de atributos ([26], [39], [75], [51], [52], [54]) ou gramáticas estocásticas sensíveis ao contexto ([78], [44]). Entretanto, o aprendizado de regras gramaticais e a análise sintática são problemáticos no mundo sensível ao contexto. Felizmente, as construções propostas nesses trabalhos são capazes de reconhecer um subconjunto próprio de linguagens sensíveis ao contexto mantendo o tempo polinomial da análise sintática. Os geradores de analisadores sintáticos para esses modelos gramaticais seriam muito úteis para novos trabalhos utilizando essas estratégias. O aprendizado de regras gramaticais para esses modelos propostos, entretanto, é abordado apenas em [3], no qual a gramática combinada com *Markov Random Field* é livre de contexto.

Formulações alternativas de gramáticas também podem ser exploradas. Gramáticas *fuzzy* foram utilizadas no reconhecimento de texturas e gramáticas paralelas (*L-System*) na construção de objetos. Gramáticas *L-System* podem facilmente representar padrões de fractais. Portanto, este modelo foi amplamente utilizado para construir imagens tais como plantas, árvores etc., favorecendo a definição de modelos consistentes a aqueles utilizados no desenvolvimento morfológico e fisiológico [69].

4.4.2 Desvantagens da Utilização de Gramáticas em Visão Computacional

Apesar da representação concisa provida pelas representações sintáticas, é importante mencionar algumas limitações da utilização de gramáticas no contexto de imagens. Embora exista uma economia de espaço na representação, o custo computacional do processamento pode ser alto. Como pode ser visto nesta revisão, existem vários métodos, dezenas de objetos e muitos tipos possíveis de representações. Padrões mais complexos frequentemente requerem gramáticas com um maior poder de representação e, portanto, com uma maior complexidade de tempo no reconhecimento. Quando o método requer uma fase de treinamento, o processamento pode demandar um alto custo adicional de computação, dependendo do tamanho do conjunto das imagens e do tipo da gramática, entre outros fatores.

Uma segunda limitação percebida é a dificuldade de processar imagens complexas. Aqui estamos considerando imagens complexas aquelas com alta resolução e/ou com alto nível de detalhes e possivelmente incluindo oclusões. Muitos estudos apresentados nesta revisão apontaram esta limitação. Por exemplo, quando utilizado um algoritmo para gerar símbolos terminais para representar grafos EDT, foi citado em [57] que o sistema desenvolvido poderia ser incapaz de processar objetos complexos. O autor considera objetos complexos aqueles no qual a distribuição dos componentes espaciais não é descritiva o suficiente e/ou existe um grande número de partes visualmente similares.

Finalmente, outra importante limitação a ser observada é a alta dependência de técnicas de pré-processamento de imagens que as gramáticas podem apresentar. Alguns trabalhos mencionados nesta revisão utilizaram técnicas para executar operação para enriquecer estruturas, como para suavizar ruídos e segmentar estruturas. Estas técnicas são bem conhecidas na literatura de Visão Computacional como altamente dependente do tipo da imagem e dos objetivos do processamento. Apenas alguns autores discutiram esse tema, mas falhas nesses processos podem levar a falhas na utilização das gramáticas, particularmente quando o objetivo é estabelecer procedimentos para armazenar e recuperar características para classificar imagens em categorias.

4.4.3 Breve Comparação com Técnicas Não-Gramaticais

Uma vez apresentadas as vantagens e as desvantagens do uso de gramáticas em geral é importante tecer uma comparação com outros métodos não-gramaticais.

Em geral, a utilização de gramáticas permite uma forma mais flexível de representar imagens principalmente quando as imagens possuem um padrão hierárquico bem estabelecido, como mencionado anteriormente. Isso ocorre porque é possível fazer um mapeamento direto entre a estrutura da imagem e as regras sintáticas. Sobre este assunto, foi mencionado em [60] que a utilização de características estruturais das imagens representadas de uma forma sintática, quando comparadas com métodos estatísticos, provê uma maior flexibilidade em algumas aplicações reais, como extração de padrões de imagens deformáveis a partir de fundos (*backgrounds*) complexos. No contexto do reconhecimento de objetos tridimensionais, foi citado em [59] que é mais fácil identificar os *patches* de superfícies utilizando primitivas visíveis em uma representação sintática do que reconhecer o objeto diretamente.

Ainda, particularmente gramáticas estocásticas provêm a probabilidade de um dado objeto pertencer a uma classe considerada, diferentemente de outros métodos discriminantes, como árvores de decisão, redes neurais, máquinas de vetores de suporte ou mesmo gramáticas determinísticas, que fornecem apenas o resultado da classificação [32].

4.4.4 Desafios e Possíveis Direções Futuras

A partir das considerações acima, é possível mostrar algumas direções que ainda permanecem abertas na literatura investigada.

Primeiramente, o estudo de imagens complexas ainda é um desafio a ser superado. Modelagem, representações, segmentações e reconhecimento de objetos complexos constituem linhas de pesquisa que podem ser exploradas. Uma das principais categorias desses objetos é composta por imagens médicas. Nos últimos anos sistemas de diagnósticos auxiliados por computadores foram propostos na literatura utilizando imagens como a principal fonte de dados. Alguns exemplos são [79] e [80]. Embora alguns desses sistemas já estejam em uso em algumas rotinas clínicas, a utilização de gramáticas neste contexto é pouco explorada. Gramáticas poderiam prover uma forma flexível para ajudar na identificação e classificação de doenças, bem como ajudar a diferenciar casos normais dos anormais. Algumas iniciativas foram incluídas nesta revisão ([53], [73]), mas ainda existem muitas questões a serem superadas. A natureza complexa de imagens médicas requer uma profunda compreensão da imagem médica considerando os princípios da modalidade, a doença alvo e os objetivos do processamento e, por esta razão, elas constituem um rico campo de pesquisa.

Outro assunto a ser investigado é o desenvolvimento de uma recuperação rápida de

estruturas sintáticas. Na área de recuperação de imagens baseado em conteúdo, métodos de indexação e recuperação rápidos estão sendo investigados para diminuir o tempo de buscas por imagens ([81],[82]). Similarmente, estruturas mais eficiente para indexação e armazenamento de representações sintáticas podem ser investigadas e, portanto, contribuir para a eficácia da utilização de gramáticas para armazenamento e recuperação de imagens.

A combinação de métodos gramaticais com outras abordagens foi lucrativa em alguns artigos analisados. Como citado anteriormente, gramáticas estocásticas livres de contexto combinadas com *Markov Random Fields* levaram a uma caracterização espacial flexível ([38], [3]). Gramáticas *fuzzy* e transformadas *Wavelet* foram utilizadas no reconhecimento de padrões em textura, na qual os símbolos terminais da gramática eram vetores de características extraídos das imagens após a aplicação da transformada *Wavelet* ([65], [66], [27]). Estratégia similar poderia ser experimentada utilizando *Spatial Random Trees*, desde que o arcabouço admitisse vetores de características como símbolos terminais [35]. Outras combinações de métodos diferentes poderiam ser exploradas para compensar as limitações uns dos outros.

A aprendizagem de regras gramaticais poderia ser mais explorada para criar automaticamente um modelo de imagem completa a partir de amostras de treinamento. Aprendizagem gramatical é uma área de pesquisa ativa, uma vez que ainda há vários problemas abertos e computacionalmente difíceis [83]. O aprendizado de gramáticas dos níveis mais altos da hierarquia de Chomsky apresentam mais desafios que as gramáticas de níveis mais baixos. Por exemplo, enquanto que existem algoritmos conhecidos para aprendizado de autômatos finitos a partir de amostras positivas e negativas ([84], [85]), a classe inteira de gramáticas livres de contexto não é identificável a partir de apenas amostras positivas [10]. Algumas abordagens para aprendizado destas classes de gramáticas incluem o aprendizado de subclasses livres de contexto [15], aprendizado a partir de exemplos estruturados [14] e o uso de heurísticas [86]. Essas estratégias e outras similares poderiam ser propostas e usadas em aplicações de Visão Computacional. Por exemplo, foi proposto em [47] um método para aprendizado de expressões regulares aumentadas que é capaz de reconhecer um subconjunto próprio de linguagens sensíveis ao contexto.

Como mencionado anteriormente, todas as aplicações de *L-Systems* pesquisadas lidavam com o problema de construções de objetos. Como *L-Systems* são gramáticas paralelas, a análise sintática se torna um tópico importante [87]. Portanto, análises sintáticas avançadas em *L-Systems* são necessárias para aplicar esse modelo a problemas de reco-

nhecimento de objetos.

Finalmente, pode-se observar a pequena quantidade de estudos no domínio tridimensional. De fato, trabalhar neste contexto inclui algumas dificuldades adicionais como problemas de oclusão e necessidade de entendimento de perspectivas de objetos, entre outros. Entretanto, o avanço da tecnologia nos últimos anos tem propiciado um aumento de sistemas tridimensionais interativos em Computação Gráfica e Realidade Virtual em vários campos de aplicação. Logo, esse é um tópico que merece uma profunda exploração. Foi indicado em [69] uma dezena de direções relacionadas à utilização de *L-Systems* na modelagem de objetos gráficos como a adição de texturas, a modelagem de superfícies complexas e a análise de simulação de complexidade. Além dos tópicos apresentados, interação e sua influência na representação da cena podem constituir uma linha de pesquisa que merece investigação.

4.4.5 Conclusões

Dada a necessidade de encontrar artigos relevantes sobre a utilização de gramáticas no reconhecimento de padrões em imagens foi realizada a revisão sistemática sobre esta área de pesquisa. A leitura dos artigos selecionados para compor esta revisão permitiu obter uma visão abrangente sobre o uso de métodos sintáticos tanto para criação de objetos novos, geralmente modelagem de plantas, como para identificação e reconhecimento de objetos e de texturas em imagens, mudança de escala e segmentação. Como pode ser observada, esta é uma área de pesquisa bastante ativa, já que a maioria dos trabalhos encontrados foi publicada na última década, em especial nos últimos cinco anos. Além disso, pode-se notar que se trata de uma linha de pesquisa muito promissora, pois vários dos artigos estudados mostraram uma taxa de reconhecimento acima de 90%. Outro ponto a ser notado é que nenhum dos artigos encontrados utilizando as palavras chaves e os repositórios escolhidos para esta revisão foi publicado por pesquisadores brasileiros, o que pode significar que esta área de pesquisa talvez esteja sendo pouco explorada em nosso país. Também foi percebido que poucos trabalhos apresentaram métodos de inferência gramatical, o que pode ser visto como uma lacuna nesta área de pesquisa. Um método de inferência gramatical pode ser bastante útil quando aplicado no auxílio de diagnóstico médico, classificação de espécies relacionadas à biologia etc.

Capítulo 5

Métodos

5.1 Arquitetura Básica

Para que fosse feita uma avaliação da aplicação de inferência de gramáticas livres de contexto estocásticas utilizando *quadtrees* no reconhecimento de padrões em imagens, foi adaptado um arcabouço desenvolvido em [4].

O arcabouço utilizado foi desenvolvido utilizando a linguagem de programação C++ com o paradigma de programação orientada a objetos (POO), uma vez que a utilização deste tipo de linguagem permite uma maior reutilização de código, expansibilidade e grande flexibilidade de implementação.

A finalidade do arcabouço é facilitar a implementação de um gerador de classificadores baseados em gramáticas estocásticas. O processo de geração do classificador envolve a escolha e execução de módulos já implementados na atual versão do arcabouço: um módulo de processamento da imagem, de um algoritmo de inferência gramatical, de um algoritmo de estimação de probabilidades, de um gerador de analisador sintático e de um gerador de classificador. A arquitetura do sistema de geração e utilização de um classificador para n classes distintas pode ser vista na Figura 32. O módulo de processamento de imagem foi desenvolvido nesse projeto de mestrado e acoplado ao arcabouço já existente, uma vez que esse arcabouço havia sido originalmente desenvolvido para reconhecimento de padrões em sequências genéticas.

O módulo de processamento das imagens recebe como entrada um conjunto de imagens de treinamento e retorna como saída as *strings* parentisadas que representam as estruturas das imagens utilizando *quadtrees*. Na Figura 33, é apresentada uma imagem (esquerda), a estrutura de *quadtree* que representa a imagem (direita) e a *string* parentisada obtida a partir desta representação. No caso desta figura, o critério de parada adotado foi que a divisão da imagem cessasse assim que todos os pixels dos quadrantes fossem da mesma cor. No arcabouço construído este critério de parada é parametrizável, sendo possível estabelecer que a divisão cesse quando uma determinada porcentagem de pixels

Figura 32 – Arquitetura do arcabouço utilizado. O módulo de processamento de imagens, em azul, foi desenvolvido nesse projeto de mestrado.

dos quadrantes seja de valor preestabelecido.

Figura 33 – Estrutura de quadtree de uma imagem e a string parentisada equivalente.

O algoritmo de inferência gramatical recebe como entrada um conjunto com as *strings* parentisadas e retorna como saída uma gramática que descreve o conjunto de *strings* utilizado como entrada. Para realizar a inferência das regras de produção da gramática foi utilizada a estrutura de árvore representada na *string* parentisada. A Figura 34 exemplifica tal inferência.

Figura 34 – Estrutura de uma quadtree que será utilizada para inferência gramatical.

A gramática que representa exatamente a string (b(bbpp)(b(bppp)bp)(pp(pppb)b)) e, portanto, a estrutura da *quadtree* apresentada na Figura 34 é $G = (V_N, V_T, R, S)$, na qual:

- S representa o símbolo inicial da gramática.
- V_N é o conjunto de símbolos não-terminais composto pelos elementos $\{S, <0>, <1>, <2>, <3>, <4>\}$.
- V_T é o conjunto de símbolos terminais, neste caso, $\{b, p\}$
- R é o conjunto composto pelas seguintes regras de produção:

- $S \rightarrow (b \ A2 \ A3 \ A4) \ [1.0]$
- $\langle 0 \rangle \rightarrow (b \ b \ p \ p) \ [1.0]$
- $\langle 1 \rangle \rightarrow (b \ \langle 3 \rangle \ b \ p) \ [1.0]$
- $\langle 3 \rangle \rightarrow (b \ p \ p \ p) \ [1.0]$
- $\langle 2 \rangle \rightarrow (p \ p \ \langle 4 \rangle \ b) \ [1.0]$
- $\langle 4 \rangle \rightarrow (p \ p \ p \ b) \ [1.0]$

O processo de inferência gramatical envolve tanto a inferência das regras como a estimação de suas probabilidades. Para a inferência das regras gramaticais foram utilizados algoritmos de inferência de gramáticas reversíveis e inferência de gramáticas tipo-inversíveis descritos nas seções 2.1.4.2 e 2.1.4.3, respectivamente. Já para a estimação das probabilidades foi utilizado o algoritmo *Tree Grammar Reestimator* descrito na seção 2.1.5.2. Neste exemplo a gramática reconhece apenas, com probabilidade 1.0, a imagem à esquerda da Figura 33. Mas o objetivo do algoritmo de inferência gramatical é inferir uma gramática que reconheça não só todas as imagens da amostra de treinamento, mas que também generalize para outras imagens semelhantes.

O arcabouço é composto por classes abstratas para os inferidores gramaticais, estimadores das probabilidades, geradores de analisadores sintáticos e geradores de classificadores, e uma ou duas implementações (subclasses concretas) para cada uma dessas classes abstratas:

- duas implementações de inferidores gramaticais (os de gramáticas reversíveis e de gramáticas tipo-inversíveis, descritos nas seções 2.1.4.2 e 2.1.4.3);
- duas implementações de estimadores de probabilidades (o *Tree Grammar Reestimator* descrito na seção 2.1.5.2 e outro algoritmo descrito em [88]);
- uma implementação de um gerador de analisadores sintáticos, que geram um analisador de Earley estocástico para uma dada gramática [89]);
- e uma implementação de um gerador de classificadores que gera um classificador bayesiano (descrito na seção 2.2).

Uma vez selecionados os algoritmos de inferência gramatical e os estimadores de probabilidades, tem-se o necessário para gerar o classificador, testá-lo e obter os resultados.

5.2 Adaptações no arcabouço

Para este projeto, duas modificações foram realizadas no arcabouço já existente. A primeira foi a criação do módulo de processamento de imagens descrito na seção anterior. A segunda modificação foi o enriquecimento do conjunto de classes de *streams*.

A comunicação básica entre os algoritmos de aprendizado ocorre por meio de *streams*. Essa comunicação deve ser independente da implementação dos componentes e do formato da representação das sequências e das gramáticas. Para tanto, estava implementada uma arquitetura baseada em *streams* que tem como objetivo abstrair as origens, destinos, formatos e vias pelas quais os componentes se comunicam. A abstração **InputStream** é responsável pelos *streams* de entrada e pode ser vista na Figura 35, enquanto que a abstração **OutputStream** é responsável pelos *streams* de saída e pode ser vista na Figura 36.

Para abstrair a noção do fluxo de entrada de dados foi utilizado o **InputStream** que pode ler diferentes tipos e formatos de dados. Desta forma, para cada tipo de *stream* que se deseja manipular deve-se implementar uma classe que herde de **InputStream** ou de uma de suas subclasses. Para este projeto duas novas classes foram implementadas **MultiSentenceInputStream** e **PlainMultiSentence** para a leitura das sequências parentisadas que representavam as imagens. Essas novas classes podem ser vistas na Figura 35 em azul.

Figura 35 – Diagrama de classes de streams de entrada.

A hierarquia das classes dos dados de entrada são:

InputStream: classe abstrata utilizada para modelar uma fonte de dados genérica.

FileInputStream: Classe que implementa um stream de entrada no qual os dados são provenientes de um arquivo. Esta classe implementa vários métodos capazes de ler

diferentes tipos de dados, por exemplo, inteiros, float, double, char e *string*. O parâmetro de seu construtor é uma *string* com o nome do arquivo.

GrammarInputStream: Trata-se de uma classe abstrata utilizada para modelar um stream de entrada cujos dados representam uma gramática. Seus métodos permitem a leitura de produções, símbolos terminais e não terminais e probabilidade. Um parâmetro **InputStream** genérico deve ser passado para o seu construtor.

BNFGrammarInputStream: Esta classe implementa a classe abstrata **GrammarInputStream** e é capaz de ler gramáticas em formato BNF (*Backus-Naur Form*). O construtor recebe como parâmetro um objeto genérico **InputStream**.

MultiSentenceInputStream: É uma classe abstrata utilizada para modelar um *stream* de entrada cujos dados é um conjunto de sequências. As sequências lidas por um **MultiSentenceInputStream** são genéricas, ou seja, não pertencem a um alfabeto e nem possuem um formato específico. O formato de cada sequência (como delimitadores e informações adicionais) deve ser modelado nas subclasses. Seu construtor recebe como parâmetro um **InputStream** genérico.

PlainMultiSentence: É uma subclasse de **MultiSentenceInputStream** utilizada para leitura das sequências considerando que cada sequência, e apenas essa informação, está em uma linha separada do *stream* de entrada. Neste projeto essa classe foi utilizada para leitura das *strings* parentisadas obtidas a partir da estrutura de *quadtree* de uma imagem. Seu construtor deve receber como parâmetro um **InputStream** genérico.

As classes da hierarquia de dados de saída são:

OutputStream: Trata-se de uma classe abstrata que é a raiz da hierarquia dos *streams* de saída.

FileOutputStream: É implementado em **FileOutputStream** um *stream* no qual a saída ocorre em um arquivo. O construtor dessa classe recebe como parâmetro uma *string* com o nome do arquivo. Esta classe possui métodos para manipular vários tipos de dados, por exemplo, inteiro, float, double, char ou *string*.

A utilização desses *streams* torna possível a alteração dos canais e formatos dos dados sem que seja necessário alterar o código fonte dos componentes que os utilizam. Para tanto, basta escolher o *stream* que melhor se encaixa a uma determinada necessidade ou implementar um *stream* novo.

Atualmente, os canais de comunicação utilizados são os **FileInputStream** e o **File-**

Figura 36 – UML do relacionamento das classes de streams de saída.

OutputStream, pois o uso de arquivos proporciona uma solução simples e por fornecer o armazenamento em disco dos resultados intermediários.

Com relação à leitura de *streams* de gramáticas, foi utilizada a subclasse **BNFGrammarInputStream** para ler as gramáticas em formato BNF. Qualquer outro tipo de gramática poderia ser utilizada, bastando que a implementação do novo *stream* seja uma subclasse de **GrammarInputStream**.

Igualmente, pode-se criar mais especializações de **MultiSentenceInputStream**. Na alteração do arcabouço realizada apenas a subclasse **PlainMultiSentence** foi implementada para ler um conjunto de *strings* parentisadas, no qual cada *string* corresponde à estrutura de uma *quadtrees* de uma imagem.

5.3 Técnicas de avaliação do método proposto

O objetivo principal deste trabalho foi avaliar a utilização de gramáticas estocásticas inferidas a partir de amostras de treinamento que continham as estruturas de *quadtrees* das imagens no problema de classificação de imagens. Como uma primeira tentativa dessa aplicação, os classificadores foram treinados para reconhecer figuras geométricas (quadrados, círculos, triângulos e retângulos). Desta forma, dada uma imagem que possuísse uma figura geométrica, o algoritmo deve classificar essa imagem em uma das quatro classes previamente conhecidas.

Essa escolha se deu devido à simplicidade das imagens testadas em relação a objetos mais complexos e também à possibilidade de utilizar essa técnica, no futuro, no reconhecimento de padrões em imagens médicas, pois sabe-se que em alguns casos pode-se diferenciar tumores benignos de malignos por meio de sua estrutura geométrica.

Para validar a técnica proposta, vários testes foram realizados. Primeiramente as

gramáticas utilizadas nos testes foram geradas utilizando dois algoritmos de inferência diferentes, o de gramáticas reversíveis, chamado daqui em diante igRT [14] e o de gramáticas tipo-inversíveis, chamado daqui em diante igMakinen [15]. Em seguida, essas gramáticas foram transformadas em gramáticas estocásticas utilizando o algoritmo *Tree Grammar Reestimator*, chamado daqui em diante de EPTGR [18]. O próximo passo foi utilizar essas gramáticas para montar os classificadores.

Os primeiros testes foram realizados com imagens que possuíam quadrados e retângulos que respeitassem os quadrantes da *quadtree*. Em seguida, os testes foram realizados incluindo as formas geométricas de círculos e triângulos e a restrição de respeitar os quadrantes da *quadtree* foi retirada.

Durante os testes foram utilizadas imagens com resolução 1024x1024 e imagens com resolução 256x256, sendo utilizado o método de validação cruzada com $k = 4$ e com $k = 5$ com diferentes tamanhos de amostra de treinamento.

O objetivo do primeiro teste foi verificar se os classificadores eram capazes de diferenciar os quadrados dos retângulos (verticais e horizontais)¹, sendo que todas as imagens respeitavam os quadrantes. No segundo teste foram adicionadas aos testes imagens de círculos e triângulos² e foi retirada a restrição de respeitar os quadrantes. Nesse segundo teste, o objetivo foi verificar se os classificadores diferenciavam quadrados das demais figuras, ou seja, os testes foram quadrados *versus* círculos, quadrados *versus* triângulos e quadrados *versus* retângulos. Esses testes foram realizados com os algoritmos igRT e igMakinen e estão sumarizados na Tabela 2.

Tabela 2 – Síntese dos testes realizados

Teste	Respeitam os quadrantes	Quantidade de imagens	Resolução das imagens	(Validação Cruzada). Valor de k	Algoritmo de inferência gramatical
-------	-------------------------	-----------------------	-----------------------	---------------------------------	------------------------------------

¹Nos retângulos horizontais, o tamanho do comprimento era sempre o dobro do tamanho da largura e nos retângulos verticais, o contrário.

²Foram utilizados apenas triângulos retângulos, sendo a base e a altura do mesmo tamanho.

Teste 1	sim	96 (32 quadrados, 32 retângulos verticais e 32 retângulos horizontais)	1024x1024	4	igRT
Teste 2	não	400 (100 quadrados, 100 círculos, 100 triângulos e 100 retângulos)	1024x1024	5	igRT
Teste 3	não	400 (100 quadrados, 100 círculos, 100 triângulos e 100 retângulos)	256x256	5	igRT

Teste 4	sim	96 (32 quadrados, 32 retângulos verticais e 32 retângulos horizontais)	1024x1024	4	igMakinen
Teste 5	não	400 (100 quadrados, 100 círculos, 100 triângulos e 100 retângulos)	256x256	5	igMakinen

Os testes quadrados *versus* círculos, quadrados *versus* triângulos e quadrados *versus* retângulos com figuras com resolução 1024x1024 não foram realizados utilizando o algoritmo IgMakinen devido ao tamanho das sequências parentisadas geradas. Mais detalhes são apresentados no próximo capítulo.

É necessário ressaltar que cada imagem possuía o fundo branco e apenas uma figura geométrica com pixels pretos. Além disso, as figuras geométricas possuíam tamanhos diferentes e se encontravam em posições diferentes dentro da imagem. A Figura 37 exhibe 5 exemplos de cada figura geométrica analisada.

Para analisar os resultados gerados foram utilizadas as seguintes métricas: sensibilidade, especificidade, precisão, erro total, medida- F^3 e não classificação. A seguir é exibida a definição de como essas métricas foram calculadas.

- sensibilidade: número de positivos que o classificador classificou como positivo / número total de positivos;

³Essa medida é a média harmônica entre a precisão e a sensibilidade.

Figura 37 – Exemplo de imagens utilizadas. A primeira linha exibe amostras de quadrados, na segunda pode-se observar alguns círculos, na terceira, triângulos e na quarta, retângulos.

- especificidade: número de negativos que o classificador classificou como negativo / número total de negativos;
- precisão: número de positivos que o classificador classificou como positivo / número total de elementos (mais ou menos) que o classificador classificou como positivo;
- erro total: 100% - acurácia;
- medida-F: $((\beta^2 + 1) * P * VP) / (\beta^2 * P + VP)$, na qual P é a precisão e VP é a quantidade de verdadeiros positivos;
- não-classificação: número total de imagens não classificadas / número total dos elementos de teste.

Essas métricas foram escolhidas pois permitem uma análise real da capacidade dos classificadores gerados. Para seus cálculos considerou-se a classe quadrados como sendo “positiva” e as demais como “negativa”.

Para o cálculo das métricas citadas, foi considerado que, se o classificador não classificou uma determinada imagem, é porque ambas as gramáticas não a reconheceram e, portanto, trata-se de um erro.

Foram utilizadas duas máquinas para a execução dos testes:

- Máquina 1: Macbook Pro, processador Intel Core i7 2.7 GHz, memória de 16 GB.
- Máquina 2: Linux, processador Intel Xeon com 24 núcleos 2.3GHz, memória de 62 GB

Capítulo 6

Resultados e Discussão

Neste capítulo são apresentados os resultados e uma discussão acerca dos testes realizados. Na primeira seção estão os resultados obtidos utilizando o algoritmo de inferência gramatical IgRT. A seção seguinte traz os resultados dos testes mediante utilização do algoritmo de inferência gramatical IgMakinen. Na seção 5.3 é discutido e comparado o desempenho de cada um dos algoritmos utilizados para testar a metodologia proposta neste trabalho. Para o cálculo das métricas utilizadas para medir o desempenho foi considerado que os quadrados são os positivos e que as demais figuras geométricas são os negativos.

Para todos os testes, foi assumida uma distribuição uniforme sobre as *prioris* das classes, ou seja, elas foram consideradas equiprováveis e, portanto, a classificação resume-se a classificar para a classe representada pela gramática que gerou o maior valor de probabilidade de uma imagem.

6.1 Testes utilizando o algoritmo IgRT

6.1.1 Teste 1: Quadrados *versus* Retângulos que respeitam os quadrantes

Nesse teste foram utilizadas 32 imagens de quadrados e 64 imagens de retângulos que também respeitam os quadrantes. As características desses retângulos são similares às dos quadrados, ou seja, fundo branco, apenas um retângulo preto na imagem e resolução de 1024x1024. Das 64 imagens de retângulos, 32 eram retângulos na horizontal e 32 na vertical, sendo que um lado do retângulo era sempre o dobro do outro lado.

Foi utilizado o método de validação cruzada com $k=4$ tanto para os quadrados como para os retângulos. Considerando os quadrados, a cada iteração uma gramática foi gerada a partir de 24 sequências de treinamento, enquanto as 8 restantes compunham a amostra de teste. Já com os retângulos uma gramática foi gerada a partir de 48 sequências de treinamento, enquanto as 16 restantes compunham a amostra de teste. Ainda no caso dos retângulos, a gramática foi treinada tanto com retângulos na vertical como na horizontal.

O classificador foi composto por um analisador sintático de quadrados e um analisador sintático de retângulos. O classificador recebe uma sequência que passa pelos dois analisadores para obter a probabilidade. Em seguida, é calculada *a posteriori* e a imagem é classificada de acordo com a classe representada pela gramática com a maior *a posteriori*.

Características das sequências que representavam as imagens

Quadrados

- Maior sequência: 41 caracteres
- Menor sequência: 6 caracteres
- Tamanho médio das sequências: 23 caracteres

Retângulos

- Maior sequência: 41 caracteres
- Menor sequência: 6 caracteres
- Tamanho médio das sequências: 25 caracteres

Características das gramáticas inferidas

Os tamanhos médios das gramáticas dos quadrados e dos retângulos considerando a quantidade de símbolos não terminais, símbolos terminais e regras de produção podem ser vistos abaixo:

Quadrados

- Quantidade de nós não terminais: 1
- Quantidade de nós terminais: 2
- Quantidade de regras de produção: 8

Retângulos

- Quantidade de nós não terminais: 1
- Quantidade de nós terminais: 2

- Quantidade de regras de produção: 8

Resultados

Os resultados destes testes podem ser vistos nas métricas calculadas em seguida:

- erro total: 0%
- não classificação: 0%
- sensibilidade: 100%
- especificidade: 100%
- medida-F: 100%
- precisão: 100%

Discussão

Analisando as métricas, pode-se observar que o classificador obteve uma taxa de acerto 100% na classificação das imagens.

Analisando mais detalhadamente alguns exemplos de sequências que representam os quadrados e alguns exemplos de sequências que representam os retângulos chegou-se a algumas conclusões:

Sequências que representam quadrados

- sequência 1: (1101)
- sequência2: ((1011)111)
- sequência3: (1011)
- sequência4: (((111(111(111(111(111(1110)))))))111)

Sequências que representam retângulos

- sequência 1: (1010)
- sequência2: (0011)
- sequência3: (1(1((1010)111)11)11)

- sequência4: (((0101)111)111)

Primeiramente, no caso dos quadrados percebeu-se que sempre que tem quatro símbolos terminais juntos (0s e 1s), sempre aparece apenas um 0 (zero) e três 1 (um). Considerando os retângulos na mesma situação, temos sempre dois 0 (zero) e dois 1 (um). Como as gramáticas utilizadas foram treinadas utilizando sequências com esse padrão, ocorre que o classificador de quadrados sempre rejeita as sequências de retângulos e que o classificador de retângulos sempre rejeita as sequências de quadrados.

Esse padrão de 0 (zeros) e 1 (uns) só acontece porque as figuras respeitam os quadrantes da *quadtrees*. Para o próximo teste foi retirada a restrição de respeitar os quadrantes.

6.1.2 Teste 2: Quadrados *versus* Círculos, Triângulos e Retângulos que não respeitam os quadrantes - (resolução 1024x1024)

Para este teste foram utilizadas 100 imagens de cada classe, quadrados, círculos, triângulos e retângulos. A restrição das imagens respeitarem os quadrantes foi retirada e as imagens possuíam resolução de 1024x1024. Os testes foram feitos sempre considerando a classe quadrado contra as demais classes, ou seja, quadrados *versus* círculos, quadrados *versus* triângulos e quadrados *versus* retângulos.

Os testes foram realizados utilizando validação cruzada com $k=5$, ou seja, as gramáticas foram treinadas com 80 imagens e os testes realizados com as 20 restantes.

Características das sequências que representavam as imagens

A Tabela 3 sumariza as características das sequências que representam as 4 classes de imagens utilizadas.

Tabela 3 – Características das sequências utilizadas no teste

Forma Geométrica	Maior sequência	Menor sequência	Tamanho médio
Quadrados	6451	6	977
Círculos	9436	326	3136
Triângulos	11836	441	6126
Retângulos	14946	6	1711

Características das gramáticas inferidas

Podemos observar na Tabela 4 os tamanhos médios das gramáticas que compuseram

os classificadores para cada uma das classes utilizadas no que diz respeito ao número de símbolos não terminais, símbolos terminais e regras de produção.

Tabela 4 – *Tamanho médio das gramáticas utilizadas no teste*

Forma Geométrica	Símbolos não terminais	Símbolos terminais	Regras de produção
Quadrados	994	2	1056
Círculos	9561	2	9652
Triângulos	2213	2	2296
Retângulos	1106	2	1173

Resultados

Os testes com essas imagens não puderam ser realizados devido à dificuldade do algoritmo IgRT e do EPTGR implementados lidarem com gramáticas e sequências muito grandes. Durante os testes foram bastante frequente problemas com alocação de memória, pois as estruturas de dados utilizadas eram muito grandes.

Discussão

Conforme explicado ao longo dessa dissertação, o arcabouço utilizado é composto de 5 componentes. O primeiro é responsável pelo processamento das imagens (transformar imagens em sequências parentisadas). O segundo módulo atua convertendo as sequências em uma gramática. O terceiro módulo transforma a gramática gerada previamente em uma gramática estocástica. O quarto módulo criará um analisador sintático para a gramática obtida e o quinto módulo é o classificador propriamente dito.

No caso deste teste não foi possível converter as gramáticas geradas em gramáticas estocásticas. O problema ocorreu devido ao grande tamanho das gramáticas obtidas e ao comprimento das sequências. Podemos olhar nas Tabelas 3 e 4 que algumas sequências possuem mais de 10.000 caracteres e que algumas gramáticas possuem mais de 1000 símbolos não terminais e mais de 1000 regras de produção. Trabalhar com objetos e estruturas desse tamanho foi inviável com os equipamentos que tínhamos disponíveis¹, pois foi bastante comum problemas com estouro de memória ainda na fase de inicialização das variáveis e das estruturas utilizadas para a estimação das probabilidades.

A solução encontrada para prosseguirmos com os testes foi diminuir a resolução das imagens utilizadas para que fossem geradas sequências menores. Esse é o próximo teste apresentado.

¹Máquina 1: Macbook Pro, processador Intel Core i7 2.7 GHz, memória de 16 GB 1600 Mhz DDR3; Máquina 2: Linux, processador Intel Xeon com 24 núcleos 2.3GHz, memória de 62 GB

6.1.3 Teste 3: Quadrados *versus* Círculos, Triângulos e Retângulos que não respeitam os quadrantes - (resolução 256x256)

Para este teste foram utilizadas as mesmas imagens utilizadas no teste 3, mas a resolução foi diminuída de 1024x1024 para 256x256 para que fossem geradas sequências menores.

Características das sequências que representavam as imagens

Conforme pode ser observado na Tabela 5 diminuir a resolução das imagens de fato fez com que as sequências que representavam cada imagem ficassem menores.

Tabela 5 – Características das sequências utilizadas no teste

Forma Geométrica	Maior sequência	Menor sequência	Tamanho médio
Quadrados	2956	6	441
Círculos	2386	76	779
Triângulos	2761	131	1270
Retângulos	2766	6	718

Características das gramáticas inferidas

A diminuição do tamanho das sequências teve como consequência a diminuição do tamanho das gramáticas utilizadas para representar cada classe, conforme exibido na Tabela 6

Tabela 6 – Tamanho médio das gramáticas utilizadas no teste

Forma Geométrica	Símbolos não terminais	Símbolos terminais	Regras de produção
Quadrados	673	2	734
Círculos	3163	2	3252
Triângulos	1317	2	1400
Retângulos	781	2	848

Resultados

Apenas o teste quadrados *versus* triângulos foram completados e, da mesma forma que no teste 2, não foi possível completar os testes quadrados *versus* círculos e quadrados *versus* retângulos devido a constantes estouro de memória. Quando não ocorria esse problema com a memória o algoritmo executava por mais de 48 horas sem conseguir terminar o processo de estimação das probabilidades. Como estávamos utilizando validação cruzada

e era necessário inferir 5 gramáticas para cada classe, somando um total de 20 gramáticas, esperar a estimação das probabilidades de cada gramática por dias era inviável.

Discussão

Considerando o teste quadrado *versus* triângulos, das 100 imagens de quadrados utilizadas, o classificador reconheceu corretamente apenas 32. Nesse caso, essas 32 sequências são aquelas utilizadas no teste 1, ou seja, foram as sequências dos quadrados que respeitaram os quadrantes. Isso significa dizer que embora tenhamos aumentado o número de amostras de treinamento, a gramática de quadrados não foi capaz de ‘aprender’ o que é um quadrado e conseguiu reconhecer apenas as sequências mais simples. Considerando os triângulos, tem-se que o classificador não conseguiu classificar nenhuma sequência, o que nos mostra claramente que um treinamento utilizando 80 sequências não é suficiente para o IgRT gerar uma gramática que consiga reconhecer essa figura geométrica.

Embora não tenha sido possível realizar os testes com as demais figuras geométricas (círculos e retângulos), os resultados obtidos com os quadrados e triângulos leva a crer que as gramáticas inferidas também não seriam capazes de reconhecer círculos e retângulos.

6.2 Testes utilizando o algoritmo IgMakinen

6.2.1 Teste 4: Quadrados *versus* Retângulos que respeitam os quadrantes

Para este teste foram utilizadas as mesmas imagens e procedimentos utilizados no teste 1, ou seja, 32 imagens de quadrados, 64 imagens de retângulos e validação cruzada com $k=4$.

Características das sequências que representavam as imagens

As características das sequências que representavam as imagens estão resumidas abaixo

Quadrados

- Maior sequência: 41 caracteres
- Menor sequência: 6 caracteres
- Tamanho médio das sequências: 23 caracteres

Retângulos

- Maior sequência: 41 caracteres
- Menor sequência: 6 caracteres
- Tamanho médio das sequências: 25 caracteres

Características das gramáticas inferidas

Abaixo são exibidas as características das gramáticas inferidas.

Quadrados

- Quantidade de nós não terminais: 2
- Quantidade de nós terminais: 2
- Quantidade de regras de produção: 11

Retângulos

- Quantidade de nós não terminais: 2
- Quantidade de nós terminais: 2
- Quantidade de regras de produção: 11

Resultados

Os resultados obtidos no teste podem ser vistos nas métricas calculadas (sensibilidade, especificidade, erro total, medida-F, acurácia e precisão) a seguir:

- erro total: 0%
- não classificação: 0%
- sensibilidade: 100%
- especificidade: 100%
- medida-F: 100%
- precisão: 100%

Discussão

Pode-se observar nos resultados obtidos que o classificador obteve uma taxa de acerto de 100% na classificação das imagens. Essa alta taxa de acerto se deve ao fato de que as sequências que representam os quadrados que respeitam os quadrantes possuem apenas um 0 (zero) e três 1 (um) quando temos 4 símbolos terminais juntos. Já as sequências que representam os retângulos que respeitam os quadrantes possuem sempre dois 0 (zero) e dois 1 (um) quando temos 4 símbolos terminais juntos. Esse padrão encontrado nas sequências foi aprendido pelos classificadores que conseguiram classificar corretamente todas as sequências utilizadas no teste.

6.2.2 Teste 5: Quadrados *versus* Círculos, Triângulos e Retângulos que não respeitam os quadrantes - (resolução 256x256)

As imagens utilizadas neste teste são as mesmas utilizadas no teste 3, ou seja, 100 imagens de cada classe (quadrados, círculos, triângulos e retângulos) com resolução 256x256. Foi utilizado o método de validação cruzada com $k=5$ e o algoritmo de inferência IgMakinen.

Características das sequências que representavam as imagens

Na Tabela 7 são exibidas as características das imagens utilizadas.

Tabela 7 – Características das sequências utilizadas no teste

Forma Geométrica	Maior sequência	Menor sequência	Tamanho médio
Quadrados	2956	6	441
Círculos	2386	76	779
Triângulos	2761	131	1270
Retângulos	2766	6	718

Características das gramáticas inferidas

Pode-se ver na Tabela 8 o tamanho médio das gramáticas geradas pelo IgMakinen no que diz respeito à quantidade de nós não terminais, nós terminais e regras de produção.

Resultados

Os resultados apresentados serão divididos em três: Quadrados *versus* Círculos, Quadrados *versus* Triângulos e Quadrados *versus* Retângulos.

Quadrados vs Círculos

Tabela 8 – Tamanho médio das gramáticas utilizadas no teste

Forma Geométrica	Símbolos não terminais	Símbolos terminais	Regras de produção
Quadrados	10	2	106
Círculos	16	2	357
Triângulos	14	2	165
Retângulos	11	2	93

Os resultados obtidos com este teste podem ser observados nas métricas calculadas abaixo.

- erro total: 37%
- não classificação: 34%
- sensibilidade: 73%
- especificidade: 53%
- medida-F: 73%
- precisão: 72,28%

Quadrados vs Triângulos

A seguir são exibidas as métricas obtidas para este teste:

- erro total: 36%
- não classificação: 29,5%
- sensibilidade: 66%
- especificidade: 62%
- medida-F: 66%
- precisão: 66%

Quadrados vs Retângulos

Os resultados dos testes podem ser visualizados nas métricas apresentadas a seguir:

- erro total: 48,5%
- não classificação: 16%
- sensibilidade: 64%
- especificidade: 39%
- medida-F: 52,5%
- precisão: 44,5%

Discussão

Considerando os testes Quadrados *versus* Círculos, pode-se observar que de fato o classificador criado consegue diferenciar o que é um quadrado e o que é um círculo para a maioria das imagens. Analisando as métricas foi percebido que a classificação ocorreu com sensibilidade de 73% e precisão de 72,28%. É importante salientar que para os cálculos das métricas foram computadas o resultado da análise de todas as imagens utilizadas nos testes, ou seja, se o classificador não conseguiu reconhecer uma determinada imagem, é considerada uma falha e, portanto, uma “classificação” errada. Para esse teste, tem-se que a taxa de não classificação foi de 34%, ou seja, das 200 imagens utilizadas no teste 68 não foram classificadas como quadrados ou círculos. Das 100 imagens de círculos, apenas 54 foram classificadas (53 classificadas corretamente e 1 classificada erroneamente), já considerando os quadrados, das 100 imagens apresentadas 78 foram classificadas (73 corretamente e 5 de forma equivocada).

Analisando os resultados dos testes Quadrados *versus* Triângulos, percebemos que a sensibilidade foi de 66%, a especificidade foi de 62% e a precisão foi de 66%. Assim como anteriormente, nem todas as imagens tiveram uma classificação, pois das 100 imagens de triângulos apenas 62 foram classificadas (todas corretamente) e das 100 imagens de quadrados apenas 79 obtiveram alguma classificação (66 de forma correta e 13 de forma incorreta), sendo a taxa de não classificação de 29,5%.

Observando os resultados obtidos no teste Quadrados *versus* Retângulos, foi percebido que o classificador errou bastante durante a classificação dessas figuras geométricas. O valor da especificidade alcançada é de apenas 39% o que significa que o classificador classificou a maioria dos retângulos como sendo quadrados. Na classificação dos quadrados os resultados foram um pouco melhor e a sensibilidade foi de 64%. Da mesma forma que nos dois testes anteriores nem todas as imagens reconhecidas, uma vez que das 100 imagens

de quadrados apenas 81 foram classificadas (64 corretamente e 17 de forma incorreta) e das 100 imagens de retângulos apenas 87 receberam alguma classificação (39 de forma correta e 48 de forma incorreta), sendo a taxa de não classificação de 16%.

Comparando os resultados desses três testes (Quadrados *versus* Círculos, Quadrados *versus* Triângulos e Quadrados *versus* Retângulos) percebemos que o pior desempenho foi alcançado nesse último teste. Isso se dá devido à maior semelhança dos quadrados com os retângulos que com os círculos e com os triângulos. Este fato fez com que as gramáticas utilizadas nos classificadores de quadrados fossem bastante semelhantes às gramáticas utilizadas nos classificadores de retângulos.

6.3 Comparação entre os algoritmos IgRT e IgMakinen

Nesta seção é feita uma comparação entre os desempenhos obtidos pelos algoritmos IgRT e IgMakinen no processo de inferência gramatical, baseando-nos nos resultados obtidos através dos testes de 1 a 7 realizados.

Testes 1 e 4 - Quadrados *versus* Retângulos que respeitam os quadrantes

As gramáticas inferidas para os quadrados e para os retângulos tanto pelo IgRT quanto pelo IgMakinen possuem praticamente o mesmo tamanho, sendo que as geradas IgRT são um pouco menor. Ambas demonstraram a mesma capacidade para diferenciar quadrados e retângulos, obtendo uma taxa de acerto de 100% na classificação.

Teste 2 - Quadrados *versus* Círculos, Triângulos e Retângulos que não respeitam os quadrantes - (resolução 1024x1024)

Para esses testes foram realizados apenas o algoritmo IgRT, mas sem sucesso, pois as sequências utilizadas eram muito grandes (imagens com resolução de 1024x1024). O IgRT gerou gramáticas muito grandes o que tornou inviável a execução dos testes. Foi visto em [4] que, utilizando amostras de treinamento cada vez maiores, as gramáticas inferidas pelo IgRT têm seu tamanho incrementado, considerando o número de produções, até chegar um ponto que começa a diminuir até estabilizar. Nos nossos treinamentos utilizamos sempre 80 imagens o que provavelmente é pouco para que o IgRT gere gramáticas pequenas. O problema é que não tínhamos em mãos a quantidade de imagens necessárias, poderíamos utilizar 100, 200 ou mais até obtermos gramáticas menores, mas devido à falta de tempo abandonamos esses testes com imagens com resolução 1024x1024 e começamos os testes

com imagens com resolução 256x256.

Testes 3 e 5 - Quadrados *versus* Círculos, Triângulos e Retângulos que não respeitam os quadrantes

Para esses testes foram utilizadas imagens com resolução 256x256 e os algoritmos de inferência IgRT e IgMakinen.

As gramáticas geradas pelo IgMakinen são muito menores que as gramáticas geradas pelo IgRT, conforme pode ser visto nas Tabelas 9, 10, 11 e 12. Isso acontece devido ao fato do algoritmo IgMakinen fazer uso da estrutura *trie* e o IgRT não. Além disso, só foi possível testar o IgRT com quadrados *versus* triângulos, pois mesmo com imagens menores as gramáticas inferidas ainda eram muito grandes. Além disso, ficou claro que 80 imagens não é suficiente para o treinamento utilizando o IgRT uma vez que os classificadores de quadrados apenas conseguiu reconhecer os quadrados que respeitavam os quadrantes e os classificadores de triângulo não conseguiu reconhecer um triângulo sequer. Em contrapartida, o algoritmo IgMakinen não só gerou gramáticas bem menores, mas também conseguiu aprender melhor as formas geométricas. Na verdade, as gramáticas geradas pelo IgMakinen só não foram capazes de reconhecer quadrados *versus* círculos e quadrados *versus* triângulos com uma alta taxa de acerto devido a alta taxa de não classificação obtida. Já desempenho do classificador quadrados *versus* retângulos foi pior que o desempenho dos testes anteriores dada as semelhanças entre essas duas formas geométricas.

Tabela 9 – Comparação das gramáticas de quadrados geradas pelo IgRT e pelo IgMakinen

	Símbolos não terminais	Símbolos Terminais	Regras de Produção
IgRT	673	2	734
IgMakinen	10	2	106

Tabela 10 – Comparação das gramáticas de círculos geradas pelo IgRT e pelo IgMakinen

	Símbolos não terminais	Símbolos Terminais	Regras de Produção
IgRT	3163	2	3252
IgMakinen	16	2	357

Os testes com o algoritmo IgRT não foram promissores por dois motivos: (i) a dificuldade computacional de lidar com gramáticas muito grandes; (ii) a necessidade de amostras de treinamento maiores para que as gramáticas inferidas fossem capazes de reconhecer as

Tabela 11 – Comparação das gramáticas de triângulos geradas pelo IgRT e pelo IgMakinen

	Símbolos não terminais	Símbolos Terminais	Regras de Produção
IgRT	1317	2	1400
IgMakinen	14	2	165

Tabela 12 – Comparação das gramáticas de retângulos geradas pelo IgRT e pelo IgMakinen

	Símbolos não terminais	Símbolos Terminais	Regras de Produção
IgRT	781	2	848
IgMakinen	11	2	93

classes das figuras geométricas. Embora a teoria de identificação no limite garanta que essa classe gramatical possa ser identificada em tempo polinomial ela não informa a quantidade de amostras necessárias para isso e nem o momento no qual o critério de parada é alcançado.

Os testes com o algoritmo IgMakinen foram mais promissores, devido ao fato de este algoritmo ter gerado gramáticas menores e pelo fato das gramáticas geradas conseguirem reconhecer algumas figuras geométricas. O problema encontrado com esses testes foram as altas taxas de imagens não classificadas, sendo que as taxas de não classificação foram de 34%, 29,5% e 16% para os testes Quadrados *versus* Círculos, Quadrados *versus* Triângulos e Quadrados *versus* Retângulos, respectivamente.

Capítulo 7

Conclusão

7.1 Principais Resultados Obtidos

Neste trabalho foi utilizado um arcabouço para geração de classificadores de sequências genéticas baseados em gramáticas estocásticas. Esse arcabouço foi adaptado para que pudesse ser utilizado no reconhecimento de padrões em imagens. Para tanto, algumas classes de **Streams** foram criadas para que o arcabouço pudesse receber como entrada *strings* parentisadas que representavam as estruturas de árvores das *quadtrees* das imagens utilizadas nos treinamentos e nos testes. Além disso, foi criado um módulo para processamento de imagens responsável por obter uma *string* parentisada utilizando a estrutura *quadtree* da imagem

Os testes realizados tinham como objetivo reconhecer figuras geométricas (quadrados, círculos, triângulos e retângulos). Para atingir tal objetivo fez-se uso de dois algoritmos de inferência gramatical, o IgRT e o IgMakinen, e um algoritmo de estimação de probabilidades, o EPTGR.

Os primeiros testes foram realizados utilizando figuras de quadrados e retângulos que respeitavam os quadrantes. Nestes testes, tanto os classificadores gerados com analisadores obtidos a partir de gramáticas inferidas com o IgRT, como os classificadores gerados com analisadores sintáticos obtidos a partir de gramáticas inferidas com o IgMakinen foram capazes de acertar corretamente a classificação de todas as figuras apresentadas.

Em seguida, foram realizados testes com imagens de quadrados, círculos, triângulos e retângulos que não respeitavam os quadrantes da *quadtree* e analisando os resultados apresentados na seção anterior percebemos que não obtivemos sucesso na utilização do IgRT. Isso ocorreu, pois algumas gramáticas geradas por este inferidor gramatical eram muito grandes (mais de 3 mil símbolos não-terminais e mais de 3 mil regras de produção), o que fez com que tivéssemos problemas com alocação de memória e com o tempo de execução dos algoritmos. Além disso, considerando o teste que conseguimos terminar, Quadrados *versus* Triângulos, percebemos que as gramáticas inferidas utilizando 80 sequências na

amostra de treinamento não eram capazes de reconhecer os objetos de sua classe. No caso dos quadrados, apenas os quadrados que respeitavam os quadrantes foram reconhecidos, enquanto que no caso dos triângulos, nenhum triângulo foi reconhecido.

Os testes efetuados utilizando o algoritmo de inferência gramatical IgMakinen foram mais promissores. Além das gramáticas geradas por este inferidor serem bem menores que as geradas pelo IgRT, elas demonstraram um melhor desempenho no reconhecimento das figuras geométricas. Nos testes Quadrados *versus* Círculos foi obtida uma sensibilidade média de 73%, uma especificidade média de 53%, uma precisão de 72,28%, um erro total de 37% e uma taxa de não classificação de 34%. Já nos testes Quadrados *versus* Triângulos foi obtida uma sensibilidade média de 66%, especificidade média de 62%, precisão de 66%, um erro total de 36% e uma taxa de não classificação de 29,5%. O pior resultado foi obtido nos testes Quadrados *versus* Retângulos, os quais forneceram uma sensibilidade média de 64%, especificidade média de 39%, precisão de apenas 44,5%, erro total 48,5% e uma taxa de não classificação de 16%.

É importante notar que para o cálculo das métricas foram consideradas todas as imagens utilizadas nos testes, ou seja, o fato de uma imagem não ter reconhecida e classificada foi considerado um erro de classificação. Considerando apenas os casos em que houve classificação, as taxas de acerto para quadrados *versus* círculos e quadrados *versus* triângulos foram excelentes (precisão de 98,75% e 100%, respectivamente).

Pode-se perceber que a representação de uma imagem utilizando métodos sintáticos permite representar uma imagem de forma concisa, principalmente considerando seus aspectos estruturais. Além disso, se as imagens forem consideradas de forma hierárquica é possível fazer o mapeamento a partir da estrutura hierárquica da imagem analisada com as regras de produção de uma gramática. Entretanto, embora a representação da imagem possa ser mais concisa, o custo computacional (memória e processamento) pode ser muito alto.

Durante os testes com o algoritmo IgRT, o poder computacional disponível não foi suficiente para que os testes fossem finalizados, sendo que ocorreram vários problemas com alocação de memória e com o tempo de processamento.

Também ficou claro que a abordagem utilizando a estrutura *quadtree* é muito sensível ao fato das figuras geométricas (quadrados e retângulos) respeitarem ou não os quadrantes. Quando as figuras respeitavam os quadrantes a taxa de classificação foi de 100%, mas quando as figuras não respeitavam essa restrição, a taxa de classificação caiu consideravelmente e várias imagens não foram reconhecidas pelos analisadores sintáticos. Este

fato mostra uma limitação dos métodos sintáticos quando são utilizados para lidar com imagens mais complexas.

As taxas de classificações corretas das imagens utilizando a metodologia proposta, em geral, não foram muito altas, principalmente devido à grande quantidade de imagens que ficaram sem classificação, pois não foram reconhecidas pelos analisadores sintáticos utilizados. Para melhorar essa taxa de imagens não classificadas é necessário investir mais tempo nos testes, por exemplo, aumentando a quantidade de imagens disponíveis nas amostras de treinamento.

7.2 Principais Contribuições

As principais contribuições obtidas com esse trabalho foram: a publicação de um artigo contendo a revisão sistemática criada, adaptação do arcabouço utilizado e utilização da estrutura de *quadtrees* para reconhecimento de objetos em imagens.

O artigo “*Using grammars for pattern recognition in images: a systematic review*” a ser publicado pela revista “*ACM Computing Surveys*” v. 46, n. 2, trata-se uma revisão bibliográfica sobre a utilização de métodos sintáticos e reconhecimento de padrões em imagens bastante profunda e com qualidade. Antes desse artigo apenas 1 revisão sobre o assunto foi encontrada [8], sendo que esta não cobria a variedade e quantidade de trabalhos relacionados ao assunto. Esse artigo pode servir como base para que novos pesquisadores tomem conhecimento das várias técnicas que vem sendo empregadas ao longo dos anos, suas principais vantagens e limitações.

Outra conquista importante foi a adaptação do arcabouço que inicialmente havia sido desenvolvido para trabalhar com sequências genéticas. Este arcabouço foi adaptado de forma que pudesse receber como entrada as estruturas de *quadtrees* das imagens, gerando, gramáticas que fossem capazes de reconhecer imagens de várias classes distintas. É importante ressaltar que esse arcabouço pode ser adaptável para trabalhar com outras estruturas e, portanto, pode ser uma ferramenta útil para outros pesquisadores.

A última conquista foi a abordagem de *quadtrees* utilizada nesse projeto. Durante a revisão bibliográfica não foi encontrada na literatura nenhuma pesquisa que tivesse utilizado essa estrutura para generalizar um modelo a ser utilizado em reconhecimento de padrões em imagens. Embora sejam necessários mais testes, essa abordagem parece promissora e poderia ser utilizada, por exemplo, para recuperação de imagens baseada em conteúdo.

7.3 Trabalhos Futuros

A seguir são apresentados os possíveis trabalhos futuros desta linha de pesquisa.

Exploração do uso do IgMakinen:

O algoritmo IgMakinen mostrou um melhor desempenho que o IgRT no reconhecimento de figuras geométricas. Seria interessante explorar melhor a sua utilização, ainda considerando as figuras geométricas, aumentando o tamanho das amostras de treinamento. Podemos observar que embora os resultados obtidos nos testes tenham sido razoáveis (Quadrados *versus* Círculos e Quadrados *versus* Triângulos), algumas figuras ficaram sem ser classificadas, ou seja, os classificadores criados não conseguiram dizer a qual classe a figura pertencia. Além disso, os classificadores cometeram bastantes erros na classificação dos Quadrados *versus* Retângulos. É necessário verificar se essa quantidade de erros seria menor caso os classificadores sejam treinados com amostras de treinamento maiores.

Utilização de imagens em tons de cinza:

Os testes realizados sempre utilizaram imagens com fundo branco e com a figura geométrica em preto. Também seria interessante estender um pouco esses testes trabalhando com figuras em tons de cinza.

Reconhecimento de objetos simples do mundo real:

Outra aplicação interessante seria a utilização de imagens com objetos simples do mundo real, por exemplo, talheres, copos, pratos, tesouras, martelos, garrafas, folhas de árvores etc. Esses objetos são relativamente simples e talvez pudessem ser classificados utilizando o arcabouço adaptado para este projeto.

Reconhecimento de padrões em imagens médicas:

Outra aplicação bastante útil seria a aplicação da metodologia desenvolvida neste projeto para o reconhecimento de padrões em imagens médicas. Principalmente imagens de tumores que pudessem ser reconhecidos através de sua forma geométrica, por exemplo, imagens de tumores benignos e malignos em glândulas mamárias, desde que a região suspeita seja demarcada manualmente pelo radiologista.

APÊNDICE A

Síntese dos artigos analisados

Tabela 13 – Tabela com a síntese de todos os artigos analisados.

Artigos	Objetivo	Critérios de Inclusão	Técnicas	Primitivas	Amostra	Gramática	Resultados
[32]	Reconhecimento de Objetos	b	Modelo similar a gramáticas estocásticas; modelos ocultos de Markov	Uma região, um componente do objeto, uma medida, etc.	Documento contendo notas musicais	Não exibida	O modelo proposto ainda não havia sido terminado
[77]	Reconhecimento de Objetos e Texturas	c	<i>Spatial Random Tree</i> ; modelos estocásticos de árvores escondidas; gramáticas estocásticas livres de contexto	Pixels e regiões da imagem	Imagens de texturas e de construções	Não exibida	100% de classificação correta das texturas e mais de 97% de classificação correta das imagens de construções
[33]	Reconhecimento de Objetos	c	<i>Spatial Random Tree</i> ; modelos estocásticos de árvores escondida; gramáticas estocásticas livres de contexto	Pixels e regiões da imagem	600 imagens de fachadas de prédios e texturas	Exibidas algumas regras de produção	Taxa de classificação correta superior a 97%
[35]	Reconhecimento de Objetos	c	<i>Spatial Random Tree</i> ; gramáticas estocásticas livres de contexto; algoritmo <i>center-surround</i> (similar ao <i>inside-ouside</i>)	Vetores de características da imagem	100 fotografias de carros e casas	Não exibida	Taxa de classificação correta superior a 97%

[3]	Reconhecimento de Objetos	a, c	Modelos de Markov e gramáticas probabilísticas; grafos AND-OR; <i>oriented triplets</i>	<i>Oriented triplets</i>	Imagens retiradas de Caltech 4 e Caltech 10 ¹	Inferida (grafos AND-OR)	Taxa de classificação correta superior a 87.6%
[38]	Reconhecimento de Objetos	c	Método <i>belief propagation</i> ; gramáticas estocásticas livres de contexto; <i>Markov Random Field</i> ; grafos AND-OR; algoritmo <i>top-down/bottom-up</i>	Blocos de pixels representando componentes da face (olhos, nariz etc)	Faces humanas (AR Face database ² ; Yale Face database ³ ; ORL ⁴ ; Image Parsing ⁵ ; Georgia ⁶)	Não exibida	Taxa de classificação correta igual a 95% relacionado à posição dos olhos em uma imagem
[39]	Reconhecimento de Objetos	c	Grafos gramaticais com atributos (grafos AND-OR); algoritmo <i>top-down/bottom-up</i>	Retângulos planares 3D projetados nas imagens	40 imagens com uma média de 38 retângulos cada	Exibida	Apresentação das imagens geradas
[26]	Reconhecimento de Objetos	c	Grafos gramaticais com atributos (grafos AND-OR); algoritmo <i>top-down/bottom-up</i>	Retângulos planares 3D projetados nas imagens	40 imagens de objetos construídos pelo homem	Exibida	Apresentação das imagens geradas
[40]	Reconhecimento de Objetos	b	grafos AND-OR	Partes de corpos humanos e roupas	Imagens de pedestres	Não exibida	Apresentada algumas imagens com os resultados
[41]	Reconhecimento de Objetos	b	Grafos AND-OR	Templates de partes atômicas	Lotus Hill Institute	Não exibida	Construção de um banco de dados de imagens

¹<http://www.vision.caltech.edu/html-files/archive.html>²<http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html>³<http://cvc.yale.edu/projects/yalefaces/yalefaces.html>⁴<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>⁵<http://www.parsing.com/>⁶<http://www.anefian.com/research/face-reco.htm>

[42]	Reconhecimento de Objetos	b	Grafos AND-OR	Características das imagens de baixo nível	Interfaces Gráficas - GUI	Exibido um exemplo	Taxas de detecção correta: 90% para <i>pushbutton</i> , 100% para <i>menulists</i> , 90% para <i>radiobutton</i> e 67% para <i>checkbox</i>
[43]	Reconhecimento de Objetos	b	Gramáticas estocásticas sensíveis ao contexto	Conjunto de pixels	Imagens de placas de carros	Não exibida	Imagens com os resultados
[44]	Reconhecimento de Objetos	b	Gramáticas estocásticas de atributos sensíveis ao contexto	Pequenos retângulos	Imagens de fachadas	Exibido um exemplo	Imagens com os resultados.
[46]	Reconhecimento de Objetos	a, c	<i>Mixture Hidden Markov Model; Adaptive Neuro-Fuzzy Inference System; Oriented Triplets</i>	<i>Oriented triplets</i>	Imagens da Caltech 101	Inferida (Não exibida)	Taxa de classificação correta igual a 80% utilizando ANFIS e 93% utilizando MHMM
[47]	Reconhecimento de Objetos	a	<i>Augmented Regular Expressions</i> ; linguagens sensíveis ao contexto; <i>Active Grammatical Inference</i>	Pixels	Placas de trânsito	Inferida	Média por volta de 0.68% para os autômatos encontrados
[45]	Reconhecimento de Objetos	b	Gramáticas estocásticas sensíveis ao contexto	Formas geométricas	Imagens aéreas	Apresentada algumas regras de produção	Exibida algumas imagens com os resultados
[48]	Reconhecimento de Objetos	b	Lógica de predicado; modelos de incerteza	Segmentos de retas, círculos, cantos etc.	USC-CAVIAR ⁷ database e um conjunto de dados próprio	Não exibida	Curvas ROC
[49]	Reconhecimento de Objetos	b	Gramáticas de adjacência	Segmentos de retas e arcos	700 instâncias de símbolos gráficos	Exibida	Taxa de classificação correta igual a 87.7%
[50]	Reconhecimento de Objetos	b	Gramáticas de adjacência	Segmentos de retas e arcos	Imagens capturadas com canetas eletrônicas	Exibidas algumas regras de produção	Apresentação das imagens geradas

⁷<http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>

[52]	Reconhecimento de Objetos	b	Grafos gramaticais ETPL(k)	Segmentos de retas e pontos	20 imagens arteriais	Geração de grafos ETPL(k)	Taxa de classificação correta igual a 85%
[51]	Reconhecimento de Objetos	b	Grafos gramaticais ETPL(k)	Segmentos de retas e pontos	Imagens de artérias	Geração de grafos ETPL(k)	Taxa de classificação correta igual a 85%
[53]	Reconhecimento de Objetos	b	Gramáticas livre de contexto	Segmentos de retas	Imagens médicas	Exibida	Taxa de classificação correta igual a 90%
[54]	Reconhecimento de Objetos	a, b	Grafos gramaticais; gramáticas de atributo livres de contexto	Centro de gravidade de um segmento da imagem e segmentos de retas	Imagens de ossos e órgãos humanos	Exibida	Taxa de classificação correta igual a 93%
[55]	Reconhecimento de Objetos	b	Modelos sintáticos hierárquicos; Gramáticas livre de contexto	Partes de objetos	Imagens de ferramentas (martelos, pás etc)	Não exibida	Imagens com os resultados.
[56]	Reconhecimento de Objetos	b	Gramáticas livre de contexto genéricas	Patches planares	Imagens de construções ⁸	Exibido um exemplo	89.3% na detecção dos patches planares. 76.2% na detecção dos parentes de cada <i>patch planar</i>
[57]	Reconhecimento de Objetos	a	<i>Sparse Kernel Feature Analysis</i> , grafos EDT	Conjunto de pixels	Imagens capturadas em ambiente urbano - carros	Geração de grafos EDT	Apresentação dos grafos gerados
[58]	Reconhecimento de Objetos	b	Gramáticas regulares	Objetos, relacionamentos geométricos e descontinuidade temporal	1 exemplo descrevendo uma mesa para o café da manhã	Exibidos alguns exemplos de regra de produção	Descrição de uma sequência de eventos baseado em imagens
[59]	Reconhecimento de Objetos	c	<i>Three-dimensional plex grammar</i>	Superfícies de objetos	Objetos geométricos (caixas, por exemplo)	Exibida	Apresentação das imagens geradas
[60]	Reconhecimento de Objetos	b	Combinação de métodos sintáticos e estruturais para representação da escrita chinesa	Traços e segmentos de retas	7920 caracteres chineses escritos à mão	Não exibida	Nota 0.774 [0-1] para o caracter exibido no artigo

⁸<http://www.wrightstatearc.comwright>

[62]	Reconhecimento de Objetos	b	<i>Shock graphs</i> e <i>shock grammar</i>	Vértices de diferentes <i>shock groups</i>	Imagens binárias ⁹	Exibido um exemplo	Exibidas imagens e tabelas com os resultados
[61]	Reconhecimento de Objetos	b	<i>Shock graphs</i> e <i>shock grammar</i>	Vértices de diferentes <i>shock groups</i>	Imagens binárias ¹⁰	Exibido um exemplo	Exibidas imagens e tabelas com os resultados
[63]	Reconhecimento de Objetos	b	<i>Shock graphs</i> e <i>shock grammar</i>	Vértices de diferentes <i>shock groups</i>	Vários tipos de imagens	Exibido um exemplo	Exibidos em imagens
[64]	Reconhecimento de Objetos	b	<i>Shock graphs</i> e <i>shock grammar</i>	Vértices de diferentes <i>shock groups</i>	Imagens de objetos em geral	Exibida	Exibidas imagens e tabelas com os resultados
[65]	Reconhecimento de Texturas	a	Transformadas <i>Wavelet</i> ; gramáticas <i>fuzzy</i>	Vetores de características	30 tipos diferentes de texturas	Inferida (a partir de regras <i>fuzzy</i> pré-definidas)	Valor superior a 85% para a regra que descrevia a textura
[66]	Reconhecimento de Texturas	a	Transformadas <i>Wavelet</i> ; gramáticas <i>fuzzy</i>	Vetores de características	Amostras de cortiças retiradas de ambiente de produção	Inferida (a partir de regras <i>fuzzy</i> pré-definidas)	Taxa de classificação correta superior a 90% para homogeneidade e superior a 94% para a detecção da base da cortiça
[27]	Reconhecimento de Texturas	a, b	Transformadas <i>Wavelet</i> ; gramáticas <i>fuzzy</i>	Vetores de características	100 amostras de cortiças retiradas de ambiente de produção	Inferida (a partir de regras <i>fuzzy</i> pré-definidas)	Taxa de classificação correta superior a 90% para homogeneidade e superior a 94% para a detecção da base da cortiça
[67]	Construção de Objetos	b	Gramáticas hierárquicas; <i>Data-driven Markov chain Monte Carlo</i>	Elementos geométricos (triângulos, retângulos etc)	80 imagens de objetos extraídos de LHI ¹¹ public database	Não exibida	Apresentação das imagens geradas
[68]	Construção de Objetos	b	<i>2D tabled matrix grammar</i>	Vetor de elementos	3 exemplos utilizando a técnica	Exibida	Apresentação das imagens geradas

⁹<http://www.lems.brown.edu/dmc/main.html>¹⁰www.lems.brown.edu/dmc¹¹<http://www.imageparsing.com/FreeDataOutline.html>

[28]	Construção de Objetos	b	<i>L-System; Markov Chain Monte Carlo</i>	Figuras geométricas (cilindros, elipses etc)	184 imagens do fungo <i>Alternaria</i>	Exibida algumas regras de produção	Apresentação das imagens geradas
[69]	Construção de Objetos	b	<i>L-system</i>	Símbolos gráficos	–	Não exibida (<i>L-system</i>)	Apresentação das imagens geradas
[70]	Construção de Objetos	a, b	<i>L-system</i>	Pixels	–	Inferidas (<i>L-System</i>)	Apresentação das imagens geradas
[71]	Construção de Objetos	b	Extensão do <i>L-system</i>	Pixels e segmentos de retas	–	Exibidas algumas regras de produção	Apresentação das imagens geradas
[72]	Construção de Objetos	b	<i>L-Systems</i>	Não mencionado	Imagens de troncos de árvores	Inferida	Exibida algumas imagens com os resultados
[29]	Segmentação de Imagem	b, c	Gramáticas estocástica livre de contexto; grafos AND-OR; algoritmo <i>top-down/bottom-up</i>	Pixels	Imagens de ambientes internos e externos	Não exibida	O método leva 10-20 min; apresentação das imagens geradas
[73]	Segmentação de Imagem	b	Gramática regular	Conjunto de pixels em formatos pré-definidos	Imagens cardíacas	Exibida	Precisão de 93.22% na estimação de áreas segmentadas
[30]	Mudança de escala	b, c	<i>Primal Sketch Representation</i> ; grafos gramaticais; <i>Markov chain Monte Carlo</i> ; <i>quadtrees</i>	Segmentos de retas, cruzes etc	Imagens de carros, construções e rostos	Não exibida	Apresentação das imagens geradas
[31]	Reconhecimento de <i>Layout</i>	b	Gramática estocástica regular	cabeçalho, rodapé e colunas de textos	Documentos	Não exibida	Algoritmo mais robusto que o utilizado para comparação
[74]	Reconhecimento de <i>Layout</i>	a, b	Gramáticas estocásticas livre de contexto	Dígitos, caracteres, símbolos, linhas etc	Documentos científicos	Inferida e criada manualmente	85% de precisão no reconhecimento do layout de documento e 94% de precisão no reconhecimento de fórmulas matemáticas

[75]	Reconhecimento de Layout	b	Grafos gramaticais sensíveis ao contexto	Textos, botões, imagens etc	Páginas web	Não exibida	Página web exibida como um grafo
[76]	Reconhecimento de Layout	b	Gramática de atributo estocástica sensível ao contexto	Regiões retangulares	Imagens de documentos	Não exibida	Imagens com os resultados.

Referências

- 1 MENEZES, P. B. Linguagens formais e automatos. Bookman: Instituto de Informatica da UFRGS, 5. ed., 2008.
- 2 RAMOS, M. V. M.; NETO, J. J.; VEGA, I. S. *Linguagens Formais - Teoria, Modelagem e Implementação*. 1st. ed. [S.l.]: Bookman Companhia Ed, 2009. ISBN 8577804534.
- 3 ZHU, L.; CHEN, Y.; YUILLE, A. Unsupervised learning of probabilistic grammar-markov models for object categories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 31, n. 1, p. 114 –128, jan. 2009. ISSN 0162-8828.
- 4 OLIVEIRA, A. M. L. Laboratorio de geracao de classificadores de sequencias. Instituto de Matematica e Estatistica - IME/USP, p. 1–142, 2002.
- 5 AHO, A. V.; ULLMAN, J. D. The theory of parsing, translation, and compiling. Prentice-Hall, Inc., Englewood Cliffs, NJ, v. 1, 1972.
- 6 SALOMAA, A. Formal languages. Academic Press, 1973.
- 7 HOPCROFT, J. E.; ULLMAN, J. D. Introduction to automata theory, languages, and computation. Addison-Wesley Publishing Company, 1979.
- 8 CHANDA, G.; DELLAERT, F. *Grammatical Methods in Computer Vision: An Overview*. [S.l.], 2004. 1-23 p.
- 9 GOLD, E. M. Language Identification in the Limit. *Information and Control*, v. 10, p. 447–474, 1967.
- 10 SAKAKIBARA, Y. Grammatical inference: An old and new paradigm. In: JANTKE, K.; SHINOHARA, T.; ZEUGMANN, T. (Ed.). *Algorithmic Learning Theory*. [S.l.]: Springer Berlin Heidelberg, 1995, (Lecture Notes in Computer Science, v. 997). p. 1–24. ISBN 978-3-540-60454-9.
- 11 LEVY, L. S.; JOSHI, A. K. Skeletal structural descriptions. *information and control*. *Information and Control*, v. 39, p. 192–211, 1978.
- 12 SEMPERE, M. J.; FOS, A. Learning linear grammars from structural information. *Lecture Notes in Artificial Intelligence*, v. 1891, p. 126–133, 2000.
- 13 TAKADA, Y. Grammatical inference for even linear languages based on control sets. *Information Processing Letters*, v. 28, n. 4, p. 193 – 199, 1988. ISSN 0020-0190.
- 14 SAKAKIBARA, Y. Efficient learning of context-free grammars from positive structural examples. *Information and Computation*, v. 97, n. 1, p. 23 – 60, 1992. ISSN 0890-5401.

- 15 MÄKINEN, E. On the structural grammatical inference problem for some classes of context-free grammars. *Inf. Process. Lett.*, v. 42, n. 1, p. 1–5, 1992.
- 16 BALDI, P.; BRUNAK, S. *Bioinformatics: The Machine Learning Approach*. 2nd. ed. [S.l.]: The MIT Press, 2001. ISBN 9780262025065.
- 17 SAKAKIBARA, Y. et al. *Stochastic Context-Free Grammars for Modeling RNA*. 1993.
- 18 SAKAKIBARA, Y. et al. *The Application of Stochastic Context-Free Grammars to Folding, Aligning and Modeling Homologous RNA Sequences*. 1993.
- 19 NUNES, F. d. L. d. S. Introdução ao processamento de imagens médicas para auxílio ao diagnóstico. *Atualizações em Informática*. 1 ed. Rio de Janeiro: PUC-Rio, v. 1, p. 73–126, 2006.
- 20 GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 3rd. ed. [S.l.]: Prentice Hall, 2007. ISBN 9780131687288.
- 21 SAMETE, H. A tutorial on quadtree research. *Multiresolution Image Processing and Analysis*, p. 212–223, 1994.
- 22 AIZAWA, K.; NAKAMURA, A. Parsing of two-dimensional images represented by quadtree adjoining grammars. *Journal of The Pattern Recognition Society*, p. 277–294, 1999.
- 23 ZHU, S.-C.; MUMFORD, D. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, Now Publishers Inc., Hanover, MA, USA, v. 2, n. 4, p. 259–362, jan 2006. ISSN 1572-2740.
- 24 MAFRA, S. N.; TRAVASSOS, G. H. Estudos primários e secundários apoiando a busca por evidência em engenharia de software. *COOPE-UFRJ*, p. 1–32, 2006.
- 25 PEDRO, R. W. D.; NUNES, F. L. S.; MACHADO, A. L. Using grammars for pattern recognition in images: a systematic review. *Computing Surveys - CSUR*, 2013.
- 26 HAN, F.; ZHU, S.-C. Bottom-up/top-down image parsing by attribute graph grammar. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Beijing, China: [s.n.], 2005. v. 2, p. 1778 –1785 Vol. 2. ISSN 1550-5499.
- 27 FERREIRA, M.; SANTOS, C.; MONTEIRO, J. Cork parquet quality control vision system based on texture segmentation and fuzzy grammar. *Industrial Electronics, IEEE Transactions on*, IEEE Transactions on Industrial Electronics, v. 56, n. 3, p. 756–765, 2009.
- 28 SCHLECHT, J. et al. Inferring grammar-based structure models from 3d microscopy data. In: *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*. Minneapolis, MN, USA: [s.n.], 2007. p. 1 –8. ISSN 1063-6919.
- 29 WU, Y.; BIAN, H. Image segmentation integrating generative and discriminative methods. In: *Web Information Systems and Mining, 2009. WISM 2009. International Conference on*. Shanghai, China: [s.n.], 2009. p. 769 –774.

- 30 WANG, Y.; BAHRAMI, S.; ZHU, S.-C. Perceptual scale space and its applications. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Beijing, China: [s.n.], 2005. v. 1, p. 58 – 65 Vol. 1. ISSN 1550-5499.
- 31 KANUNGO, T.; MAO, S. Stochastic language models for style-directed layout analysis of document images. *Image Processing, IEEE Transactions on*, v. 12, n. 5, p. 583 – 596, may 2003. ISSN 1057-7149.
- 32 STUCKELBERG, M.; DOERMANN, D. On musical score recognition using probabilistic reasoning. In: *Document Analysis and Recognition, 1999. ICDAR '99. Proceedings of the Fifth International Conference on*. Bangalore, India: [s.n.], 1999. p. 115 –118.
- 33 WANG, W. et al. Hierarchical stochastic image grammars for classification and segmentation. *Image Processing, IEEE Transactions on*, v. 15, n. 10, p. 3033 –3052, oct. 2006. ISSN 1057-7149.
- 34 SISKIND, J. et al. Stochastic grammars for images on arbitrary graphs. *IEEE Workshop on Statistical Signal Processing*, p. 407, 2003.
- 35 SISKIND, J. et al. Spatial random tree grammars for modeling hierarchical structure in images with regions of arbitrary shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 29, n. 9, p. 1504–1518, 2007.
- 36 LIN, L. et al. A stochastic graph grammar for compositional object representation and recognition. *Pattern Recognition*, v. 42, p. 1297–1307, 2009.
- 37 KADIR, T.; BRADY, M. Saliency, scale and image description. *Computer Vision*, p. 83–105, 2008.
- 38 REDDY, H.; KARIBASAPPA, K.; DAMODARAM, A. Probabilistic parser for face detection. In: *Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on*. Delhi, India: [s.n.], 2009. p. 1 –7.
- 39 HAN, F.; ZHU, S.-C. Bottom-up/top-down image parsing with attribute grammar. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 31, n. 1, p. 59 –73, jan. 2009. ISSN 0162-8828.
- 40 ROTHROCK, B.; ZHU, S.-C. Human parsing using stochastic and-or grammars and rich appearances. In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. Barcelona, Catalonia, Spain: [s.n.], 2011. p. 640 –647.
- 41 PENG, S. et al. A database schema for large scale annotated image dataset. In: *Image and Signal Processing, 2008. CISP '08. Congress on*. Sanya, China: [s.n.], 2008. v. 3, p. 57 –62.
- 42 PARAG, T. et al. A grammar for hierarchical object descriptions in logic programs. In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. Providence, RI, USA: [s.n.], 2012. p. 33 –38. ISSN 2160-7508.

- 43 JIN, Y.; GEMAN, S. Context and hierarchy in a probabilistic image model. In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. New York, NY, USA: [s.n.], 2006. v. 2, p. 2145 – 2152. ISSN 1063-6919.
- 44 TYLECEK, R.; SARA, R. Modeling symmetries for stochastic structural recognition. In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. Barcelona, Catalonia, Spain: [s.n.], 2011. p. 632 –639.
- 45 WANG, Q.; JIANG, Z. A grammatical framework for building rooftop extraction. In: *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*. Cape Town, South Africa: [s.n.], 2009. v. 3, p. III–334 –III–337.
- 46 SOLTANPOUR, S.; EBRAHIMNEZHAD, H. Learning novel object parts model for object categorization. In: *Telecommunications (IST), 2010 5th International Symposium on*. Tehran, Iran: [s.n.], 2010. p. 796 –800.
- 47 SAINZ, M.; SANFELIU, A. Learning bidimensional context-dependent models using a context-sensitive language. In: *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*. Vienna, Austria: [s.n.], 1996. v. 4, p. 565 –569 vol.4. ISSN 1051-4651.
- 48 SHET, V. et al. Predicate logic based image grammars for complex pattern recognition. In: *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*. Miami, FL, USA: [s.n.], 2009. p. 7. ISSN 2160-7508.
- 49 MAS, J.; SANCHEZ, G.; LLADOS, J. An adjacency grammar to recognize symbols and gestures in a digital pen framework. In: MARQUES, J. S.; BLANCA, N. Perez de la; PINA, P. (Ed.). *Pattern Recognition and Image Analysis*. [S.l.]: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3523). p. 115–122. ISBN 978-3-540-26154-4.
- 50 MAS, J. et al. Representing and parsing sketched symbols using adjacency grammars and a grid-directed parser. In: LIU, W.; LLADOS, J.; OGIER, J.-M. (Ed.). *Graphics Recognition. Recent Advances and New Opportunities*. [S.l.]: Springer Berlin Heidelberg, 2008, (Lecture Notes in Computer Science, v. 5046). p. 169–180. ISBN 978-3-540-88184-1.
- 51 TRZUPEK, M.; OGIELA, M.; TADEUSIEWICZ, R. Image content analysis for cardiac 3d visualizations. In: *Knowledge-Based and Intelligent Information and Engineering Systems*. [S.l.]: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5711). p. 192–199. ISBN 978-3-642-04594-3.
- 52 TRZUPEK, M.; OGIELA, M.; TADEUSIEWICZ, R. Intelligent image content description and analysis for 3d visualizations of coronary vessels. In: NGUYEN, N.; KIM, C.-G.; JANIÁK, A. (Ed.). *Intelligent Information and Database Systems*. [S.l.]: Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6592). p. 193–202. ISBN 978-3-642-20041-0.
- 53 OGIELA, L.; TADEUSIEWICZ, R.; OGIELA, M. Cognitive modeling in medical pattern semantic understanding. In: *Multimedia and Ubiquitous Engineering, 2008. MUE 2008. International Conference on*. Busan, South Korea: [s.n.], 2008. p. 15 –18.

- 54 OGIELA, L.; OGIELA, M.; TADEUSIEWICZ, R. Mathematical linguistics in cognitive medical image interpretation systems. *Journal of Mathematical Imaging and Vision*, Springer US, v. 34, p. 328–340, 2009. ISSN 0924-9907.
- 55 GIDAS, B.; ZELIC, A. Object recognition via hierarchical syntactic models. In: *Digital Signal Processing Proceedings, 1997. DSP 97., 1997 13th International Conference on*. Santorini, Greece: [s.n.], 1997. v. 1, p. 315–318.
- 56 TOSHEV, A.; MORDOHAI, P.; TASKAR, B. Detecting and parsing architecture at city scale from range data. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. San Francisco, CA, USA: [s.n.], 2010. p. 398–405. ISSN 1063-6919.
- 57 GLOMB, P. Image language terminal symbols from feature analysis. In: *Imaging Systems and Techniques, 2007. IST '07. IEEE International Workshop on*. Krakow, Poland: [s.n.], 2007. p. 1–6.
- 58 CHRISTENSEN, H.; MATAS, J.; KITTLER, J. Using grammars for scene interpretation. In: *Image Processing, 1996. Proceedings., International Conference on*. Lausanne, Switzerland: [s.n.], 1996. v. 1, p. 793–796 vol.2.
- 59 LIN, W.-C.; FU, K.-S. A syntactic approach to three-dimensional object recognition. *Systems, Man and Cybernetics, IEEE Transactions on*, IEEE Transactions on Systems, Man and Cybernetics, v. 16, p. 405–422, 1986.
- 60 GAO, J.; DING, X.; ZHENG, J. Image pattern recognition based on examples - a combined statistical and structural-syntactic approach. In: FERRI, F. et al. (Ed.). *Advances in Pattern Recognition*. [S.l.]: Springer Berlin Heidelberg, 2000, (Lecture Notes in Computer Science, v. 1876). p. 57–66. ISBN 978-3-540-67946-2.
- 61 HINGWAY, S.; BHURCHANDI, K. A simple graph theoretic approach for object recognition. In: *Emerging Trends in Engineering and Technology (ICETET), 2011 4th International Conference on*. Port Louis, Mauritius: [s.n.], 2011. p. 200–205. ISSN 2157-0477.
- 62 ZABOLI, H.; RAHMATI, M. An improved shock graph approach for shape recognition and retrieval. In: *Modelling Simulation, 2007. AMS '07. First Asia International Conference on*. Phuket, Thailand: [s.n.], 2007. p. 438–443.
- 63 SIDDIQI, K.; KIMIA, B. A shock grammar for recognition. In: *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*. San Francisco, CA, USA: [s.n.], 1996. p. 507–513.
- 64 SIDDIQI, K. et al. Shock graphs and shape matching. In: *Computer Vision, 1998. Sixth International Conference on*. Bombay, India: [s.n.], 1998. p. 222–229.
- 65 FERREIRA, M.; SANTOS, C.; MONTEIRO, J. Texture cue based tracking system using wavelet transform and a fuzzy grammar. In: *Industrial Informatics, 2007 5th IEEE International Conference on*. Vienna, Austria: [s.n.], 2007. v. 1, p. 393–398. ISSN 1935-4576.
- 66 FERREIRA, M.; SANTOS, C.; MONTEIRO, J. Texture segmentation based on fuzzy grammar for cork parquet quality control. In: *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*. Vigo, Spain: [s.n.], 2007. p. 1832–1837.

- 67 LUO, P. et al. Hierarchical 3d perception from a single image. In: *Proceedings of the 16th IEEE international conference on Image processing*. Piscataway, NJ, USA: IEEE Press, 2009. (ICIP'09), p. 4209–4212. ISBN 978-1-4244-5653-6.
- 68 SUBRAMANIAN, K.; MARY, A.; DERSANAMBIKA, K. Splicing array grammar systems. In: HUNG, D.; WIRSING, M. (Ed.). *Theoretical Aspects of Computing-ICTAC 2005*. [S.l.]: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3722). p. 125–135. ISBN 978-3-540-29107-7.
- 69 PRUSINKIEWICZ, P.; LINDENMAYER, A.; HANAN, J. Development models of herbaceous plants for computer imagery purposes. *SIGGRAPH Comput. Graph.*, ACM, New York, NY, USA, v. 22, n. 4, p. 141–150, jun. 1988. ISSN 0097-8930.
- 70 SUN, R. et al. Image-based lightweight tree modeling. In: *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*. New York, NY, USA: ACM, 2009. (VRCAI '09), p. 17–22. ISBN 978-1-60558-912-1.
- 71 HEMBERG, M.; OREILLY, U.-M. Extending grammatical evolution to evolve digital surfaces with genr8. In: KEIJZER, M. et al. (Ed.). *Genetic Programming*. [S.l.]: Springer Berlin Heidelberg, 2004, (Lecture Notes in Computer Science, v. 3003). p. 299–308. ISBN 978-3-540-21346-8.
- 72 QU, H. et al. Automata-based l-grammar extraction from multiple images for virtual plants. In: *Bio-Inspired Computing: Theories and Applications, 2008. BICTA 2008. 3rd International Conference on*. Adelaide, South Australia: [s.n.], 2008. p. 89–96.
- 73 HAMDI, S.; ABDALLAH, A. B.; BEDOUI, M. Grammar-based image segmentation and automatic area estimation. In: *Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean*. Yasmine Hammamet, Tunisia: [s.n.], 2012. p. 356–359. ISSN 2158-8473.
- 74 SHILMAN, M.; LIANG, P.; VIOLA, P. Learning nongenerative grammatical models for document analysis. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Beijing, China: [s.n.], 2005. v. 2, p. 962–969. ISSN 1550-5499.
- 75 KONG, J. et al. Web interface interpretation using graph grammars. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, v. 42, n. 4, p. 590–602, july 2012. ISSN 1094-6977.
- 76 MAO, S.; ROSENFELD, A.; KANUNGO, T. Stochastic attributed k-d tree modeling of technical paper title pages. In: *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*. Barcelona, Catalonia, Spain: [s.n.], 2003. v. 1, p. I – 533–6 vol.1. ISSN 1522-4880.
- 77 WANG, W. et al. Classification of images using spatial random trees. In: *Statistical Signal Processing, 2005 IEEE/SP 13th Workshop on*. Novosibirsk, Russia: [s.n.], 2005. p. 449–452.
- 78 YAO, B.; YANG, X.; WU, T. Image parsing with stochastic grammar: The lotus hill dataset and inference scheme. In: *Computer Vision and Pattern Recognition Workshops*,

2009. *CVPR Workshops 2009. IEEE Computer Society Conference on*. Miami, FL, USA: [s.n.], 2009. p. 8. ISSN 2160-7508.
- 79 NUNES, F. L. S.; SCHIABEL, H.; GOES, C. E. Contrast enhancement in dense breast images to aid clustered microcalcifications detection. *Journal of Digital Imaging*, v. 20, n. 1, p. 53–66, March 2007.
- 80 DOI, K. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computerized Medical Imaging and Graphics*, v. 31(4-5), p. 191–211, 2007.
- 81 BOHM, C.; BERCHTOLD, S.; KEIM, D. A. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 33, n. 3, p. 322–373, 2001. ISSN 0360-0300.
- 82 PETRAKIS, E.; FALOUTSOS, C.; LIN, K.-I. Imagemap: an image indexing method based on spatial similarity. *Knowledge and Data Engineering, IEEE Transactions on*, v. 14, n. 5, p. 979–987, Sep/Oct 2002. ISSN 1041-4347.
- 83 SAKAKIBARA, Y. Learning context-free grammars using tabular representations. *Pattern Recogn.*, Elsevier Science Inc., New York, NY, USA, v. 38, n. 9, p. 1372–1383, set. 2005. ISSN 0031-3203.
- 84 ANGLUIN, D. Computational learning theory: survey and selected bibliography. In: *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 1992. (STOC '92), p. 351–369. ISBN 0-89791-511-9.
- 85 RON, D.; SINGER, Y.; TISHBY, N. On the learnability and usage of acyclic probabilistic finite automata. In: *Proceedings of the eighth annual conference on Computational learning theory*. New York, NY, USA: ACM, 1995. (COLT '95), p. 31–40. ISBN 0-89791-723-5.
- 86 SAKAKIBARA, Y.; MURAMATSU, H. Learning context-free grammars from partially structured examples. In: OLIVEIRA, A. (Ed.). *Grammatical Inference: Algorithms and Applications*. [S.l.]: Springer Berlin Heidelberg, 2000, (Lecture Notes in Computer Science, v. 1891). p. 229–240. ISBN 978-3-540-41011-9.
- 87 FERNAU, H. Parallel grammars: A phenomenology. *Grammars*, Kluwer Academic Publishers, v. 6, p. 25–87, 2003. ISSN 1386-7393.
- 88 FU, K. *Syntactic pattern recognition and applications*. [S.l.]: Prentice-Hall, 1982. (Prentice-Hall advanced reference series: Computer science). ISBN 9780138801205.
- 89 EARLEY, J. An efficient context-free parsing algorithm. *Commun. ACM*, ACM, New York, NY, USA, v. 13, n. 2, p. 94–102, fev. 1970. ISSN 0001-0782.