



**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CURITIBA**

GERÊNCIA DE PESQUISA E PÓS-GRADUAÇÃO

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL - CPGEI**

Hugo Alberto Perlin

**Reconhecimento de Objetos em Imagens Digitais
Utilizando Otimização por Enxame de Partículas**

DISSERTAÇÃO DE MESTRADO

Curitiba,
27 de Março de 2009.

Livros Grátis

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

Milhares de livros grátis para download.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

DISSERTAÇÃO
apresentada à UTFPR
para obtenção do grau de

MESTRE EM CIÊNCIAS

por

Hugo Alberto Perlin

**Reconhecimento de Objetos em Imagens Digitais
Utilizando Otimização por Enxame de Partículas**

Banca Examinadora:

Presidente e Orientador:

Prof^a. Dr^a. Tania Mezzadri Centeno

UTFPR

Examinadores:

Prof. Dr. Heitor Silvério Lopes

UTFPR

Prof. Dr. Hugo Vieira Neto

UTFPR

Prof. Dr. Alessandro Lameiras Koerich

PUCPR

Curitiba,
27 de Março de 2009.

HUGO ALBERTO PERLIN

**Reconhecimento de Objetos em Imagens Digitais
Utilizando Otimização por Enxame de Partículas**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do grau de “Mestre em Ciências” - Área de Concentração: Informática Industrial.

Orientador: Prof^a. Dr^a. Tania Mezzadri Centeno

Co-Orientador: Prof. Dr. Heitor Silvério Lopes

Curitiba

2009

Ficha catalográfica elaborada pela Biblioteca da UTFPR – Campus Curitiba

P451r	<p>Perlin, Hugo Alberto Reconhecimento de objetos em imagens digitais utilizando otimização por enxame de partículas / Hugo Alberto Perlin. – 2009. xxv, 73 p. : il. ; 30 cm</p> <p>Orientador: Tania Mezzadri Centeno Co-orientador: Heitor Silvério Lopes Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Área de Concentração: Informática Industrial, Curitiba, 2009 Bibliografia: p. 69-73</p> <p>1. Processamento de imagens – Técnicas digitais. 2. Visão por computador. 3. Sistemas de reconhecimento de padrões. 4. Otimização por enxame de partículas. I. Centeno, Tania Mezzadri, orient. II. Lopes, Heitor Silvério, co-orient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Área de Concentração em Informática Industrial. IV. Título.</p> <p>CDD 621.3</p>
-------	--

*Dedico esta dissertação a meus pais,
que me deram o amor e a educação
necessários para chegar até aqui,
e também a minha noiva, pelo amor,
carinho e apoio sempre.*

AGRADECIMENTOS

Os meus sinceros agradecimentos são para todas as pessoas que de alguma forma ou de outra me deram apoio e incentivo para a conclusão deste trabalho. Em especial gostaria de agradecer à:

– Professora Doutora Tania Mezzadri Centeno, por aceitar me orientar e auxiliar no decorrer do trabalho;

– Professor Doutor Heitor Silvério Lopes, que tornou-se o co-orientador do trabalho e se mostrou de fundamental importância para a conclusão do mesmo, agradeço a paciência e o incentivo;

– Ao Rafael Betito, amigo e colega de mestrado com quem tive várias discussões de idéias, algumas plausíveis, outras, digamos, absurdas;

– Ao casal de grandes amigos Wagner e Luciana, pelos momentos de descontração e também pela ajuda na organização das idéias;

– A minha noiva, pela leitura e revisão do texto;

– Ao pessoal do laboratório de Bio-Informática, pelas conversas e ajuda operacional;

– Aos colegas do IFPR Campus Paranaguá, pelo apoio.

*“Existe uma coisa que uma longa existência me ensinou:
toda a nossa ciência, comparada à realidade,
é primitiva e inocente; e, portanto,
é o que temos de mais valioso.”*

Albert Einstein

SUMÁRIO

LISTA DE FIGURAS	xiii
LISTA DE TABELAS	xvii
LISTA DE ABREVIATURAS E SIGLAS	xix
LISTA DE SÍMBOLOS	xxi
RESUMO	xxiii
ABSTRACT	xxv
1 INTRODUÇÃO	1
1.1 MOTIVAÇÃO	1
1.2 OBJETIVOS	2
1.3 ESTRUTURA DO TRABALHO	2
2 FUNDAMENTAÇÃO TEÓRICA	5
2.1 OTIMIZAÇÃO E MÉTODOS DE OTIMIZAÇÃO	5
2.1.1 Computação Bioinspirada	7
2.1.1.1 Algoritmos Evolucionários	8
2.1.1.2 Inteligência Coletiva	9
2.2 <i>PARTICLE SWARM OPTIMIZATION</i>	10
2.2.1 Breve Histórico	10
2.2.2 O Método	11
2.2.3 Outras Abordagens	14
2.2.4 Aplicações do PSO	14
2.3 CONCEITOS DE PROCESSAMENTO DE IMAGENS	15
2.3.1 Modelo RGB	15
2.3.2 Transformações Geométricas	15
2.3.3 Histograma	17
2.3.4 Ruído	18
2.4 SISTEMA DE VISÃO COMPUTACIONAL	19

2.5	RECONHECIMENTO DE OBJETOS	21
2.5.1	Estratégias de Reconhecimento	21
2.5.2	Medidas de Similaridade	23
2.5.2.1	Norma L1	23
2.5.2.2	Informação Mútua	23
2.6	TRABALHOS CORRELATOS	25
3	METODOLOGIA	27
3.1	RECONHECIMENTO DE OBJETOS COMO PROBLEMA DE OTIMIZAÇÃO	27
3.1.1	Descrição do <i>Template Matching</i>	29
3.2	MEDIDAS DE SIMILARIDADE	30
3.2.1	Norma L1	30
3.2.2	Informação Mútua	31
3.2.3	Análise de Superfície de <i>Fitness</i>	33
3.3	APLICAÇÃO DO PSO	37
3.3.1	Detalhes de Implementação	37
3.3.2	Ajuste de parâmetros do PSO	39
3.4	CONJUNTO DE TESTES	44
3.5	COMPARAÇÃO COM OUTRA ABORDAGEM EVOLUCIONÁRIA	44
4	EXPERIMENTOS E RESULTADOS	47
4.1	EXPERIMENTOS PRELIMINARES	47
4.1.1	Experimento 1	47
4.1.2	Experimento 2	48
4.2	AVALIAÇÃO DO MÉTODO	48
4.2.1	Sensibilidade à Translação	49
4.2.2	Sensibilidade à Escala e Rotação	51
4.2.3	Sensibilidade a Ruído	52
4.2.4	Sensibilidade à Oclusão	55
4.2.5	Sensibilidade a Contraste e Brilho	56
4.3	COMPARAÇÃO COM OUTRA ABORDAGEM EVOLUCIONÁRIA	59
5	DISCUSSÃO E CONCLUSÕES	65
5.1	ANÁLISE DOS RESULTADOS	65
5.1.1	Experimentos Preliminares	65
5.1.2	Avaliação do Método	65
5.1.3	Comparação com outra Abordagem Evolucionária	67

5.2	CONCLUSÕES	67
5.3	TRABALHOS FUTUROS	68
	REFERÊNCIAS BIBLIOGRÁFICAS	69

LISTA DE FIGURAS

1	Exemplos de ótimos globais e locais encontrados na função $\cos(3\pi x)/x$	6
2	Cubo RGB.	16
3	Exemplo de uma imagem transladada.	16
4	Exemplo de uma imagem com escala.	17
5	Exemplo de uma imagem com rotação.	17
6	Exemplo de uma imagem (a) e seu respectivo histograma. Retirado de (MCHUGH, 2009).	18
7	Exemplos de adição de ruído impulsivo S&P (b) e ruído gaussiano (c) em uma imagem (a).	19
8	Exemplo de arquitetura de um sistema de visão computacional.	20
9	Exemplo de detecção e reconhecimento de placa de veículos. Retirado de (ZIMMERMANN, 2009).	22
10	Exemplo de <i>template matching</i> . Uma imagem contendo o objeto (modelo) é sobreposta sobre a imagem de paisagem e é deslizada sobre todos os pixels da mesma. No ponto onde o valor da medida de similaridade for o melhor significa que ocorreu um casamento entre o objeto e a paisagem, o que implica no reconhecimento do objeto naquela imagem de paisagem, representado por um retângulo amarelo na imagem de paisagem.	28
11	Passo e dados necessários para tornar o <i>template matching</i> invariante à translação, à rotação e à escala.	30
12	Exemplo de duas imagens com 8 níveis de cinza	32
13	Histograma conjunto entre as imagens das figuras 12(a) e 12(b).	33
14	(a) Cena complexa utilizada como IP. (b) IR a ser reconhecida.	34
15	Superfície de <i>fitness</i> para a função Norma L1.	35
16	Superfície de <i>fitness</i> para a função Informação Mútua.	36
17	(a) IP (800 x 600) utilizada para o Teste 1 e Teste 2. (b) IR para o Teste 1. (c) IR para o Teste 2.	40
18	(a) IP (1024 x 768) para o Teste 3. (b) IR para o Teste 3.	41
19	(a) IP (1024 x 768) para o Teste 4. (b) IR para o Teste 4.	41
20	Distribuição da ocorrência de valores dos parâmetros do PSO para 4 problemas diferentes.	42

21	Exemplos de imagens de paisagem contidas no banco <i>McGill Calibrated Colour Image Database</i>	45
22	Exemplos de objetos contidos no banco <i>COIL-100</i>	45
23	Uma instância do problema de reconhecimento de objetos gerada sinteticamente. Esta imagem possui uma série de objetos de interesse. Nesta imagem foi introduzido ruído Gaussiano com sigma igual a 0,5, simulando uma situação que pode ocorrer na aquisição da imagem.	46
24	(a) IR. (b) IP com o objeto desejado transladado, rotacionado e degradado por ruído gaussiano com média 0 e desvio padrão 0,5. (c) Mesma IP degradada por ruído gaussiano com média 0 e desvio padrão 2,0.	47
25	(a) IR. (b) IP. (c) Recorte da solução encontrada pelo PSO (c), quando foi utilizada a IR (a) e a IP (b). (d) Diferença entre a solução encontrada e a IR (a).	48
26	(a,b,c) Imagens de paisagem coloridas utilizada nos testes. (d,e) Imagens de referência que contem os objetos desejados.	50
27	(a,b,c) Imagens de paisagem em níveis de cinza utilizada nos testes. (d,e) Imagens de referência em níveis de cinza.	50
28	(a) Conjunto de IPs com variação no fator de escala aplicado nos objetos inseridos. (b) Média entre os erros relativos fornecidos pelos três conjuntos de testes, para os diferentes fatores de escala, no reconhecimento das IR Obj 1 e Obj 2, tanto coloridas quanto em níveis de cinza.	53
29	(a) Conjunto de IPs com variação no ângulo de rotação aplicado nos objetos inseridos. (b) Média entre os erros relativos fornecidos pelos três conjuntos de testes, para os diferentes ângulos de rotação, no reconhecimento das IR Obj 1 e Obj 2, tanto coloridas quanto em níveis de cinza.	54
30	(a) Média dos erros relativos para o reconhecimento da IR Obj. 1 colorida e em níveis de cinza. (b) Média dos erros relativos para o reconhecimento da IR Obj. 2 colorida e em níveis de cinza.	55
31	(a) Uma das imagens de paisagem usada no experimento com oclusão de 25% da superfície dos objetos. (b) Detalhe da simulação de oclusão nos objetos.	57
32	(a) Conjunto de IPs com variação de brilho. (b) Média entre os erros relativos fornecidos pelos três conjuntos de testes, para os diferentes níveis de brilho, no reconhecimento das IR Obj 1 e Obj 2, tanto coloridas quanto em níveis de cinza.	58

33	(a) Conjunto de IPs com variação de contraste. (b) Média entre os erros relativos fornecidos pelos três conjuntos de testes, para os diferentes níveis de contraste, no reconhecimento das IR Obj 1 e Obj 2, tanto coloridas quanto em níveis de cinza.	59
34	IR (a) e IP (c) utilizada em um dos testes de comparação. O retângulo destaca a solução encontrada pelo PSO.	61
35	IR (a) e IP (c) utilizada em um dos testes de comparação. O retângulo destaca a solução encontrada pelo PSO.	62
36	IR (a) e IP (c) utilizada em um dos testes de comparação. O retângulo destaca a solução encontrada pelo PSO.	63
37	Quantidade de avaliações de <i>fitness</i> necessárias para os três métodos determinarem a melhor solução.	64

LISTA DE TABELAS

1	Informações para o cálculo da entropia da imagem da figura 12(a).	32
2	Informações para o cálculo da entropia da imagem da figura 12(b).	32
3	Intervalos de valores permitidos para cada um dos parâmetros de uma solução. .	37
4	Intervalo de valores permitidos para a velocidade.	38
5	Conjunto de valores utilizados no experimento fatorial para determinação dos parâmetros do PSO	40
6	Solução encontrada pelo PSO para a IR da figura 24(a) e IP da figura 24(b). . .	48
7	Solução encontrada pelo PSO para a IR da figura 24(a) e IP da figura 24(c). . .	48
8	Média das estatísticas das diferenças entre os valores encontrados pelo PSO e o valor de “ <i>ground truth</i> ” para a IR Obj 1 sendo localizada nas três IPs, tanto colorida quanto em níveis de cinza, na ocorrência de translação.	51
9	Média das estatísticas das diferenças entre os valores encontrados pelo PSO e o valor de “ <i>ground truth</i> ” para a IR Obj 2 sendo localizada nas três IPs, tanto colorida quanto em níveis de cinza, na ocorrência de translação.	51
10	Média dos desvios padrão para cada fator de escala, tanto colorida quanto em níveis de cinza, para as duas IR.	52
11	Média dos desvios padrão para cada fator de escala, tanto colorida quanto em níveis de cinza, para as duas IR.	52
12	Média dos desvios padrão para os níveis de ruído S&P e Gaussiano inseridos nas IPs, tanto coloridas quanto em níveis de cinza, para o reconhecimento da IR Obj 1.	56
13	Média dos desvios padrão para os níveis de ruído S&P e Gaussiano inseridos nas IPs, tanto coloridas quanto em níveis de cinza, para o reconhecimento da IR Obj 2.	56
14	Média das estatísticas das diferenças dos valores encontrados pelo PSO e o valor de “ <i>ground truth</i> ” para a IR Obj 1 ser localizada nas três IPs com oclusão, tanto colorida quanto em níveis de cinza, com oclusão parcial.	56
15	Média das estatísticas das diferenças entre os valores encontrados pelo PSO e o valor de “ <i>ground truth</i> ” para a IR Obj 2 ser localizada nas três IPs com oclusão, tanto colorida quanto em níveis de cinza, com oclusão parcial.	57

16	Média dos desvios padrão para os testes de mudança de brilho, tanto coloridas quanto em níveis de cinza.	60
17	Média dos desvios padrão para os níveis de ruído S&P e Gaussiano inseridos nas IPs, tanto coloridas quanto em níveis de cinza, para o reconhecimento da IR Obj 2.	60
18	Soluções fornecidas pelos métodos para encontrar a IR mostrada na figura 34(a) na IP mostrada na figura 34(b).	61
19	Soluções fornecidas pelos métodos para encontrar a 35(a) na IP 35(b).	62
20	Soluções fornecidas pelos métodos para encontrar a IR mostrada na figura 36(a) na IP mostrada na figura 36(b).	63

LISTA DE ABREVIATURAS E SIGLAS

ACO	<i>Ant Colony Optimization</i>
AE	Algoritmos Evolucionários
CGA	Algoritmo Genético Compacto
emCGA	Algoritmo Genético Compacto com elitismo e mutação
HSI	Matiz, Saturação, Intensidade
HSV	Matiz, Saturação, Valor
IM	Informação Mútua
IP	Imagem de Paisagem
IR	Imagem de Referência
PSO	<i>Particle Swarm Optimization</i>
RGB	<i>Red, Blue, Green</i>
S&P	Ruído Sal-e-Pimenta
SVC	Sistema de Visão Computacional
SVM	<i>Support Vector Machine</i>
TM	<i>Template Matching</i>

LISTA DE SÍMBOLOS

θ	Ângulo de rotação
IP_{rec}	Recorte da imagem de paisagem
m	Altura de uma imagem
n	Largura de uma imagem
s	Fator de escala
x	Valor de translação no eixo x
y	Valor de translação no eixo y

RESUMO

O reconhecimento automático de objetos tem como objetivo fazer com que dispositivos computacionais tenham a capacidade de extração de informação visual do ambiente onde estão inseridos. Uma das formas de tratar o problema de reconhecimento é através da técnica de *Template Matching* visualizada como um problema de otimização global. Neste caso se deseja determinar a posição de um objeto em uma imagem, bem como seu ângulo de rotação e qual o fator de escala utilizado. A otimização por enxame de partículas (*Particle Swarm Optimization* - PSO) é um método bastante difundido na literatura e amplamente empregado em problemas de otimização global e foi utilizado como parte da solução. As medidas de similaridade são de grande importância para o funcionamento do método, assim foram realizados estudos para a escolha de uma boa função. Também foram realizados experimentos para determinar um conjunto de parâmetros adequados para o método de otimização, com o objetivo de melhorar a qualidade das soluções fornecidas. Para determinar a sua eficácia, o método foi testado com imagens coloridas e em níveis de cinza, com translação, rotação, mudança de escala e oclusão parcial dos objetos, bem como presença de ruído e simulação de mudança de luminosidade. Ainda foi feita uma comparação com outra abordagem evolucionária para o reconhecimento de objetos, a fim de verificar o desempenho do método proposto. Os resultados mostraram que o PSO foi capaz de encontrar os objetos predefinidos com bom desempenho, sugerindo que o método proposto é bastante promissor, especialmente no que diz respeito a robustez e simplicidade. Sendo assim, esta solução pode ser uma opção a ser empregada no reconhecimento de objetos em aplicações reais de visão computacional.

PALAVRAS-CHAVE:

Visão Computacional, Reconhecimento de Objetos, Otimização, Particle Swarm Optimization..

ABSTRACT

OBJECT RECOGNITION IN DIGITAL IMAGES USING PARTICLE SWARM OPTIMIZATION

The automatic recognition of objects aims to extract the visual information by computational devices. One way of addressing the problem of object recognition is through the technique of Template Matching viewed as a problem of global optimization. The objective is to determine the position of an object in an image, its rotation angle and the scale factor used. The Particle Swarm Optimization (PSO) method is fairly widespread in the literature and widely used in global optimization problems, and was chosen as part of the solution. The measures of similarity are important to the operation of the method, so studies were performed to select a good function. Also, experiments were performed to determine a set of parameters suitable for the optimization method, with the aim of improving the quality of solutions provided. To determine its effectiveness, the method was tested with color and gray-level images, with translation, rotation, scale change and partial occlusion of objects, as well the presence of noise and change of brightness in the images. Moreover, a comparison was made with another evolutionary approach to object recognition, to verify the performance of the proposed method. The results showed that the PSO was able to find the predefined objects with high performance, suggesting that the proposed method is very promising, especially with respect to robustness and simplicity. So this may be an option to be used in real-world object recognition applications of computer vision.

KEY-WORDS:

Computer Vision, Object Recognition, Optimization, Particle Swarm Optimization.

CAPÍTULO 1

INTRODUÇÃO

Uma pessoa é capaz de detectar e reconhecer objetos a partir de um conjunto de características que os representam, tais como cores, texturas, formas geométricas, entre outras. Ao olhar para uma cena, uma pessoa busca extrair características presentes naquela imagem e compara com características memorizadas, caso exista um objeto que a pessoa conheça este será identificado.

O processo de localização e reconhecimento de um objeto por uma pessoa é feito quase que instantaneamente. Entretanto, quando se deseja que esta tarefa seja realizada por um dispositivo computacional, o problema passa a ser considerado de grande complexidade.

O reconhecimento automático de objetos tem como objetivo fazer dispositivos computacionais tenham a capacidade de extrair, de forma eficiente, a informação visual do ambiente onde estão inseridos. Esta tarefa é uma parte importante do processo de extração de informação a partir de imagens, onde tal informação é utilizada na tomada de decisões.

Inúmeros métodos podem ser empregados para resolver computacionalmente este problema de reconhecimento de padrões. Pode-se citar o uso de redes neurais, máquinas de vetores de suporte (SVM), métodos estocásticos e algoritmos evolucionários. Estes últimos possuem certa robustez para lidar com problemas de otimização global, que é como a detecção de objetos é tratada neste trabalho.

Diversos métodos de computação evolucionária são conhecidos, entre eles estão os algoritmos genéticos, evolução diferencial, colônia de formigas e a otimização por enxame de partículas (PSO - do inglês *Particle Swarm Optimization*). O PSO é um método que foi concebido com base no comportamento coletivo de bandos de pássaros e cardumes de peixes. Este método tornou-se bastante difundido e amplamente utilizado em diversas áreas, tanto pela facilidade de implementação quanto pelos bons resultados fornecidos.

1.1 MOTIVAÇÃO

Com a difusão de equipamentos para captura de imagens, uma quantidade considerável de informação visual é gerada. Diante de um grande banco de imagens pode ser necessário saber quais das imagens contêm um determinado objeto de interesse. A automação de processos como busca de objetos, ou mesmo a leitura automática de endereços de correspondência, requer o emprego de técnicas de processamento de imagens e método de inteligência artificial capazes

de realizar as tarefas necessárias, fornecendo resultados satisfatórios em tempo hábil.

Além disso, a automação de processos industriais necessita que dispositivos computacionais sejam programados a fim de que sejam capazes de tomar decisões baseados em um conjunto de informações. Em algumas tarefas, grande parte da informação utilizada para o cumprimento do objetivo pode ser obtida de forma visual. Assim, deve ser empregado algum sistema de visão capaz de prover as condições mínimas para a aquisição e processamento da informação visual de forma eficiente e eficaz.

Como já mencionado, o reconhecimento de objetos é parte fundamental no processamento da informação visual. Desta forma a escolha de métodos eficazes de reconhecimento é essencial para garantir a qualidade da informação visual extraída. O uso de técnicas de processamento de imagens juntamente com métodos de inteligência artificial é a forma mais comum de solução para o problema de reconhecimento.

O emprego do PSO neste trabalho tem como intuito desenvolver uma aplicação eficiente e robusta para realizar a detecção de objetos em imagens, visto que este problema possui relevância em várias áreas do conhecimento e também possui grande complexidade.

1.2 OBJETIVOS

O objetivo geral deste trabalho é implementar o método de otimização por enxame de partículas e aplicá-lo ao problema de detecção de objetos em imagens digitais. Como objetivos específicos têm-se:

- Estudo da escolha de funções de similaridade entre imagens para serem utilizadas pelo método de otimização;
- Estudo de sensibilidade dos parâmetros do método nos resultados;
- Estudo do desempenho do PSO quando da existência nas imagens, de transformações geométricas; mudança de brilho e contraste; presença de ruído; e oclusão;
- Comparação do PSO com outra abordagem evolucionária, chamada de Algoritmo Genético Compacto com operadores de elitismo e mutação, aplicada ao reconhecimento de objetos.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está dividido da seguinte forma. No capítulo 2 é feita uma introdução aos assuntos abordados, esclarecendo alguns conceitos necessários para o bom entendimento do

restante do texto. Neste mesmo capítulo são relatados alguns trabalhos encontrados na literatura que utilizam algoritmos evolucionários para a solução do problema de reconhecimento de objetos. A metodologia de estudo e implementação do método para reconhecimento é mostrada no capítulo 3. Os resultados dos testes realizados para a validação e verificação da qualidade das soluções fornecidas pelo método proposto são expostos no capítulo 4. A discussão dos resultados bem como as conclusões obtidas são relatadas no capítulo 5.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é feita uma revisão sobre os principais conceitos que fundamentam e delimitam o escopo deste trabalho. O intuito aqui é fazer com que o leitor ainda não familiarizado com termos utilizados pela área de otimização e processamento de imagens consiga ter a base necessária para entender a continuação da dissertação. Outro objetivo aqui é fazer um apinhado geral sobre algumas linhas de pesquisa na área de reconhecimento de objetos utilizando métodos de otimização.

2.1 OTIMIZAÇÃO E MÉTODOS DE OTIMIZAÇÃO

O termo otimização é utilizado para definir uma seqüência de passos realizada para encontrar o melhor valor de uma dada função de custo ou objetivo, respeitando restrições impostas pelo problema. Assim, vários problemas ou processos podem ser formulados através da uma função e solucionados como um problema de otimização (EBERHART; SHI; KENNEDY, 2001).

Um conjunto de valores para as variáveis da função, que satisfaçam as restrições, é chamado de solução candidata. O conjunto de possíveis valores que as variáveis podem assumir é chamado de espaço de busca, sendo que a quantidade de dimensões deste espaço depende do número de variáveis que compõem a função objetivo. A melhor configuração de valores é chamada de solução ótima, pois quando é empregada na função fornece o menor valor da função dentre todos os possíveis valores (para um problema de minimização), ou o maior valor da função (para um problema de maximização) (HAUPT; HAUPT, 2004).

Como um problema de maximização pode ser transformado em um problema de minimização, e vice-versa, uma definição formal para um problema de minimização é dada como (OMRAN, 2004):

Seja uma função $f : S \rightarrow \mathfrak{R}$, onde $S \subseteq \mathfrak{R}^N$, e N é a dimensão do espaço de busca S , encontre $w^* \in S$, tal que $f(w^*) \leq f(w)$, $\forall w \in S$

O fato de $f(w^*)$ ser o menor valor dentre todos os possíveis, implica que este ponto do espaço N-dimensional é o ótimo global da função, não existindo nenhum valor melhor do que aquele em todo o espaço de busca. Além do ótimo global existem os ótimos locais que são valores para a função que representam o menor valor numa determinada vizinhança. Assim, para uma determinada região do espaço de busca existe um valor que é o melhor, mas quando

analisado o espaço de busca como um todo, aquele ponto pode não ser o ótimo global. Pode-se então dizer que um ótimo global é um ótimo local, mas não o contrário. A figura 1, apresenta um mínimo local e um mínimo global.

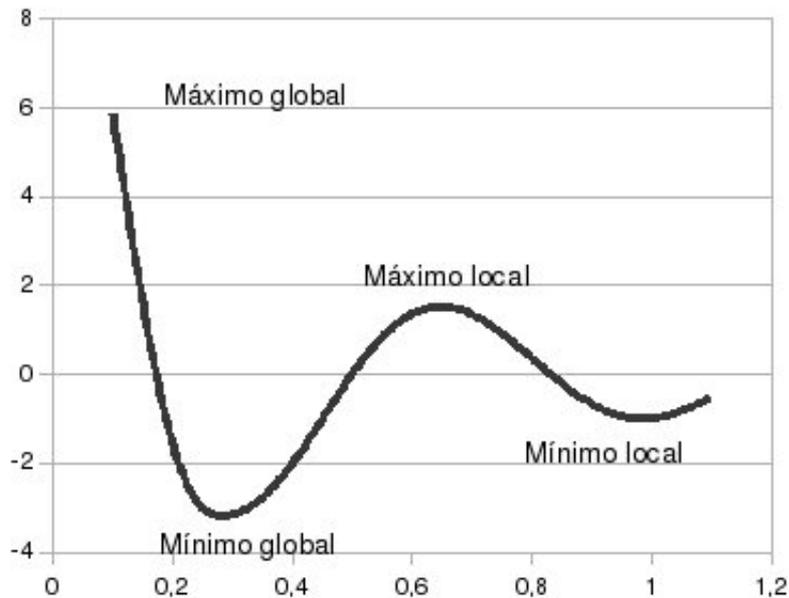


Figura 1: Exemplos de ótimos globais e locais encontrados na função $\cos(3\pi x)/x$

Quando é modelado um problema baseado em otimização, geralmente não se sabe qual é a solução ótima para o problema. Sendo assim, é necessário buscar dentre todas as possíveis soluções para o problema, aquela que melhor o resolva. O processo utilizado para a busca do mínimo global é chamado de algoritmo de otimização global (OMRAN, 2004).

Existem basicamente duas classes de métodos para a resolução de problemas de otimização, que são a classe dos métodos exatos e a dos métodos heurísticos ou aproximados (PINTER, 2002).

O método exato mais simples para a determinação do ótimo global é através da busca exaustiva, ou também chamado de força bruta, onde todas as possíveis soluções pertencentes ao espaço de busca são testadas, e então o melhor valor pode ser encontrado. Para funções com pequenos espaços de busca, esta forma de otimização pode ser conveniente. Mas como a maior parte dos problemas de otimização global é classificada como NP - Difícil, o emprego de busca exaustiva não é adequado, visto que o tamanho do espaço de busca cresce exponencialmente de acordo com o tamanho do problema (NIEVERGELT, 2000).

Os métodos aproximados procuram encontrar soluções o mais próximas possível do ótimo global em tempo polinomial, utilizando-se da premissa que boas soluções estão sempre perto de outras soluções boas. Baseado neste conhecimento entende-se que não é necessário que

todo o espaço de busca seja visitado para que uma solução muito perto do ótimo seja encontrada. Um problema dos algoritmos aproximados é que eles não garantem encontrar a solução ótima.

Uma forma de implementação de um método aproximado é através do uso de regras probabilísticas, que fazem com que o método tenha um comportamento não determinístico durante o processo de busca, os quais são conhecidos como estocásticos. Pelo fato da existência de certo grau de aleatoriedade no processo de busca, o método, quando preso em um ótimo local, pode ter a oportunidade de sair daquele ponto e continuar a busca. Estes métodos não dependem de informações adicionais além daquela fornecida pela função objetivo, tornando-os bastante simples de implementar e entender (TSANG, 1993).

Dentre algoritmos estocásticos, pode-se citar a *Tabu Search*, o *Hill-Climbing* e o *Simulated-Annealing*. Hoos e Stützle (2005) são uma boa fonte de informações sobre estes métodos. Uma linha de desenvolvimento de métodos estocásticos de busca que vem crescendo nos últimos tempos é a computação bioinspirada onde, a partir da observação da natureza, idéias para a criação de métodos de busca são implementadas em computadores e utilizadas para a resolução dos mais variados tipos de problemas.

2.1.1 Computação Bioinspirada

Nos últimos anos surgiram métodos que se enquadram na classe de aproximados e que utilizam conceitos encontrados na natureza. Este paradigma de desenvolvimento de algoritmos é chamado de computação natural, ou computação biologicamente inspirada.

O fato de seres vivos usarem técnicas e conhecimentos para sobreviver, seja fugindo de predadores, buscando alimento ou resolvendo uma série de problemas complexos, é uma fonte de inspiração para os pesquisadores das mais diversas áreas. Isso se deve ao fato de muitos acreditarem que os milhões de anos de evolução das espécies presentes na natureza permitiram o desenvolvimento e utilização de uma série de mecanismos, ferramentas e processos que levaram essas espécies a sobreviver (PAUN, 2005).

Duas linhas básicas de desenvolvimento de métodos e processos bioinspirados são a computação inspirada na natureza e a simulação ou emulação de fenômenos naturais. No primeiro, os métodos são baseados em observações do comportamento de organismos e sociedades, gerando assim, inspiração para o desenvolvimento de sistemas que visam à solução de problemas complexos. Enquadram-se nesta categoria os algoritmos evolucionários (AE) , a neurocomputação, os sistemas imunológicos artificiais e a inteligência coletiva (*swarm intelligence*). A segunda linha trata da simulação de processos e eventos que ocorrem na natureza tentando recriar virtualmente o comportamento dos seres, permitindo assim uma melhor compreensão da natureza. Um dos resultados desta segunda linha é a criação de sistemas de vida artificial, ou seja, sistemas que imitam a vida como os seres humanos conhecem. (CHEN,

2007).

O foco deste trabalho é a computação inspirada na natureza. Mais especificamente, o método utilizado neste trabalho encaixa-se na classe de inteligência coletiva. Muitos conceitos utilizados pela inteligência coletiva têm semelhanças com AEs. Assim, será feita uma breve introdução sobre AEs.

2.1.1.1 Algoritmos Evolucionários

A evolução é um processo de modificação genética dos seres vivos, quando necessária sua adaptação às mudanças do meio ambiente onde os mesmos estão inseridos. Ela ocorre por meio de cruzamento entre indivíduos de uma população (recombinação genética), bem como por meio de mutações genéticas, permitindo que os organismos desenvolvam novos mecanismos e habilidades para solucionar problemas. Os olhos da águia, o sonar dos morcegos, o olfato dos cães, o cérebro humano são resultados de modificações necessárias para a adaptação dos seres a problemas impostos pela natureza (CASTRO, 2002) (RAIDL, 2005).

Além dos dois mecanismos evolucionários citados anteriormente, o cruzamento e a mutação, a evolução das espécies também conta com um terceiro mecanismo, que é a seleção natural. A teoria da evolução de Darwin (DARWIN, 1859), sugere que os organismos melhor adaptados ao ambiente possuem uma maior possibilidade de sobrevivência, e esta adaptação será repassada aos seus descendentes através da herança genética. Assim, dois indivíduos aptos, ao se reproduzirem, transmitem a sua carga genética, permitindo que a nova também seja apta.

Os AE utilizam os mesmos conceitos e mecanismos ou operadores genéticos para o desenvolvimento de métodos de otimização. Assim como na natureza, os algoritmos evolucionários utilizam populações de indivíduos no processo de busca, na qual cada um dos indivíduos é uma solução candidata à resolução do problema. A idéia é manter esta população em constante evolução, utilizando para isso os operadores genéticos. A qualidade de cada um dos indivíduos é avaliada através do valor da função de *fitness*. Uma função de *fitness* é uma equação matemática que agrega a função objetivo a ser otimizada juntamente com penalidades ou recompensas, dependendo da configuração da solução candidata. Neste trabalho e em alguns outros, a função de *fitness* é composta somente pela função objetivo. Assim, neste trabalho o termo função objetivo ou função de *fitness* é usado indiscriminadamente para representar a qualidade de uma solução candidata durante o processo de busca.

Como já explicitado anteriormente, os AE pertencem à classe de métodos de otimização inspirados pela natureza. Estes, por sua vez, são classificados como métodos estocásticos. Portanto, os algoritmos evolucionários também podem possuir dentro de seu processo de busca, um comportamento aleatório.

A principal diferença entre os AE e os outros algoritmos estocásticos, está no fato de que os primeiros mantêm durante o processo de busca um conjunto finito de soluções candidatas do problema, o qual é chamado de população. Através da evolução desta população, com o decorrer do tempo, novas regiões do espaço de busca são exploradas, procurando localizar pontos que possuam os melhores valores para a função objetivo.

Um AE tem sua população iniciada de forma aleatória ou através de heurísticas de construção de soluções candidatas. Após a inicialização de cada indivíduo, entra-se em um laço de repetição que ocorre até que um determinado critério de parada seja atingido, o qual pode ser por qualidade de solução ou por quantidade de iterações. Dentro do laço de repetição, onde a evolução realmente ocorre, os indivíduos têm sua qualidade avaliada através do cálculo da função de *fitness*. Com base nesta qualidade e através de um mecanismo aleatório, os indivíduos mais aptos são selecionados para a geração de uma nova população de descendentes. Parte desses novos indivíduos pode sofrer algum tipo de mutação em seus genes, gerando assim, pequenas modificações nas soluções candidatas. De posse da nova população, o processo de evolução recomeça (FONSECA; FLEMING, 1995).

O processo de busca através de um AE pode ser sintetizado nos seguintes passos:

1. Inicialize a população com valores aleatórios que representem possíveis soluções para o problema;
2. Avalie cada solução;
3. Baseado em algum critério de qualidade, selecione as melhores soluções;
4. Gere uma nova população a partir das soluções selecionadas;
5. Volte ao passo 2 enquanto um critério de parada não for atingido.

De forma geral, qualquer AE possui, basicamente, esta seqüência de passos. A diferença entre os métodos reside na forma como as melhores soluções são selecionadas, na forma como as soluções são representadas, ou na forma como uma nova população é gerada (BANKS; VINCENT; ANYAKOHA, 2007). São três os principais tipos de AE encontrados na literatura: as estratégias evolucionárias, a programação genética e os algoritmos genéticos.

2.1.1.2 Inteligência Coletiva

Analisando e extraindo características do comportamento de grupos como bandos de pássaros, colônias de formigas, cardumes de peixes, enxames de abelhas e também a interação social dos humanos, nota-se que estes utilizam a capacidade de cooperação para a resolução de

seus problemas, que na maioria das vezes consiste na busca por alimento e defesa (EBERHART; SHI, 2001).

O comportamento coletivo de populações fornece meios para a resolução de problemas, ou seja, o fato de um bando de pássaros voar sem se tocarem, a forma como um cardume de peixes se movimenta na tentativa de sobreviver a um ataque de predadores ou o fato de formigas utilizarem trilhas de feromônios para indicar a presença e direção da fonte de alimento, mostram que estas sociedades de organismos possuem meios de interação extremamente eficazes e complexos. A utilização desta inteligência coletiva permite o desenvolvimento de métodos de otimização que procuram imitar as ações dos indivíduos de uma determinada população (EBERHART; SHI, 2001).

Dois são os principais métodos de otimização que utilizam o conceito de inteligência coletiva. O primeiro é a otimização por colônia de formigas (ACO – *Ant Colony Optimization*), na qual há uma comunicação indireta através de feromônios para indicar caminhos mais curtos para a fonte de alimento. O segundo é a otimização por enxames de partículas (PSO – *Particle Swarm Optimization*), que utiliza a comunicação entre os indivíduos para o repasse de boas experiências, permitindo que os indivíduos mudem sua rota de busca com base em seu conhecimento e do conhecimento adquirido pela população como um todo (EBERHART; SHI, 2001) (ABRAHAM; GUO; LIU, 2006).

2.2 PARTICLE SWARM OPTIMIZATION

2.2.1 Breve Histórico

O PSO é uma técnica de otimização desenvolvida por Kennedy e Eberhart (1995). O conceito básico por trás do PSO é a simulação de um sistema social simplificado, baseada no comportamento de cardumes de peixes e bandos de pássaros, onde existe a cooperação entre os indivíduos do grupo, buscando uma posição que lhes proporcione manter uma distância ótima entre os indivíduos (KENNEDY; EBERHART, 1995).

A intenção inicial do PSO era apenas de tentar reproduzir computacionalmente o movimento de bandos de pássaros. Para isso várias melhoras do algoritmo foram feitas, como a inserção de cálculo de vizinhança e aceleração por distância. Após vários testes, notou-se que o método de simulação poderia ser empregado como uma ferramenta de otimização de funções. Isto foi percebido através da constatação de que geralmente um bando de pássaros pousa em um lugar onde exista a presença de comida (EBERHART; SHI; KENNEDY, 2001).

Nas primeiras versões do PSO, o método era capaz de otimizar somente funções algébricas bidimensionais, pois somente era necessário saber as coordenadas (x, y) do ponto onde havia comida. Porém a maioria dos problemas encontrados hoje possui muito mais do que duas

dimensões, sendo necessário modificar o método para que este pudesse tratar de problemas com dimensionalidades mais elevadas.

Esta técnica possui fortes semelhanças com outras técnicas de computação evolucionária, como algoritmos genéticos e programação genética. Assim como em um AG, o PSO tem sua população inicializada de maneira aleatória e utiliza o conceito de modificações em cada um de seus indivíduos para que estes evoluam com o passar do tempo. Entretanto, uma grande diferença entre o PSO e os AE é o fato de que o primeiro baseia-se na cooperação social entre seus indivíduos, enquanto o segundo utiliza a competição entre seus indivíduos (BANKS; VINCENT; ANYAKOHA, 2007).

O PSO modifica seus indivíduos a partir do cálculo de uma velocidade, que é baseada em informações do próprio indivíduo e informações da população em geral (EBERHART; SHI, 2001).

Suas grandes vantagens são: facilidade de implementação, e pouca necessidade de memória. Um dos pontos interessantes deste método, bem como de outros AE, é a possibilidade de explorar vários pontos do espaço de busca ao mesmo tempo, aumentando assim a possibilidade da solução ótima ser encontrada.

2.2.2 O Método

No método computacional os pássaros foram substituídos por agentes ou partículas, as quais quando observadas em conjunto formam a população necessária para a realização da busca. Cada partícula representa uma solução candidata e, com o decorrer do tempo cada partícula modifica as componentes de sua solução, através de informações pessoais e do grupo.

A população de partículas é inicializada de forma aleatória no espaço de busca e também é atribuída uma velocidade a cada partícula, para que estas possam modificar as componentes de sua solução de modo a se locomover pelo espaço de busca, procurando encontrar novas posições que sejam melhores. O termo “melhor” aqui é definido como uma posição no espaço de busca que possua um resultado para a função objetivo que se aproxime do ótimo. Como nos AGs, a qualidade de cada partícula é definida pela função de *fitness*.

Cada partícula armazena a melhor solução encontrada por ela durante o processo de busca, a qual é chamada de (*pbest - personal best*). Outro dado conhecido por uma partícula é a melhor solução encontrada pela população (*gbest - global best*) ou grupo de vizinhos (*lbest - local best*).

A cada passo de tempo a velocidade é modificada de forma a fazer com que cada partícula consiga ir à direção da melhora de sua solução. Esta modificação é feita através de uma diferença ponderada entre a solução atual e o *pbest* e também a diferença entre a solução atual e o *gbest*. Assim as partículas procuram determinar uma velocidade que as conduza para uma

posição mais próxima à melhor solução encontrada pela população sem desconsiderar a sua melhor solução. Na primeira versão do algoritmo (KENNEDY; EBERHART, 1995), a velocidade era calculada de acordo com a equação (1).

$$Vel_i^{t+1} = c_1 r_1 (pbest_i^t - q_i^t) + c_2 r_2 (gbest_i^t - q_i^t) \quad (1)$$

A variável q representa a solução candidata atual, onde o índice i se refere a cada uma das dimensões do problema, e o índice t denota as iterações. r_1 e r_2 são valores aleatórios com distribuição normal entre 0 e 1. As variáveis c_1 e c_2 , são duas constantes de aceleração, as quais possuem grande influência sobre o comportamento das partículas durante o processo de busca. Estas duas constantes permitem que uma partícula rume para o ponto $pbest$, quando c_1 é maior que c_2 , ou tenham a busca mais influenciada pelo ponto $gbest$ (EBERHART; SHI, 2001).

Na equação 1, o termo $c_1 r_1 (pbest_i^t - q_i^t)$, é chamado de componente cognitivo, por possibilitar que uma partícula seja atraída para o melhor ponto encontrado por ela. Esta é uma característica dos seres vivos, pois tendem a retornar a locais no ambiente onde tiveram melhor desempenho (BERGH; ENGELBRECHT, 2006).

Já o segundo termo $c_2 r_2 (gbest_i^t - q_i^t)$, na mesma equação, reflete o comportamento coletivo da partícula e é chamado de componente social. É ele que mostra o quão distante a partícula está do melhor ponto encontrado por toda a população. Através desta componente, uma partícula pode se utilizar do sucesso de outras partículas para conseguir encontrar um ponto melhor no espaço de busca (BERGH; ENGELBRECHT, 2006).

Com valores muito altos para c_1 , a população de partículas tende a ser dividida em pequenos grupos isolados, valores altos para c_2 podem ocasionar convergência prematura, indicando a estagnação da população em ótimos locais.

Como os passos de busca são determinados pelo cálculo das velocidades, existe a possibilidade de que o valor da velocidade faça com que o novo ponto fique fora do espaço de busca. Assim, é feito o controle da aceleração das partículas através de um parâmetro determinado pelo usuário, chamado V_{max} . Após o cálculo da nova velocidade, um teste verifica se este valor se encontra dentro do intervalo $[-V_{max}, V_{max}]$. Caso seja falso, o valor da velocidade assume um dos limites do intervalo (EBERHART; SHI, 1998) (BANKS; VINCENT; ANYAKOHA, 2007).

A escolha do valor de V_{max} influi fortemente no desempenho do método. É através deste parâmetro que é feito um balanço entre a exploração do espaço de busca, utilizando-se valores altos para V_{max} , e a *exploitation* ou busca local, optando-se por um V_{max} pequeno. Sendo assim, é necessário fazer ajustes finos para V_{max} . É sabido que a escolha do valor ótimo de V_{max} depende do problema (POLI; KENNEDY; BLACKWELL, 2007).

Como qualquer outro método, a escolha de valores ideais para os parâmetros é uma

tarefa difícil e geralmente depende do problema onde o método está sendo aplicado. Assim, seria interessante se o próprio método no decorrer do processo de busca conseguisse ajustar seus parâmetros de modo a obter boas soluções.

Eberhart e Shi (2001) propuseram a inclusão de um novo termo w que é chamado de momento de inércia. O intuito da introdução deste novo termo era evitar restringir velocidades. Desde então, a maior parte das implementações do PSO utilizam a modificação da velocidade de acordo com a equação (2).

$$Vel_i^{t+1} = w * Vel_i^t + c_1 r_1 * (pbest_i^t - q_i^t) + c_2 r_2 * (gbest_i^t - q_i^t) \quad (2)$$

De posse da velocidade no próximo passo de tempo, é possível então atualizar as coordenadas da solução atual, gerando uma nova solução, de acordo com a equação (3).

$$q_i^{t+1} = q_i^t + Vel_i^{t+1} \quad (3)$$

O algoritmo PSO pode ser definido pelos seguintes passos:

- a) Inicializar a população de partículas com valores aleatórios para cada componente da solução a ser encontrada;
- b) Avaliar a qualidade da solução pertencente a cada partícula. Caso for a primeira iteração do método, o valor de $pbest$ é o valor do $fitness$ da partícula. Caso contrário, se o valor de $fitness$ da solução atual for melhor do que $pbest$, atualizar $pbest$;
- c) Encontrar a melhor partícula da população e atualizar $gbest$. Na primeira iteração, o $fitness$ da primeira partícula pode ser utilizado para inicializar o $gbest$;
- d) Calcular as novas velocidades de acordo com a equação (2);
- e) Obter a nova solução de acordo com a equação (3);
- f) Repetir os passos a partir do item b) até que um determinado critério de parada seja atingido, podendo este ser o número máximo de iterações ou um valor de $fitness$ satisfatório.

Analisando o algoritmo descrito, pode-se verificar que o PSO é de fácil implementação, e também necessita de poucos recursos computacionais para sua execução, não contabilizando aqui os recursos para a avaliação de uma solução que, dependendo do problema, pode requerer mais poder de processamento.

2.2.3 Outras Abordagens

Desde a proposição do PSO, várias modificações e estudos foram feitos procurando melhorar seu desempenho. A seguir são mostradas algumas destas proposições.

Uma abordagem adicional que pode melhorar o desempenho do método é a utilização de formas de diversificação do processo de busca. Quando é verificado que não existe evolução das soluções fornecidas pelas partículas em um determinado número de iterações, é aconselhável que a população atual seja extinta e uma nova população seja criada, possibilitando assim a visita a novos pontos do espaço de busca antes não visitados. Neste caso, as informações sobre o melhor ponto do espaço de busca visitado até o momento da extinção devem ser armazenadas. Em Hemberger, Lopes e Junior. (2007) e Xie, Zhang e Yang (2002) é discutida a eficácia desta solução.

Outra possibilidade de implementação do PSO é adotar diferentes topologias de comunicação entre as partículas. No método clássico, a topologia é em estrela, onde existe comunicação entre todas as partículas da população. Entre as topologias sugeridas na literatura estão a circular, a grade, e a aleatória. Verificou-se que a topologia utilizada tem grande influência no comportamento do método, dependendo da função objetivo utilizada, mas não se pode concluir qual a melhor topologia (POLI; KENNEDY; BLACKWELL, 2007).

Clerc e Kennedy (2002), modificaram o cálculo das velocidades de cada partícula, adicionando um termo de restrição, buscando controlar a convergência das partículas de forma elegante, garantindo que não houvesse explosões para fora do espaço de busca e também não fosse necessário determinar um valor máximo para as velocidades.

2.2.4 Aplicações do PSO

Em Poli (2008), é feito um relato sobre as principais linhas de pesquisa e áreas onde o método PSO foi aplicado. Cerca de 650 artigos foram analisados, mostrando que o método está bastante difundido tanto na área acadêmica, quanto na área industrial. Muitas equipes utilizam o PSO pelo seu alto desempenho quando comparado com outros métodos de otimização global, bem como a sua fácil implementação. Entre as áreas de aplicação do método estão projetos de antenas, engenharia biomédica, redes de comunicação, classificação e *clustering*, otimização combinatorial, controle de motores e máquinas, finanças, entre outros. Na área de processamento de imagens e vídeos também foram encontrados trabalhos que abrangem reconhecimento de íris, fusão de imagens, detecção de objetos, entre outros.

O PSO, por ser flexível e ágil, pode ser utilizado para resolver problemas de otimização global ou qualquer outro problema que possa ser convertido em um problema de otimização.

2.3 CONCEITOS DE PROCESSAMENTO DE IMAGENS

Os métodos de reconhecimento de objetos utilizam as imagens digitais como fonte principal de informação. Uma imagem é uma função bidimensional de intensidade de luz $f(x,y)$, onde x e y representam coordenadas de um ponto no espaço cartesiano. Estas intensidades também são referenciadas como níveis de cinza ou brilho da imagem, no caso monocromático.

Para que uma imagem possa ser processada computacionalmente é necessário que tanto as coordenadas dos pontos quanto os níveis de cinza sejam discretizados. Após o processo de discretização se obtêm uma imagem digital, que pode ser melhor entendida como uma matriz onde suas células armazenam as informações sobre a intensidade de luz, ou nível de cinza, da imagem naquele ponto. Tais células são chamadas de elementos da imagem, elementos da figura, ou mais comumente de *pixels* (do inglês, *picture elements*). As coordenadas de um *pixel* são determinadas pelos índices das linhas e colunas da matriz (GONZALEZ; WOODS, 2000).

Neste trabalho serão utilizadas tanto imagens em níveis de cinza e quanto imagens coloridas, as quais são representadas pelo modelo RGB, que será abordado a seguir.

2.3.1 Modelo RGB

Para que seja possível utilizar cores em imagens digitais, é necessário fazer a escolha de um modelo de especificação de cores. Este modelo define um padrão de como as cores são representadas. Os modelos mais comuns utilizados no processamento de imagens são o RGB (*red, green, blue*), o HSI (matiz, saturação, intensidade) e o HSV (matiz, saturação, valor).

Neste trabalho optou-se por utilizar o modelo RGB, por ser amplamente difundido e atender às necessidades de representação de imagens coloridas. Este modelo baseia-se em um cubo, mostrado na figura 2. Então uma cor RGB, deve ser um ponto situado naquele cubo. Uma imagem neste modelo é composta por três camadas independentes, uma para cada componente da cor, também chamadas de canais. Um *pixel* no modelo RGB, é uma estrutura de dados que armazena três valores distintos, um para cada componente cor. A quantidade de cores possíveis para uma determinada imagem depende da quantidade de *bits* definida em cada *pixel*, que geralmente é de 8 bits para cada canal de cor.

2.3.2 Transformações Geométricas

Durante o processamento, uma imagem pode sofrer várias operações, entre elas as transformações geométricas, que permitem a realização de translações, rotações e mudança de escala.

Para efetuar a translação de um ponto com as coordenadas (x,y) , adicionam-se quantidades inteiras às suas coordenadas. Assim, cada ponto (x,y) pode ser movido por x_d unidades

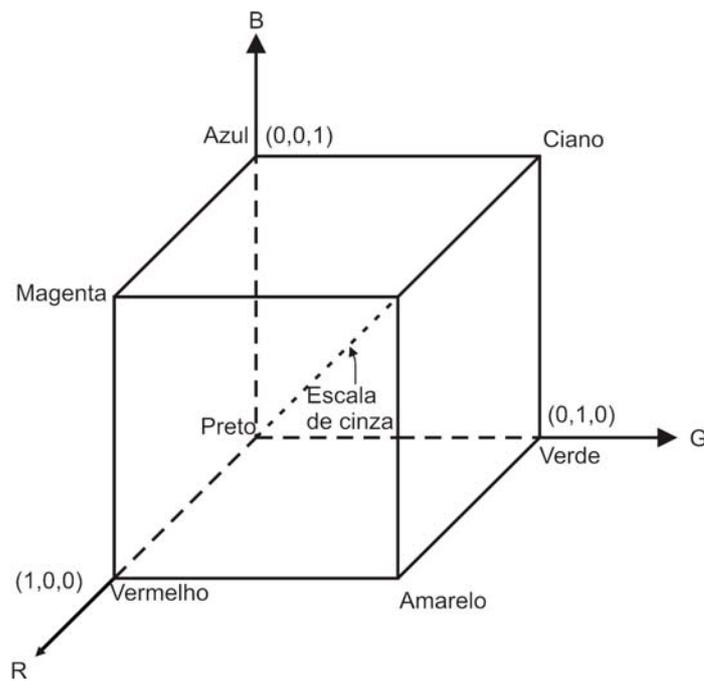


Figura 2: Cubo RGB.

em relação ao eixo x , e por y_d unidades em relação ao eixo y . Logo, o ponto (x^*, y^*) , pode ser escrito de acordo com as equações 4 e 5. A figura 3 mostra um exemplo de translação, onde o objeto foi transladado 100 pixels para a direita e 50 pixels para cima.

$$x^* = x + x_d \quad (4)$$

$$y^* = y + y_d \quad (5)$$

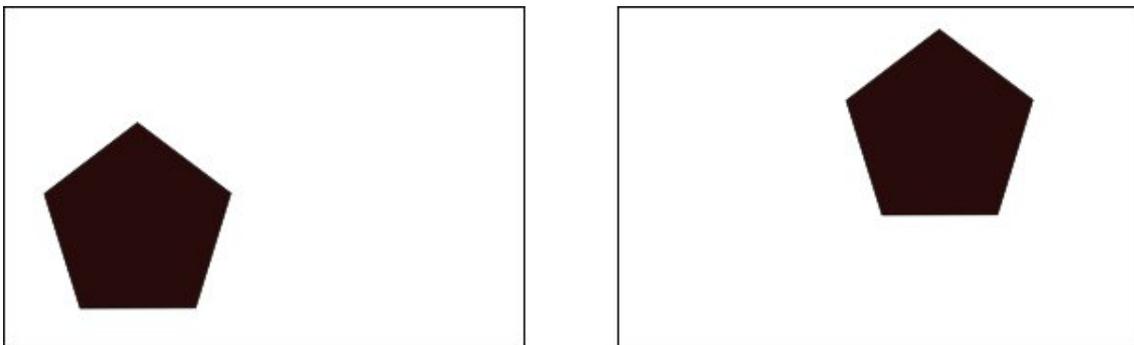


Figura 3: Exemplo de uma imagem transladada.

Podem-se efetuar mudanças de escala (ou redimensionamento) de um ponto no eixo x ou no eixo y através de multiplicações pelos fatores de escala S_x ou S_y , respectivamente, como mostrado nas equações 6 e 7. Para que as proporções das imagens não sejam alteradas, é necessário que $S_x = S_y$. A figura 4 mostra a aplicação de um fator de escala 1,3 sobre o objeto

contido na imagem.

$$x^* = S_x * x \quad (6)$$

$$y^* = S_y * y \quad (7)$$

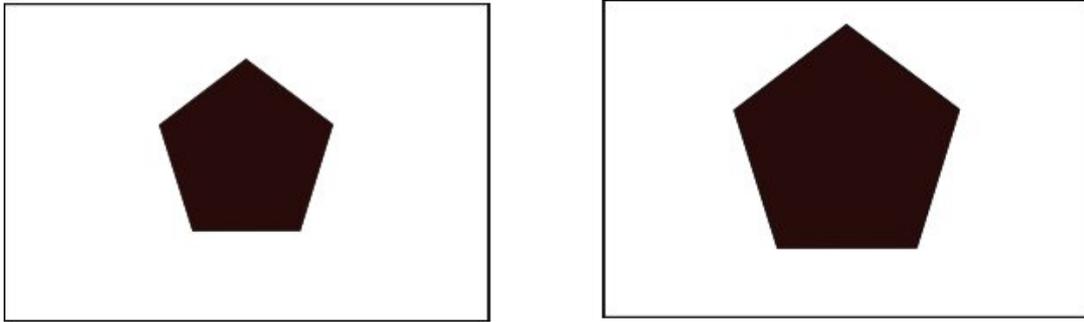


Figura 4: Exemplo de uma imagem com escala.

A realização da rotação de um ponto (x, y) no plano xy com base em um ângulo θ , é obtida através das equações 8 e 9. A figura 5 exemplifica a aplicação da operação de rotação com ângulo de 45° sobre o objeto contido na imagem.

$$x^* = x * \cos(\theta) - y * \sin(\theta) \quad (8)$$

$$y^* = y * \cos(\theta) + x * \sin(\theta) \quad (9)$$

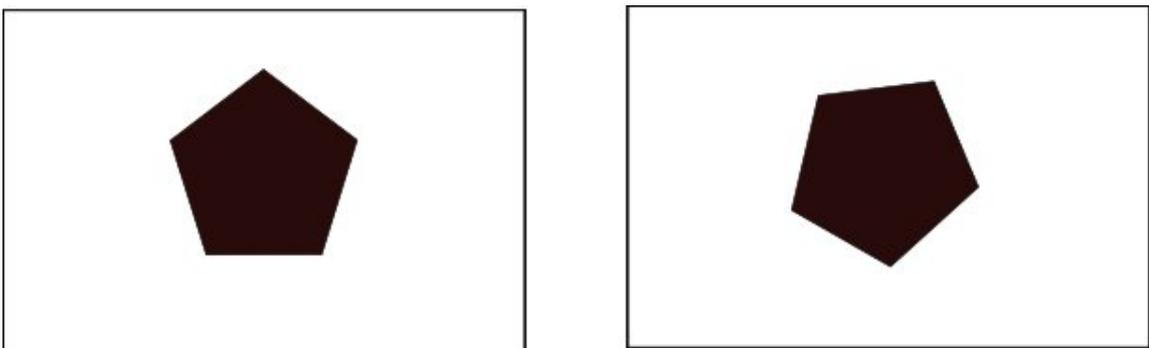


Figura 5: Exemplo de uma imagem com rotação.

2.3.3 Histograma

O histograma de uma imagem representa a freqüência de níveis de cinza ou cores naquela imagem. Dada uma imagem com níveis de cinza variando de 0 até $N - 1$, um histograma pode ser visto como uma função discreta $p(C_k) = \frac{n_k}{T}$, onde C_k é um dos níveis de cinza, n_k é a

quantidade de pixels com aquele nível de cinza e T é o total de pixels da imagem (ACHARYA; RAY, 2005). A figura 6(b) mostra o histograma obtido através da contagem dos pixels da imagem 6(a).

Com base nas informações providas pelo histograma é possível obter uma descrição global da imagem. A partir da análise e modificação do histograma pode-se melhorar a qualidade de uma imagem. No caso deste trabalho, o histograma é empregado para o cálculo de estatísticas necessárias para a obtenção da informação mútua entre as imagens.

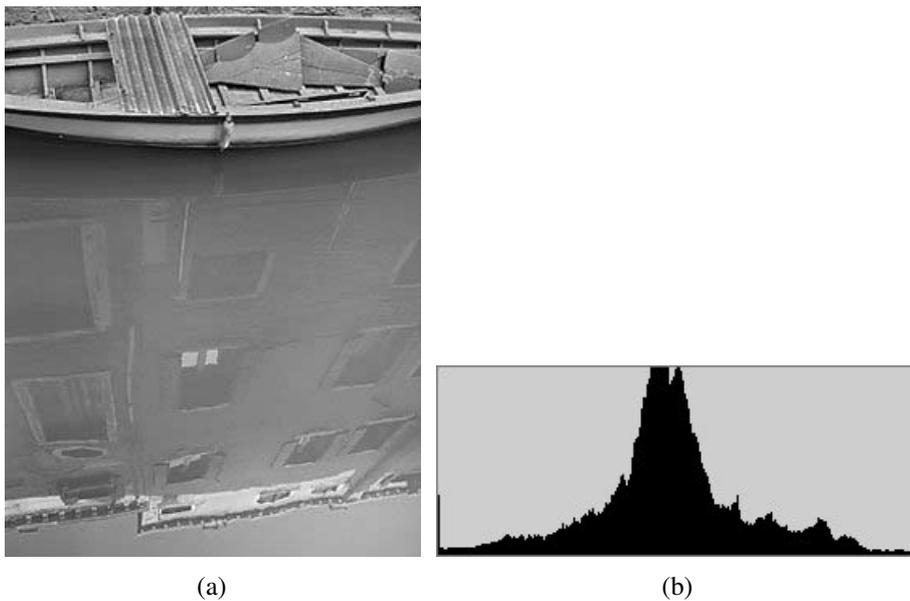


Figura 6: Exemplo de uma imagem (a) e seu respectivo histograma. Retirado de (MCHUGH, 2009).

2.3.4 Ruído

Devido às imperfeições nos sistemas de aquisição e armazenamento/transmissão de imagens digitais, vários tipos de problemas podem degradar a qualidade das imagens. Entre os principais problemas pode-se citar variações aleatórias de cores, variações de brilho, perda de foco e interferências eletromagnéticas. Tal degradação é conhecida como ruído, que pode ser visto como uma componente indesejada de uma imagem.

Dois tipos de ruídos comumente encontrados em imagens são o Gaussiano e o Impulsivo. Um tipo de ruído impulsivo é o Sal-e-Pimenta (S&P) (GARNETT; HUEGERICH; CHUI; WENJIE, 2005). Existem técnicas de processamento de imagens que podem amenizar ou retirar boa parte do ruído de imagens, mas isto implica em escolher o filtro adequado para minimizar o ruído bem como demanda tempo para realizar a devida correção.

O ruído Gaussiano representa uma degradação da imagem seguindo uma distribuição normal Gaussiana $G(\bar{x}, \sigma)$, definida pela média \bar{x} e pelo desvio padrão σ . O ruído Gaussiano

representa degradações em imagens geradas por agentes naturais como, interferência na conversão de sinal luminoso em sinal elétrico ou ruído elétrico durante a aquisição da imagem (BONCELET, 2005).

O ruído impulsivo S&P ocorre geralmente por mau funcionamento de *hardware*, problemas na transmissão digital das imagens ou mesmo no armazenamento. A imagem degradada por este tipo de ruído causa a impressão de ter sido salpicada por pontos brancos e pretos que afetam apenas uma parte do conteúdo da imagem (BONCELET, 2005).

A figura 7(a), mostra uma imagem aparentemente sem a presença de ruído Gaussiano ou S&P. As figuras 7(c) e 7(b), mostram a degradação da imagem por ruído Gaussiano e por ruído S&P, respectivamente.

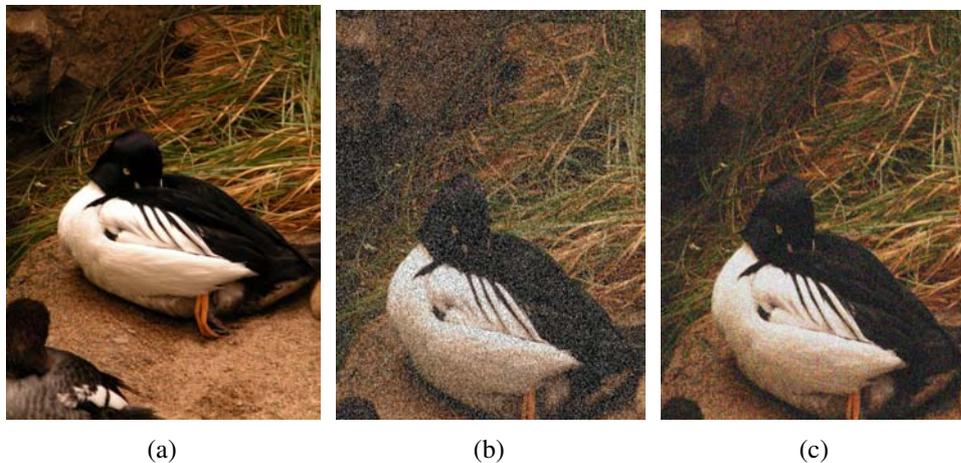


Figura 7: Exemplos de adição de ruído impulsivo S&P (b) e ruído gaussiano (c) em uma imagem (a).

2.4 SISTEMA DE VISÃO COMPUTACIONAL

Grande parte da informação que chega aos humanos vem por meio da visão. Tais informações são formas, cores e texturas, que permitem a tomada de decisões com base em algum conhecimento prévio. A percepção visual é a relação entre um estímulo visual recebido e um modelo previamente conhecido. Com base nas imagens que um humano enxerga é feita uma relação com os modelos conhecidos e memorizados e então é possível a extração de informação útil (FORSYTH; PONCE, 2002).

Com a difusão de equipamentos para captura de imagens, uma quantidade considerável de informação visual é gerada. Diante de um grande banco de imagens pode ser necessário saber quais das imagens contêm um determinado objeto de interesse. A automação de processos como busca de objetos, ou mesmo a leitura automática de endereços de correspondência, requer o emprego de técnicas de processamento de imagens capazes de realizar as tarefas necessárias,

fornecendo resultados satisfatórios em tempo hábil.

Um sistema de visão computacional (SVC) é um conjunto de técnicas de processamento de imagens e métodos de inteligência artificial agregados a fim de permitir a automatização de tarefas que utilizem informação visual como meio para tomada de decisões. Assim como para os humanos nos quais o sentido da visão fornece meios para que um indivíduo consiga se localizar no mundo onde vive, um sistema de visão computacional tem o mesmo objetivo com a diferença que o meio onde as informações são processadas é um computador (VERNON, 1991). A visão computacional é um mecanismo indispensável em muitas linhas de produção. Meios de automatização de tarefas ou controle de qualidade são formas de aumentar a produção bem como manter um alto nível de qualidade nos produtos finais (ZUECH, 2000).

A figura 8 mostra um exemplo de arquitetura de um sistema de visão computacional básico. Cada um dos componentes é responsável por uma parte do processo de extração da informação visual para permitir a tomada de decisão.

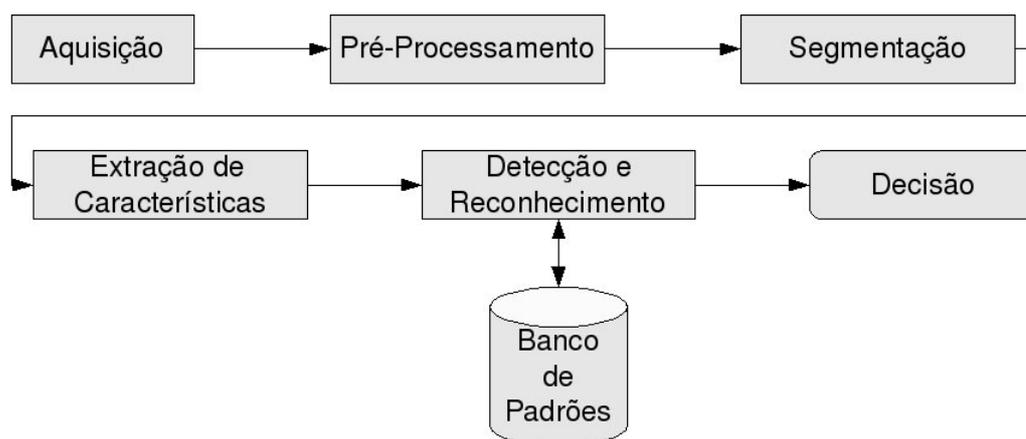


Figura 8: Exemplo de arquitetura de um sistema de visão computacional.

O primeiro passo é a aquisição da imagem, ou seja, é necessário o emprego de algum tipo de sensor capaz de receber informação luminosa e digitalizar o sinal a fim de fornecer uma imagem digital. O segundo passo consiste em aplicar técnicas de processamento de imagem objetivando melhorar a qualidade das imagens, aumentando assim a possibilidade de sucesso dos outros estágios. O próximo estágio é a segmentação onde a imagem de entrada é dividida em partes ou os objetos que a constituem são separados. Esta tarefa é uma das mais complexas, pois é com base na divisão fornecida pela segmentação que se dará o reconhecimento dos objetos. Assim, métodos robustos de segmentação favorecem a obtenção de bons resultados finais.

De posse da imagem segmentada, é realizada então a extração de características que permitam então a diferenciação e a identificação de cada um dos objetos que estão presentes na imagem. Tais características podem ser arestas, estruturas básicas, texturas, cores, entre outros. Com base nas características selecionadas dos objetos, é realizada uma consulta em uma base

de conhecimento, onde geralmente são armazenados padrões de características conhecidas de objetos. A similaridade entre as características obtidas e as armazenadas é empregada para definir se um objeto está presente na imagem ou realizar o correto reconhecimento do mesmo.

2.5 RECONHECIMENTO DE OBJETOS

A detecção e reconhecimento de objetos são passos fundamentais em visão computacional. A partir deles é que se consegue obter as informações relevantes de um conjunto de imagens. O que para os seres humanos é uma tarefa trivial, para um sistema computacional torna-se um processo complexo. Desta forma torna-se necessário o emprego de várias técnicas para que bons resultados sejam obtidos.

O termo detectar refere-se ao fato de a partir de uma imagem alvo encontrar todas ou a maioria das ocorrências de um objeto naquela imagem. O reconhecimento trata de separar o objeto localizado em classes. Em alguns casos, o fato de detectar a presença de um objeto em uma dada cena já permite que seja feita a classificação do mesmo. Sendo assim, dependendo do método empregado o termo detecção e reconhecimento podem ser utilizados indiscriminadamente (AMIT, 2002) (WANG; SHI; SONG; SHEN, 2007).

Na figura 9 é mostrado um exemplo onde foi empregada a detecção e o reconhecimento em placas de veículos. Para a obtenção dos dígitos que compõem a licença é necessário primeiramente localizar regiões da imagem que contenham dígitos e possuam características semelhantes a um retângulo. Em seguida localiza-se e extraem-se cada um dos dígitos. O resultado desta fase então é repassado a um método de reconhecimento que classifica o dígito, permitindo então a correta compreensão da licença do veículo.

Diferentes aplicações fazem uso de reconhecimento de objetos em uma imagem. Dentre elas pode-se citar a navegação autônoma de robôs, onde estes conseguem se locomover de forma eficiente pelo ambiente através da localização e reconhecimento de objetos, marcações, entre outros. Outras aplicações são: assistência à motoristas (LAIKA; STECHELE, 2007), reconhecimento de placas de veículos (JIAO; YE; HUANG, 2009), detecção de faces (JANG; HAN; KIM, 2004), detecção de pessoas em ambientes (KROTOSKY; TRIVEDI, 2007) e reconhecimento de falhas em processos industriais (FELISBERTO; SILVÉRIO; CENTENO; ARRUDA, 2006).

2.5.1 Estratégias de Reconhecimento

Neste trabalho, o termo imagem de paisagem (IP) é utilizado para referenciar a imagem onde se supõem que o objeto esteja presente. O termo imagem de referência (IR) é utilizado para definir a imagem que contém o objeto a ser reconhecido, ou seja, é o modelo padrão do



Figura 9: Exemplo de detecção e reconhecimento de placa de veículos. Retirado de (ZIMMERMANN, 2009).

objeto.

Para o reconhecimento de objetos em imagens, geralmente é necessário a junção de várias técnicas de processamento de imagens com técnicas de inteligência artificial. As técnicas de processamento de imagens são empregadas para a obtenção de uma representação significativa do objeto de interesse, bem como para extrair as informações contidas na IP, a fim de permitir a correta identificação do objeto (GONZALEZ; WOODS, 2000) (ZUECH, 2000).

A partir das características é necessário que seja empregada alguma forma de classificação ou comparação para que então seja efetivamente realizado o reconhecimento. A tarefa de busca e reconhecimento do objeto de interesse na IP é uma tarefa complexa e que pode demandar uma grande quantidade de tempo e poder computacional. Assim, técnicas de inteligência artificial são utilizadas para a redução da necessidade de recursos e também para aumentar a qualidade das respostas. Dentre as formas comumente empregadas na literatura estão o *template matching*(TM), métodos estatísticos, métodos sintáticos ou estruturais e redes neurais (ZUECH, 2000).

Métodos estatísticos de reconhecimento utilizam funções de decisão para classificar um determinado objeto em uma classe. O emprego de probabilidades para a classificação é importante pois a forma como um objeto pode aparecer em uma imagem, na maioria dos casos, é aleatória (GONZALEZ; WOODS, 2000).

Métodos estruturais se baseiam na descrição do objeto através do uso de primitivas juntamente com a relação espacial entre elas. Geralmente tais métodos utilizam o casamento de cadeias para a realização do reconhecimento (GONZALEZ; WOODS, 2000).

As redes neurais utilizam elementos de computação não-linear, os neurônios, interligados por meio de conexões formando uma rede. Para uma rede neural realizar a classificação é necessário uma fase de treinamento prévio. Apresentando sucessivamente conjuntos de padrões

de treinamento a uma rede, suas conexões são ajustadas a fim de formular uma função de decisão. Depois de realizado o treinamento, padrões desconhecidos são apresentados à rede e esta responde com a classificação (GONZALEZ; WOODS, 2000).

A técnica de TM procura identificar uma região de uma imagem que seja mais parecida possível com um modelo ou *template*. O processo de procura é realizado sobrepondo-se o *template* à imagem em diferentes pontos. Em cada um desses pontos é calculada uma medida de similaridade, que diz o quanto são parecidos o modelo e aquela região da imagem (ZUECH, 2000).

2.5.2 Medidas de Similaridade

Para a realização do TM é necessário definir uma medida de similaridade entre a representação do objeto e parte da imagem alvo, onde se supõe que esteja o objeto. Esta necessidade de definição permite que o TM seja visualizado como um problema de otimização. Assim um método de otimização como o PSO pode então, ser utilizado para encontrar o maior ponto de similaridade entre o modelo e a imagem. Esta foi a abordagem empregada neste trabalho, como será explicitado na seção 3.1.

Diversas métricas podem ser utilizadas, entre elas pode-se citar a correlação cruzada, o erro quadrático e a distância euclidiana. Neste trabalho duas métricas foram estudadas: a informação mútua e a soma da diferença absoluta entre pixels.

2.5.2.1 Norma L1

O somatório do módulo das diferenças, também conhecida como norma L1, é utilizada para calcular a distância entre dois vetores (RUSSELL; SINHA, 2001). Interpretando uma imagem como um vetor, podemos utilizar esta métrica para comparar a semelhança entre duas imagens. Assim, pode-se escrever a norma L1 como na equação (10), onde n e m são a largura e altura da IR. IP_{rec} é um recorte da IP de mesma dimensão da IR. Os índices i e j são usados para percorrer as duas imagens pixel-a-pixel. A grande vantagem desta função é a simplicidade. O resultado da aplicação da norma L1 entre duas imagens é o grau de dissimilaridade entre as mesmas.

$$L1 = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} |IR(i, j) - IP_{rec}(i, j)| \quad (10)$$

2.5.2.2 Informação Mútua

A informação mútua (IM) é uma medida que pode ser interpretada como a quantidade de informação que uma variável aleatória informa sobre uma outra variável. Viola e Wells III

(1997) propuseram a utilização desta função como métrica para realizar o registro de imagens. Após a proposição vários trabalhos mostraram a viabilidade do seu uso como medida de similaridade entre imagens (XIANG; SHUQIAN; KUNCHENG, 1999) (KROTOSKY; TRIVEDI, 2007).

Visualizando uma imagem como uma seqüência de valores aleatórios, podemos então utilizar a IM para computar a similaridade entre duas imagens. Desta forma quanto maior for o valor da IM calculada entre duas imagens maior a similaridade entre elas. A equação 11 mostra uma das formas de se obter a IM entre IR e IP_{rec} , onde $H()$ representa a entropia. Três informações são necessárias para o cálculo da IM, a entropia da IR a entropia da IP e a entropia conjunta entre IR e IP_{rec} .

$$IM(IR, IP_{rec}) = H(IR) + H(IP_{rec}) - H(IR, IP_{rec}) \quad (11)$$

A entropia pode ser vista como uma medida de quantidade de informação, comumente utilizada na área da teoria da informação. Dado um canal de transmissão, deseja-se saber qual é a quantidade de informação que chega ao receptor quando este recebe uma determinada mensagem. Shannon (2001) definiu uma medida de quantidade de informação baseada nas probabilidades de ocorrência de símbolos ou palavras dentro de uma mensagem. Segundo a formulação de Shannon, quanto mais rara a ocorrência de um símbolo maior será a quantidade de informação gerada quando este símbolo for recebido (PLUIM; MAINTZ; VIERGEVER, 2003).

A entropia de Shannon também pode ser analisada como uma medida de incerteza. Assim, se em um conjunto de símbolos a probabilidade de ocorrência é a mesma, existe um alto nível de incerteza de qual símbolo será recebido, levando a entropia chegar ao seu valor máximo. Caso exista a probabilidade de ocorrência de um único símbolo, sabe-se *a priori* qual será o símbolo recebido, ou seja, a incerteza é mínima (PLUIM; MAINTZ; VIERGEVER, 2003). A equação 12 é a forma mais comum de definição da entropia de Shannon, onde P_i denota a probabilidade de ocorrência de um símbolo que compõe a mensagem.

$$H = \sum p_i \log \frac{1}{P_i} \quad (12)$$

A entropia conjunta de duas imagens pode ser obtida a partir da equação 13, onde $C_{i,j}$ denota a probabilidade conjunta de ocorrência de dois níveis de cinza. Tal probabilidade conjunta pode ser calculada de posse do histograma conjunto das duas imagens. O processo para a obtenção do histograma conjunto é explanado na seção 3.2.2 (pág. 31).

$$H = \sum C(i, j) \log \frac{1}{C(i, j)} \quad (13)$$

Analisando a equação 11, verifica-se que para maximizar a IM a entropia conjunta deve

ser diminuída, pois tanto a entropia da IR quanto a entropia da IP não podem ser alteradas. A entropia conjunta é reduzida em função da semelhança entre as imagens. Isso se deve ao fato de que a entropia conjunta é obtida com base no histograma conjunto das imagens. Assim se duas imagens forem bastante semelhantes o histograma conjunto apresentará pouca dispersão nas ocorrências, fato que reduzirá o valor da entropia conjunta.

2.6 TRABALHOS CORRELATOS

Em Owechko e Medasani (2005) o conceito de enxames cognitivos foi aplicado ao problema de localização de pessoas e veículos. Um enxame cognitivo é uma população de partículas que trocam informações entre si, de modo a cooperarem e conseguirem atingir um determinado objetivo, ou seja, uma forma diferente de se explicar os conceitos que regem o PSO. Foi desenvolvido um *framework* baseado em técnicas estatísticas de reconhecimento de padrões juntamente com o PSO, com o objetivo de localização rápida e acurada de objetos. O método mostrou-se bastante eficiente, gerando bons resultados com um ganho considerável de desempenho quando comparado com o método de busca exaustiva.

Saisan, Medasani e Owechko (2005) utilizam o eficiente mecanismo de busca fornecido pelo PSO juntamente com classificação de objetos 2D baseada em características, para localização e reconhecimento de objetos 3D. Duas abordagens foram implementadas. Na primeira, cada partícula do PSO é vista como uma janela bidimensional para a imagem, onde as partículas sobrevoam procurando por regiões que mais se pareçam com o objeto desejado. Utilizando um operador de projeção, foi possível realizar a detecção no espaço tridimensional. A segunda é a utilização de várias poses 2D do objeto que quando unidas, por meio de *stereo methods*, permitem a determinação das coordenadas de um objeto 3D. Foram obtidos bons resultados para a segunda abordagem, mas os autores esperam conseguir melhores resultados com base na primeira abordagem.

Em Perez e Vallejos (2006) foi empregado o método do PSO para selecionar *templates* usados na detecção de faces. Foram feitas algumas modificações no PSO para a obtenção de melhores resultados, como a modificação condicional da melhor partícula e também o uso de vizinhança para a escolha da melhor partícula. Os resultados mostraram que o PSO conseguiu selecionar *templates* com um número reduzido de pontos, permitindo assim uma economia no tempo de processamento quando comparado com técnicas antropométricas tradicionais.

Outra solução do problema de reconhecimento de objetos que emprega computação evolucionária foi proposta em (FELISBERTO; SILVÉRIO; CENTENO; ARRUDA, 2006). Uma implementação de algoritmo genético tradicional foi feita para a determinação dos parâmetros de definição de uma janela com uma trilha de solda, similar a um dado modelo. O método

conseguiu cerca de 94,4% de acerto, indicando que esta é uma boa alternativa para a extração de imagens de cordões de solda a partir de imagens radiográficas para a realização de inspeções.

Kobayashi, Nakagawa, Imae e Zhai (2007) desenvolveram um método para perseguição de objetos em tempo real utilizando o PSO e agregando a ele um algoritmo apropriado de atualização e uma técnica de reinicialização. O método possui algumas limitações como a restrição de haver apenas um objeto e não admitir mudanças bruscas na posição do objeto. Após um conjunto de testes, foi concluído que o método possui alto desempenho quando comparado com outros métodos.

Em Zhang et al. (2008) o método PSO é utilizado para a resolução do problema de perseguição visual, onde o movimento de um objeto em uma seqüência de imagens deve ser estimado. Como o processo de perseguição é dinâmico, foi necessário incluir a informação temporal ao PSO para torná-lo mais robusto na solução do problema. Além disso, baseado no valor da função de *fitness* os parâmetros do PSO são mudados de modo a adaptá-lo ao problema. Os resultados mostraram que o PSO é um bom método para ser utilizado na resolução do problema de perseguição visual.

Ba-Karait e Shamsuddin (2008) aplicaram o PSO na resolução do problema de reconhecimento de dígitos manuscritos. O PSO é responsável por determinar o melhor conjunto de centróides das classes. Os resultados mostraram que o método é promissor e estável.

Recentemente, Silva, Lopes e Lima (2008) propuseram o uso de um algoritmo genético compacto com elitismo e mutação (emCGA) para o reconhecimento de objetos em imagens. O trabalho desenvolvido utiliza, assim como é o caso deste, uma técnica de computação evolucionária para a resolução do problema de otimização de uma função para o reconhecimento de objetos. Os resultados obtidos neste trabalho foram bastante satisfatórios, mostrando que o emCGA é uma técnica promissora para o problema de reconhecimento.

No capítulo a seguir será explanada a metodologia proposta para a solução do problema de reconhecimento de objetos em imagens digitais.

CAPÍTULO 3

METODOLOGIA

3.1 RECONHECIMENTO DE OBJETOS COMO PROBLEMA DE OTIMIZAÇÃO

Dadas duas imagens, a IR, a qual contém um objeto de referência ou padrão, e a IP, onde se supõe que o objeto esteja, deseja-se determinar se a IR está contida na IP. Um meio para obter essa informação é aplicar a técnica de TM que consiste em comparar a IR com um recorte de IP (de mesmas dimensões da IR), e determinar o quanto são parecidas. Para aplicar esta técnica como solução para o reconhecimento de objetos é necessário comparar inúmeros recortes.

A idéia básica do TM é utilizar em seu processo de comparação uma janela deslizante sobre a imagem de paisagem. Cada um dos pixels da IP seria o centro da janela. O conteúdo coberto pela janela, aqui chamado de recorte, é utilizado para a comparação com a referência ou modelo. Sendo assim, esta técnica somente pode fornecer informações sobre a translação do objeto dentro da IP. A figura 10 mostra o funcionamento do TM.

Métodos tradicionais de reconhecimento de objetos necessitam de técnicas de segmentação de imagens para extrair os objetos de interesse. Uma das vantagens de se utilizar o TM como forma de detectar objetos em uma imagem, é que o método não necessita realizar a segmentação *a priori*, e nem conhecer nenhuma característica em especial do objeto desejado, como arestas ou formatos característicos. O método só necessita de um exemplo. Este fato torna esta solução um tanto quanto genérica, ou seja, é possível que funcione com a maioria das classes de objetos.

Uma desvantagem de se utilizar o TM somente como uma janela deslizante é que um objeto pode aparecer em várias posições dentro de uma imagem, bem como ter sofrido rotações e também transformações de escala. Diante dessas transformações o TM é incapaz de realizar a detecção ou reconhecimento com eficácia.

Sabendo que tais transformações podem ocorrer, localizar um objeto em uma IP consiste na determinação das coordenadas de translação, o ângulo de rotação e o fator de escala. Desta forma, pode-se dizer que o TM fica invariante à translação, à escala e à rotação.

Isto aumenta consideravelmente a quantidade de recortes da IP que devem ser comparados com a IR para que sejam realizados a detecção e o reconhecimento do objeto. Cada recorte passa a ser uma solução possível para o problema, tornando o tamanho do espaço de busca muito grande.

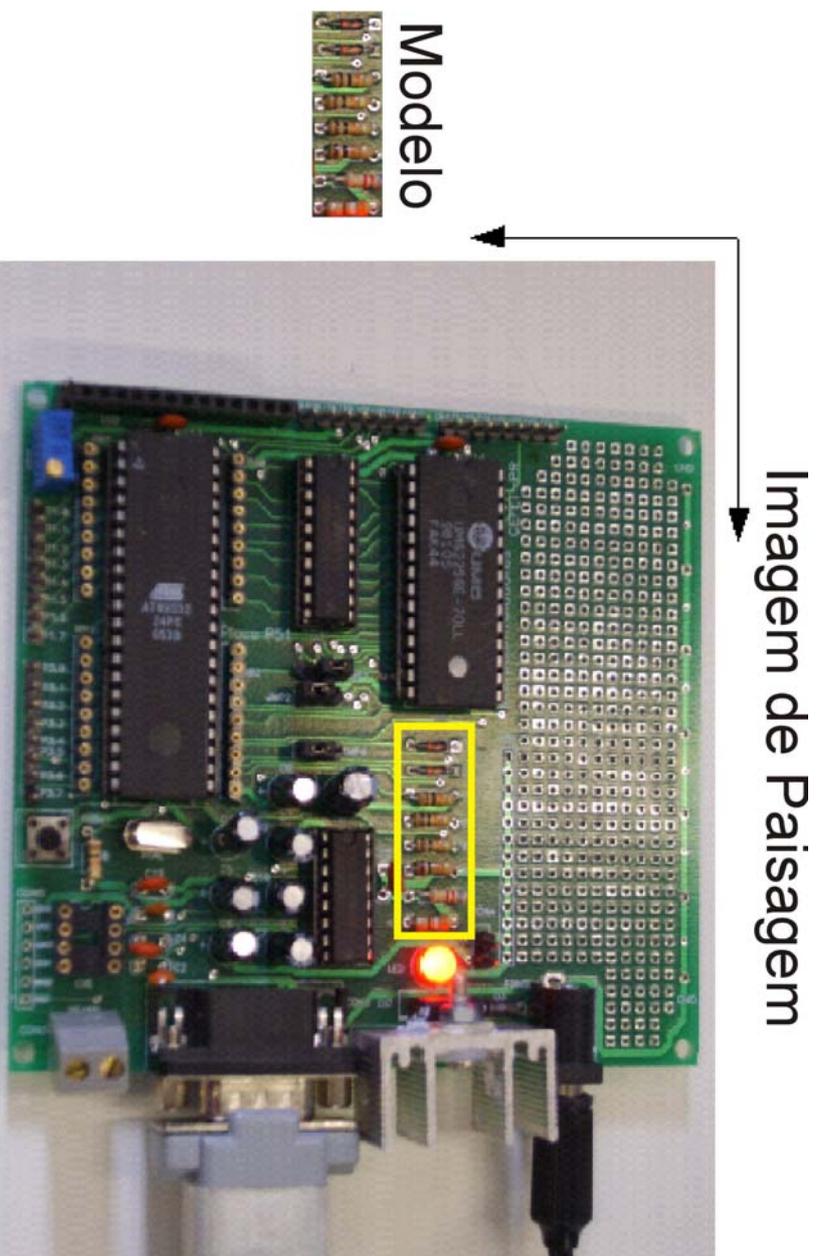


Figura 10: Exemplo de *template matching*. Uma imagem contendo o objeto (modelo) é sobreposta sobre a imagem de paisagem e é deslizada sobre todos os pixels da mesma. No ponto onde o valor da medida de similaridade for o melhor significa que ocorreu um casamento entre o objeto e a paisagem, o que implica no reconhecimento do objeto naquela imagem de paisagem, representado por um retângulo amarelo na imagem de paisagem.

Encontrar a melhor solução para este problema passa a ser inviável através de um algoritmo de força bruta, ou seja, testar todas as possíveis soluções demanda muito tempo de processamento. Como a medida de similaridade utilizada no TM é uma função matemática que fornece um valor para uma dada configuração dos parâmetros de uma solução, pode-se interpretar o TM como um problema de otimização. Sendo assim, deseja-se encontrar a combinação de valores para as transformações que possibilite a extração do recorte da IP que quando comparado com a IR através da medida de similaridade forneça o seu melhor valor ou um valor muito próximo.

Visualizando a técnica de TM como um problema de otimização, é possível utilizar um método de busca inteligente que necessite de poucos recursos tanto de tempo como de poder computacional, e forneça soluções de alta qualidade. Como o TM mostra a parte ou recorte da IP que mais se parece com o objeto de interesse, tem-se então uma proposta de solução para o problema de reconhecimento de objetos.

3.1.1 Descrição do *Template Matching*

Para realizar o TM, tornando-o invariante à translação, à rotação e à escala, é necessário definir a forma de representar uma solução. Uma solução é um vetor de 4 posições: (x, y, s, θ) , onde x e y são as coordenadas cartesianas do ponto central da imagem de referência relativas à IP, s é o fator de escala, ou *zoom*, e θ é o ângulo de rotação aplicado à IR localizada na imagem de paisagem.

Com base nos valores dos parâmetros desta solução, é feito um recorte com as dimensões da IR a partir da IP, com centro no ponto (x, y) . Sobre este recorte é realizada a transformação de escala, com base no fator s da solução. A escala tanto pode ser de ampliação, caso $s > 1$ ou redução, quando $s < 1$. Caso $s = 1$, a transformação de escala não modifica a imagem.

Após a realização da escala, é aplicada uma rotação sobre o recorte. O resultado desta rotação tenta colocar o recorte na mesma orientação da imagem do objeto de referência.

De posse do recorte com as transformações aplicadas, é feito então o cálculo de similaridade. O resultado desta função mostra o quanto o recorte, obtido a partir dos parâmetros da solução candidata, é parecido com a imagem do objeto de referência. Este valor define a qualidade de uma solução dentro de um conjunto de possíveis soluções candidatas.

A figura 11 descreve de forma sintetizada a seqüência de passos que é executada para a realização do TM. Quando seguida esta seqüência consegue-se tornar o TM invariante à translação, à escala e à rotação. As flechas na figura denotam dados de entrada necessários para a realização de cada um dos passos, denotados pelos retângulos. Estes passos são realizados para cada uma das soluções candidatas a serem testadas, por isso deve-se procurar realizá-los de forma eficiente.

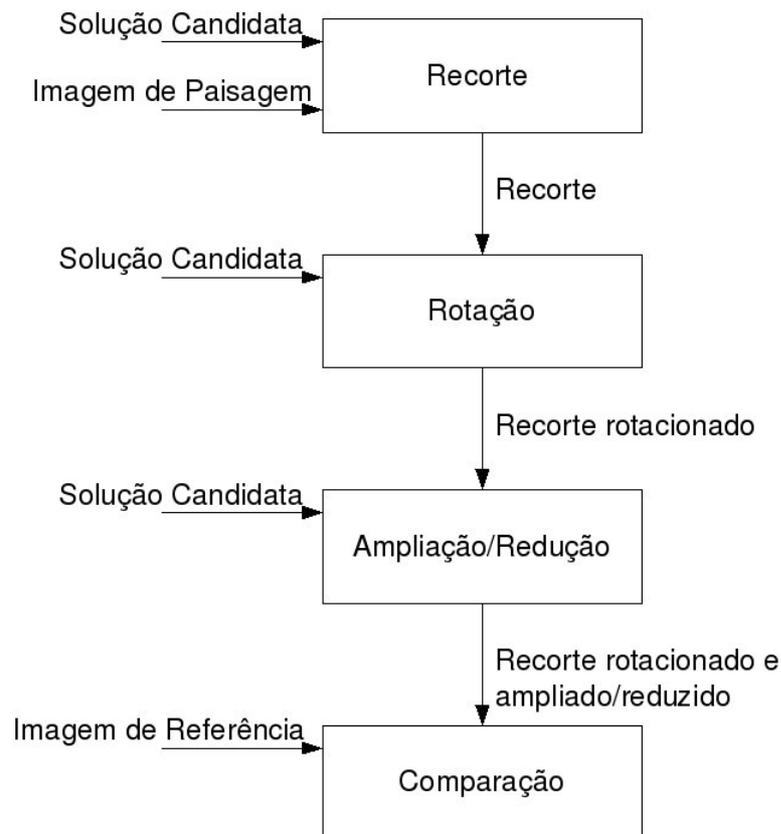


Figura 11: Passo e dados necessários para tornar o *template matching* invariante à translação, à rotação e à escala.

3.2 MEDIDAS DE SIMILARIDADE

A medida de similaridade é um fator essencial para o funcionamento correto do TM. É necessário que seja empregada uma função que consiga mostrar o quanto as imagens são iguais ou diferentes. A partir desta comparação será feito o processo de reconhecimento de um objeto.

Como já explicitado na seção 2.5.2 (pág.23), diversas funções matemáticas podem ser utilizadas como medidas de similaridade ou dissimilaridade. Duas medidas foram escolhidas para compor o estudo desenvolvido neste trabalho, a norma L1, ou soma do módulo das diferenças dos pixels e a informação mútua. A escolha da primeira deu-se pela sua simplicidade de implementação, pois somente necessita de dados que podem ser obtidos diretamente das imagens. Já a segunda medida é largamente empregada como medida de similaridade na construção de métodos de *image registration* (PLUIM; MAINTZ; VIERGEVER, 2003).

3.2.1 Norma L1

A norma L1 indica a diferença entre dois vetores quaisquer, ou seja, é uma medida de dissimilaridade. Assim quanto maior for o valor da norma, maior a diferença entre os vetores. No caso da aplicação no reconhecimento de objetos, se deseja encontrar duas imagens com um

alto grau de similaridade, então o valor da norma deve ser o menor possível.

Para tentar reduzir o tempo gasto com os cálculos da norma L1, foi adotado o uso da biblioteca de visão computacional OpenCV ¹. Duas funções desta biblioteca são utilizadas: a *cv_getquadranglesubpix* e a *cv_norm*. A primeira é utilizada para realizar um recorte da imagem de paisagem com base nas informações da solução candidata. É nesta função que são aplicadas as transformações geométricas sobre o recorte da imagem. De posse do recorte, é utilizada a segunda função que realiza o cálculo de L1 de forma bastante otimizada.

Como será explicado na seção 3.3, uma solução candidata pode assumir qualquer valor dentro de um determinado intervalo. Diante disso, é possível que as coordenadas de translação (x, y) fiquem próximas à borda da IP. Sendo assim, é necessário que a métrica possua algum tipo de tratamento para a realização dos cálculos próximos à borda. No caso da norma L1 a própria função *cv_getquadranglesubpix* trata disso, replicando os *pixels* que compõem a borda para o restante do recorte que ficou “fora” da IP.

3.2.2 Informação Mútua

A fim de explicar como é realizado o cálculo para obtenção da IM entre duas imagens, um exemplo será mostrado. Sejam as figuras 12(a) e 12(b) exemplos de duas imagens em níveis de cinza. Supondo que sejam utilizados 3 *bits* para representar cada nível de cinza, cada *pixel* pode assumir um valor entre 0 e 7.

Primeiramente calcula-se a entropia de cada uma das imagens. A entropia de cada uma das imagens pode ser obtida através do histograma. As tabelas 1 e 2 mostram o histograma das imagens (2^a coluna), bem como outras informações necessárias para a obtenção da entropia. É necessário atentar para o fato da ocorrência de um determinado nível de cinza for igual 0, o que implica que o valor de $p_i \log \frac{1}{p_i}$ ser 0 pois $\lim_{p \rightarrow 0} p \log p = 0$. Fazendo a soma dos valores de $p_i \log \frac{1}{p_i}$ de cada tabela, obtém-se que a entropia para a figura 12(a) é 0,86 e para a figura 12(b) a entropia é 0,77.

A figura 13 mostra o histograma conjunto gerado entre as ocorrências dos níveis de cinza entre a imagem das figuras 12(a) e 12(b). A primeira linha representa os níveis de cinza da primeira imagem, e a primeira coluna representa os níveis de cinza da segunda imagem. Com base nos valores de ocorrência, a entropia conjunta vale 1,30.

De posse dos valores das entropias de cada imagem bem como da entropia conjunta, podemos calcular a informação mútua conforme a equação 11 (pág.24). O valor da IM para este exemplo é 0,33. Este valor demonstra que as duas imagens possuem pouca similaridade. Uma

¹OpenCV (Open Source Computer Vision) é uma biblioteca de funções de programação para o desenvolvimento de aplicações na área de visão computacional. As funções da biblioteca são escritas na linguagem C++, mas existem versões para python, java e matlab. Disponível na internet em: <http://sourceforge.net/projects/opencvlibrary/>

	0	1	2	3	4	5	6	7
0	2	0	0	1	0	1	0	2
1	0	1	0	0	0	0	1	1
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	1	0	0
4	0	0	0	1	1	0	1	0
5	0	0	0	1	0	1	0	0
6	0	0	1	0	0	1	0	0
7	1	1	1	1	0	1	2	0

Figura 13: Histograma conjunto entre as imagens das figuras 12(a) e 12(b).

3.2.3 Análise de Superfície de *Fitness*

O método de otimização empregado para realizar a busca da melhor solução para o problema de reconhecimento de objeto deve explorar um espaço de soluções. Em um problema de duas dimensões, o espaço de busca pode ser visto como uma superfície, onde nos eixos x e y tem-se os valores possíveis para as variáveis, e no eixo z o valor da função objetivo no ponto (x,y) . A forma desta superfície, como a presença de rugosidades, ou a existência de vários pontos de máximo ou mínimo, podem indicar como o método de busca irá se comportar.

Como existem várias funções de similaridade que podem ser empregadas e no caso deste trabalho foram escolhidas duas, deve-se decidir qual delas é a mais apropriada. A escolha da medida foi baseada na visualização e análise da superfície gerada pela função de *fitness*, de forma que a decisão independe do método de otimização utilizado.

No caso do reconhecimento de objetos, para que seja possível realizar a visualização da superfície de busca, o problema que possui quatro dimensões, translação no eixo x , translação no eixo y , rotação e escala, foi reduzido para um problema com duas dimensões, translação no eixo x e translação no eixo y .

A partir desta simplificação, a construção da superfície seguiu os seguintes passos:

1. Foram escolhidas imagens de paisagem e imagens com os objetos desejados;
2. O valor da medida de similaridade foi calculado em cada pixel da imagem de paisagem, centrando-se a imagem de referência sobre o pixel atual;
3. O valor da medida de similaridade foi armazenado e posteriormente submetido a um software que interpola os pontos gerando uma superfície.

A norma L1 é uma medida de dissimilaridade, portanto quanto menor o seu valor mais parecidas serão as duas imagens, ou seja, pontos de baixa dissimilaridade aparecerão como vales na superfície. Já a IM é uma medida de similaridade fornecendo altos valores para imagens semelhantes, sendo assim pontos de alta similaridade aparecerão como picos em uma superfície. Procurando facilitar a análise, a superfície da norma L1 foi invertida, onde os pontos de baixa dissimilaridade são mostrados como picos na superfície. As superfícies do espaço de busca para a IR da figura 14(b) deslocada sobre a IM da figura 14(a), são mostradas para a norma L1 e a para IM nas figuras 15 e 16, respectivamente.

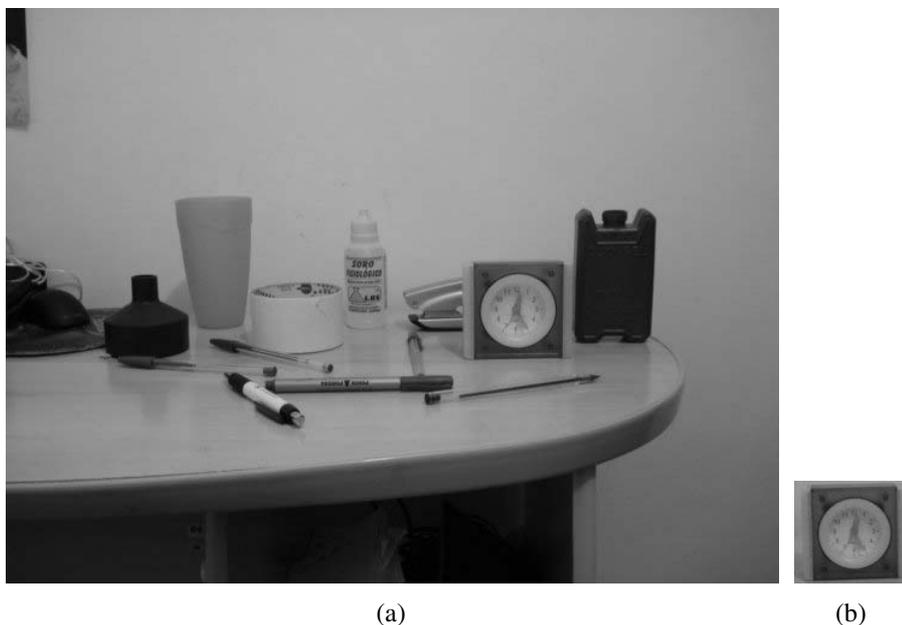


Figura 14: (a) Cena complexa utilizada como IP. (b) IR a ser reconhecida.

Observando-se as duas superfícies, pode-se verificar que qualquer uma das medidas consegue um bom nível de distinção entre o que é o objeto e o que não é o objeto desejado, pois as duas apresentam um pico bastante ressaltado em sua superfície, exatamente sobre o ponto aonde a IR aparece na IP. Porém, numa análise mais minuciosa, verifica-se que a superfície gerada pela L1 é mais suave que a superfície da IM. Uma superfície de busca suave pode ser explorada eficientemente por uma heurística como o PSO, devido à baixa presença mínimos/máximos locais.

Além disso, o tempo computacional necessário para o cálculo de cada uma das medidas de similaridade também foi levado em conta na decisão. Enquanto o cálculo de L1 para cada recorte leva frações de segundo para ser obtido, a IM necessita de alguns segundos para ser calculada.

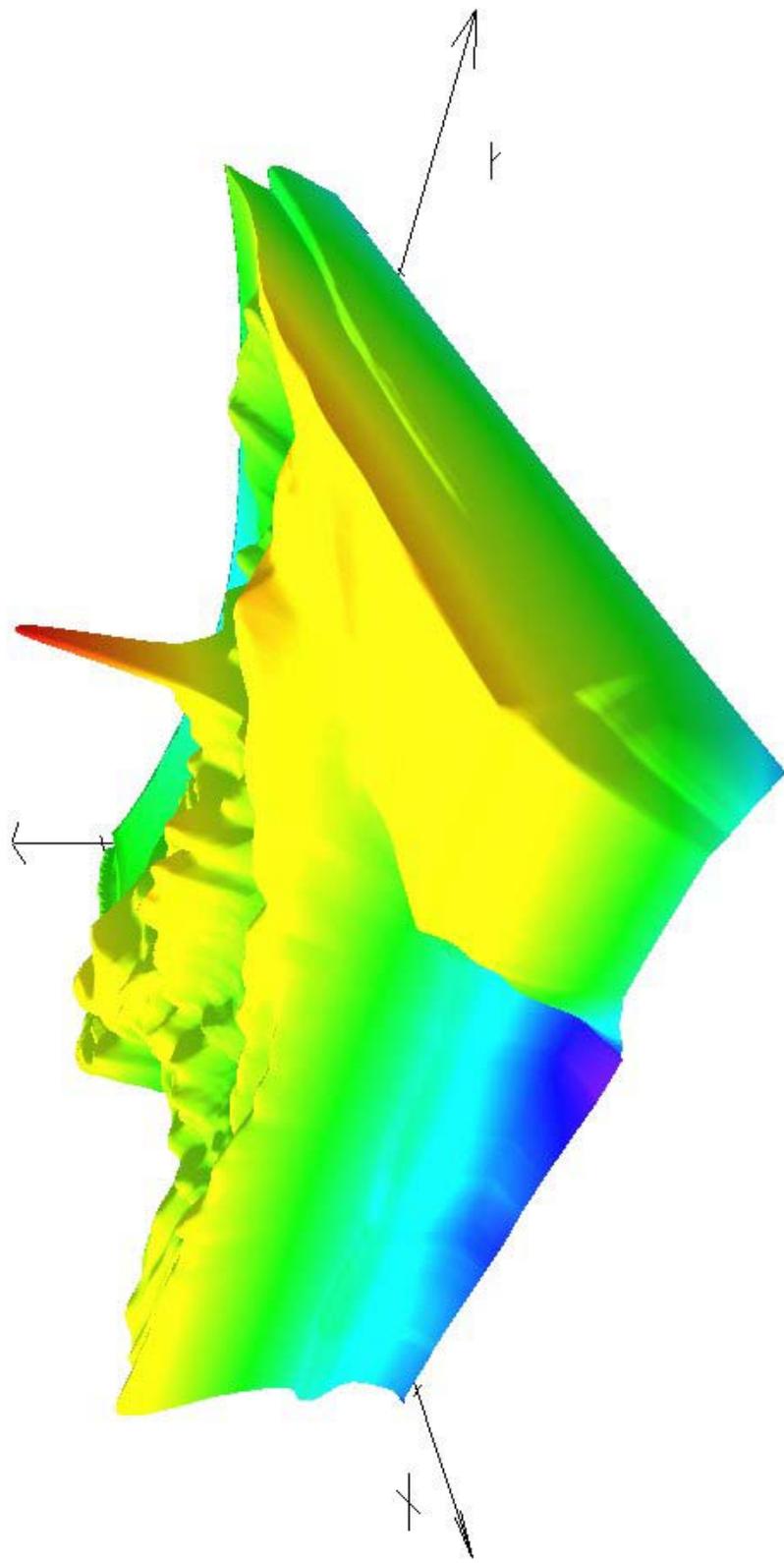


Figura 15: Superfície de *fitness* para a função Norma L1.

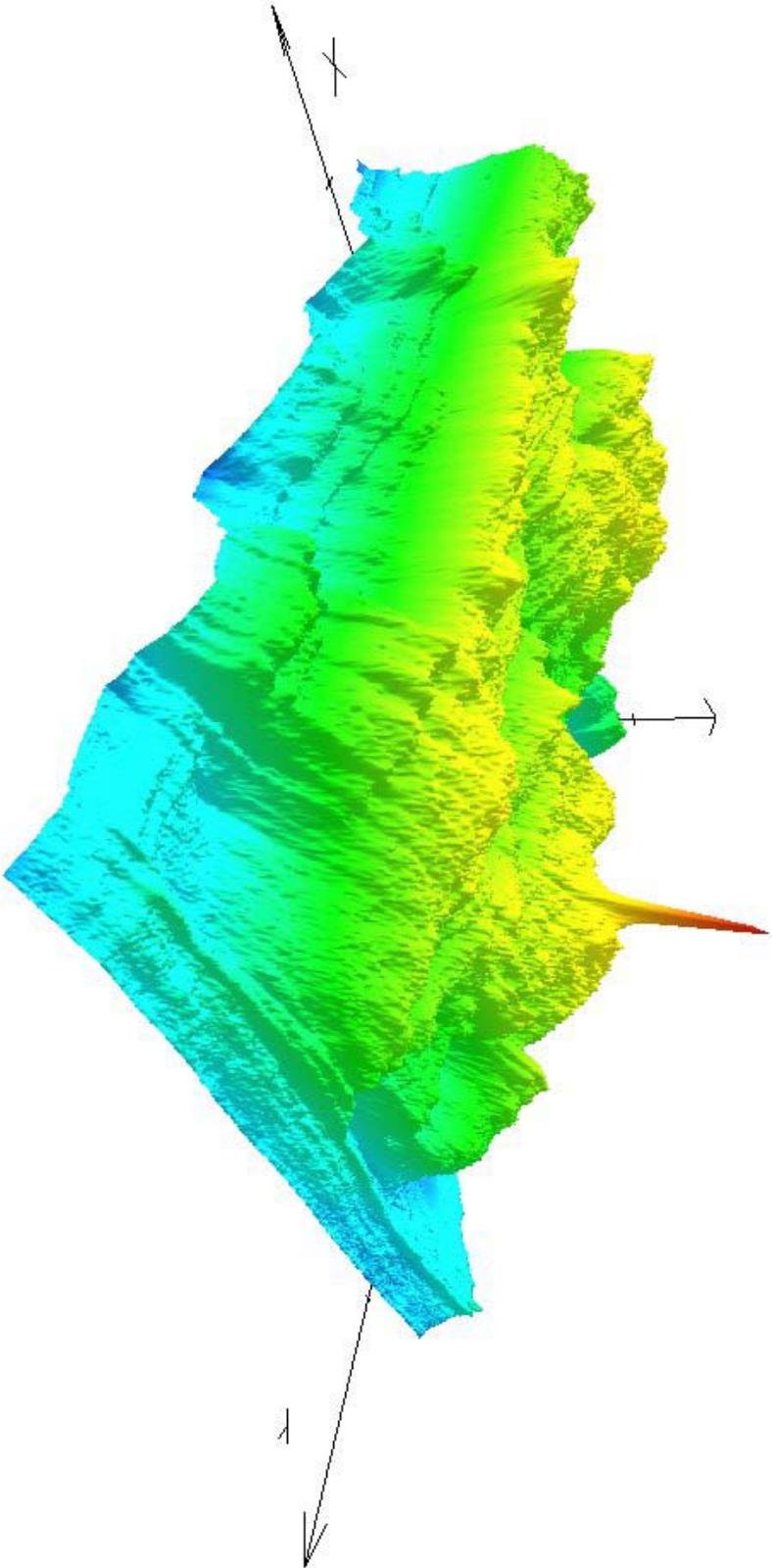


Figura 16: Superfície de *fitness* para a função Informação Mútua.

3.3 APLICAÇÃO DO PSO

O PSO é um método robusto e amplamente difundido na literatura, visto sua vasta aplicação em diversos problemas de otimização. O problema de reconhecimento como formulado neste trabalho, consiste na busca dos valores que determinam as coordenadas de translação, o fator de escala e o ângulo de rotação de um determinado objeto que pode ou não estar inserido em uma dada IP.

Como o TM necessita da comparação entre recortes da IP com a IR, seria então necessário que inúmeras comparações fossem realizadas para encontrar a melhor combinação de valores para a translação, rotação e ampliação/redução. Este fato torna inviável a utilização da força bruta, ou seja, comparar toda e qualquer combinação de valores para os parâmetros. Assim, foi escolhido o PSO por ser um método de busca eficiente, possibilitando que uma boa solução seja encontrada em tempo aceitável.

3.3.1 Detalhes de Implementação

O algoritmo implementado para o PSO é apresentado na seção 2.2. A versão básica do PSO foi empregada como solução para este problema em função da facilidade de compreensão bem como simplicidade de codificação, o que permite uma implementação compacta e eficiente do método.

Cada partícula que compõe a população do PSO é um vetor de dimensão 4, no qual cada dimensão armazena o valor de um dos parâmetros da solução do problema de reconhecimento. Os valores possíveis para cada dimensão estão dentro de intervalos definidos em tempo de execução e que, em parte dependem da IP utilizada. A tabela 3 mostra os intervalos utilizados em cada uma das dimensões, onde m e n são, respectivamente, a largura e a altura da IP (em *pixels*). Os intervalos para os valores de s e θ foram definidos para a redução de parte do espaço de busca, objetivando melhorar a taxa de convergência do método.

Tabela 3: Intervalos de valores permitidos para cada um dos parâmetros de uma solução.

Parâmetro	Intervalo
x	$[0..m - 1]$
y	$[0..n - 1]$
s	$[0, 5..1, 5]$
θ	$[-\pi/2.. \pi/2]$

Os valores da tabela 3 são utilizados como limites inferiores e superiores no momento da inicialização aleatória de cada partícula da população. Após a inicialização, o método entra em um laço de repetição, onde a condição de parada é atingir o número máximo de iterações cujo valor deve ser especificado. Em cada iteração do laço, são realizadas duas operações, a

atualização de cada partícula e a avaliação dos seus valores de *fitness*.

A atualização da partícula compreende a realização da movimentação da partícula no espaço de busca com base na velocidade calculada. O cálculo da velocidade utiliza a posição atual da partícula, a velocidade na iteração anterior e também a posição da melhor partícula de toda a população. A equação 2 (pág.13) é empregada para a obtenção da velocidade da partícula na iteração atual. De posse da velocidade pode-se então mover a partícula para um novo ponto do espaço de busca, utilizando a equação 3 (pág.13).

No processo do cálculo da velocidade é feita o truncamento do valor calculado, caso este ultrapasse os limites apresentados na tabela 4. Se o novo valor for menor que o limite inferior do intervalo, então a velocidade passa a ter o valor do limite inferior e vice-versa. Note-se que o intervalo permitido varia de acordo com um parâmetro do método chamado V_{Max} . Este parâmetro tem grande influência no processo de busca, visto que é ele que determina o tamanho do passo de busca realizado por cada partícula. Valores altos deste parâmetro tornam a busca mais global, permitindo que a partícula visite pontos distantes no espaço de busca. Do contrário, valores baixos para V_{Max} fazem com que a partícula tenha um comportamento local de busca.

Tabela 4: Intervalo de valores permitidos para a velocidade.

Parâmetro	Intervalo
x	$-(m * V_{Max}) .. (m * V_{Max})$
y	$[-(n * V_{Max}) .. (n * V_{Max})]$
s	$[-(1,0 * V_{Max}) .. (1,0 * V_{Max})]$
θ	$[-(\pi/2 * V_{Max}) .. (\pi/2 * V_{Max})]$

Calculada a nova posição da partícula, é feita a avaliação de *fitness* da partícula naquela posição. Neste momento, caso o *fitness* obtido mostre que esta partícula é melhor que a partícula *gbest*, então aquela partícula passa ser a *gbest*. Assim, a atualização da partícula *gbest* é feita de forma assíncrona, o que segundo Wilke (2005) propicia uma rápida influência da melhor partícula em toda a população.

Durante o processo de busca o método pode ficar preso em um mínimo local, o que restringe o espaço de busca a uma pequena região. Se o ponto onde as partículas estão presas possuir um valor de *fitness* baixo provavelmente a solução fornecida pelo PSO será de baixa qualidade. Tendo isso em vista, é aplicada na implementação o que se chamou de explosão² da população, ou dizimação em massa. Assim, se em determinado número de iterações não ocorrer melhorias na partícula *gbest*, toda a população é reinicializada de forma aleatória. A única informação que permanece armazenada é a solução encontrada pela partícula *gbest*.

Ao final do laço de repetição, a partícula classificada como *gbest* possui a solução final.

²O termo explosão denota o fato de que após a explosão as partículas são espalhadas de forma aleatória pelo espaço de busca, sem mudança no tamanho da população

Esta é a solução apresentada para o usuário como sendo o ponto do espaço de busca mais similar com o objeto desejado. É importante ressaltar que o método, apesar de robusto, pode ficar preso em mínimos locais apresentando assim soluções de baixa qualidade. Este comportamento é inerente a métodos de busca aproximados.

3.3.2 Ajuste de parâmetros do PSO

O PSO tem associado a si um conjunto de parâmetros utilizados para ajustar o comportamento das partículas e da população como um todo durante o processo de busca. Tais parâmetros são o tamanho da população, a quantidade máxima de iterações, os parâmetros de aprendizagem c_1 e c_2 e o fator de inércia w , utilizados na equação 2 (pág.13). Outros dois parâmetros são os valores de V_{Max} e a quantidade de iterações antes da explosão (*Explosao*).

O correto ajuste de tais parâmetros tem uma grande influência sobre o PSO, permitindo que este possa realizar uma boa exploração do espaço de busca, bem como obter soluções com um refinamento adequado. A situação ideal seria que tais parâmetros fossem ajustados dinamicamente durante o processo de busca. Alguns estudos propõem metodologias para auto-ajuste de parâmetros, mas no entanto não existe um consenso até o momento.

Eberhart e Shi (2001) sugerem um tamanho de população entre 20 a 50 partículas, número que depende do problema, assim como a quantidade máxima de iterações. Os valores das variáveis c_1, c_2 podem estar no intervalo de 0 a 4, mas tipicamente são ajustados com o valor 2. Geralmente usa-se o valor 0,5 para o fator de inércia w . Para V_{Max} geralmente indica-se o uso do intervalo de valores possíveis para a dimensão, assim se uma variável pode assumir valores entre -10 e 10, V_{Max} seria ajustado para 20.

Este conjunto de parâmetros depende do problema onde o PSO será aplicado. Assim, utilizar os valores sugeridos pela literatura poderia prejudicar a eficácia do método. Diante disso, foi aplicada uma metodologia para determinar o conjunto apropriado de parâmetros para o problema de reconhecimento.

Um experimento fatorial foi feito usando diferentes valores para os parâmetros c_1, c_2, w, V_{max} e *Explosao*. Os valores utilizados nos testes consideram um intervalo de 50% acima e abaixo dos valores sugeridos pela literatura como mostrado na Tabela 5. Um total de 243 combinações diferentes dos parâmetros foi gerado a partir desta tabela. Para evitar algum tipo de anomalia, para cada combinação de parâmetros foram executadas 100 rodadas independentes do PSO, todas elas utilizando sementes aleatórias diferentes. O tamanho da população para este experimento foi ajustado para 30 partículas e um máximo de 300 iterações.

Quatro diferentes testes foram definidos neste experimento, utilizando 4 imagens de paisagem e 4 objetos para serem reconhecidos. A escolha das imagens foi baseada na complexidade das mesmas, bem como o uso de cores e níveis de cinza. Com base neste critério de

Tabela 5: Conjunto de valores utilizados no experimento fatorial para determinação dos parâmetros do PSO .

Parâmetro	Valores Testados
c_1	[1;2;3]
c_2	[1;2;3]
w	[0.25;0.5;0.75]
V_{max}	[0.1;0.5;1]
$Explosao$	[10,50, ∞]

escolha, objetiva-se encontrar um conjunto de parâmetros para o PSO que seja adequado para uma grande variedade de instâncias do problema de reconhecimento.

O Teste 1 (IP mostrada na figura 17(a) e IR mostrada na figura 17(b)) pode ser considerado como o que possui o menor grau de dificuldade, visto que o objeto possui uma cor única na IP. O Teste 2 usa a mesma IP do Teste 1, mas com uma IR diferente (figura 17(c)). O objeto aqui é escuro e menor que aquele do Teste 1. O tamanho do objeto aumenta o grau de dificuldade, pois pode ser confundido com outras partes da imagem de paisagem, resultando em uma resposta errada.

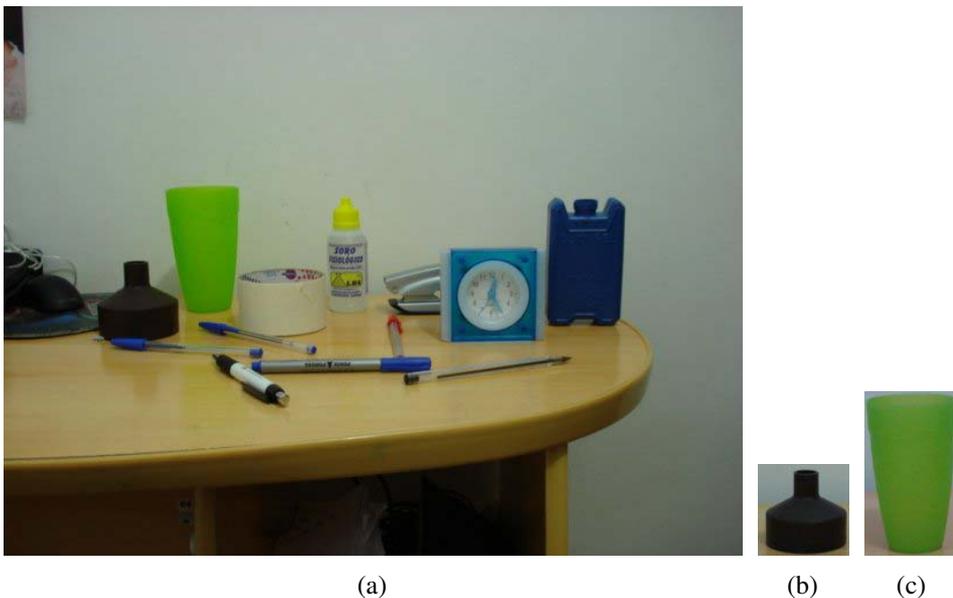


Figura 17: (a) IP (800 x 600) utilizada para o Teste 1 e Teste 2. (b) IR para o Teste 1. (c) IR para o Teste 2.

No Teste 3 (IP mostrada na figura 18(a) e IR mostrada na figura 18(b)) foram utilizadas imagens em níveis de cinza, onde a quantidade de informação é reduzida, pois somente o de intensidade é utilizado ao contrário das imagens em cores, onde três canais de cores formam o conteúdo da imagem. Este fato, por si só, torna este teste mais difícil do que os anteriores. Além disso, o conteúdo da imagem de paisagem possui características que aumentam ainda mais o grau de dificuldade. Existem vários objetos bastante semelhantes com o objeto a ser encontrado,

fato que indica a existência de vários mínimos locais, capazes de confundir o método. Outra característica é a presença de distorção (entenda-se distorção aqui como uma rotação em um plano diferente do x,y), causando a impossibilidade de um casamento perfeito entre a imagem de referência e a imagem de paisagem, modificando, assim, a superfície de busca visualizada pelo PSO.

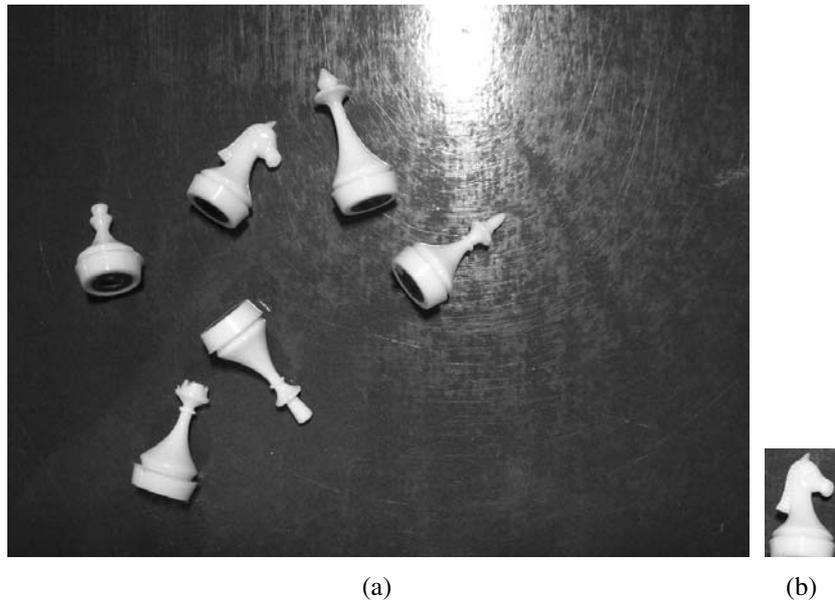


Figura 18: (a) IP (1024 x 768) para o Teste 3. (b) IR para o Teste 3.

As imagens das figuras 19(a) e 19(b) são, respectivamente, a IP e a IR que compõem o Teste 4. Aqui as imagens estão em níveis de cinza e a IP também possui a característica de possuir vários mínimos locais, fazendo com que o objeto desejado seja difícil de ser encontrado, visto que o PSO pode ficar preso em um ponto do espaço de busca.

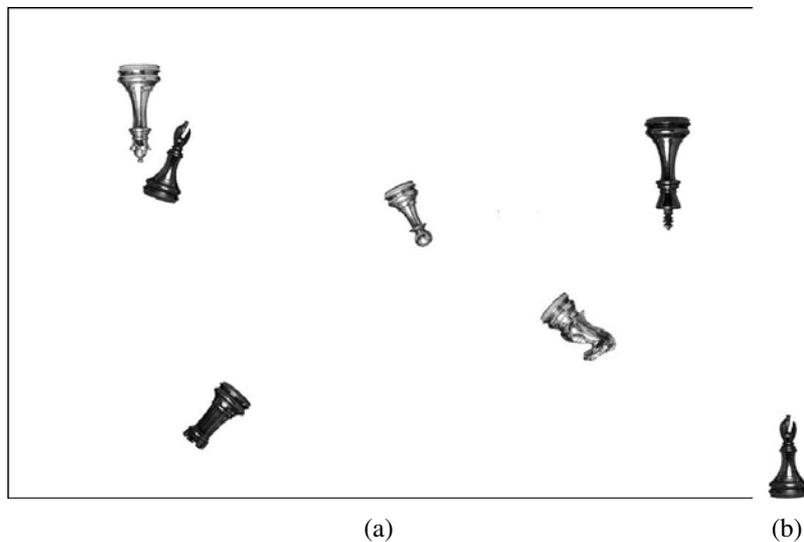


Figura 19: (a) IP (1024 x 768) para o Teste 4. (b) IR para o Teste 4.

Após a execução de 243 experimentos, com um total de 97200 rodadas do PSO, a média de *fitness* foi calculada para cada conjunto de parâmetros em cada um dos 4 testes. Os 243 conjuntos foram ordenados em ordem crescente da média de *fitness*. Quanto menor o valor da média maior o número de vezes que o objeto foi encontrado na imagem, visto que a L1 é uma medida que deve ser minimizada. Assim, o valor da média foi utilizada como medida da qualidade de cada um dos 243 conjuntos de parâmetros. Os 20 melhores conjuntos foram utilizados para a construção dos gráficos mostrados na figura 20. O conteúdo dos gráficos pode ser visto como um histograma da ocorrência de um determinado valor do parâmetro entre os 20 melhores conjuntos.

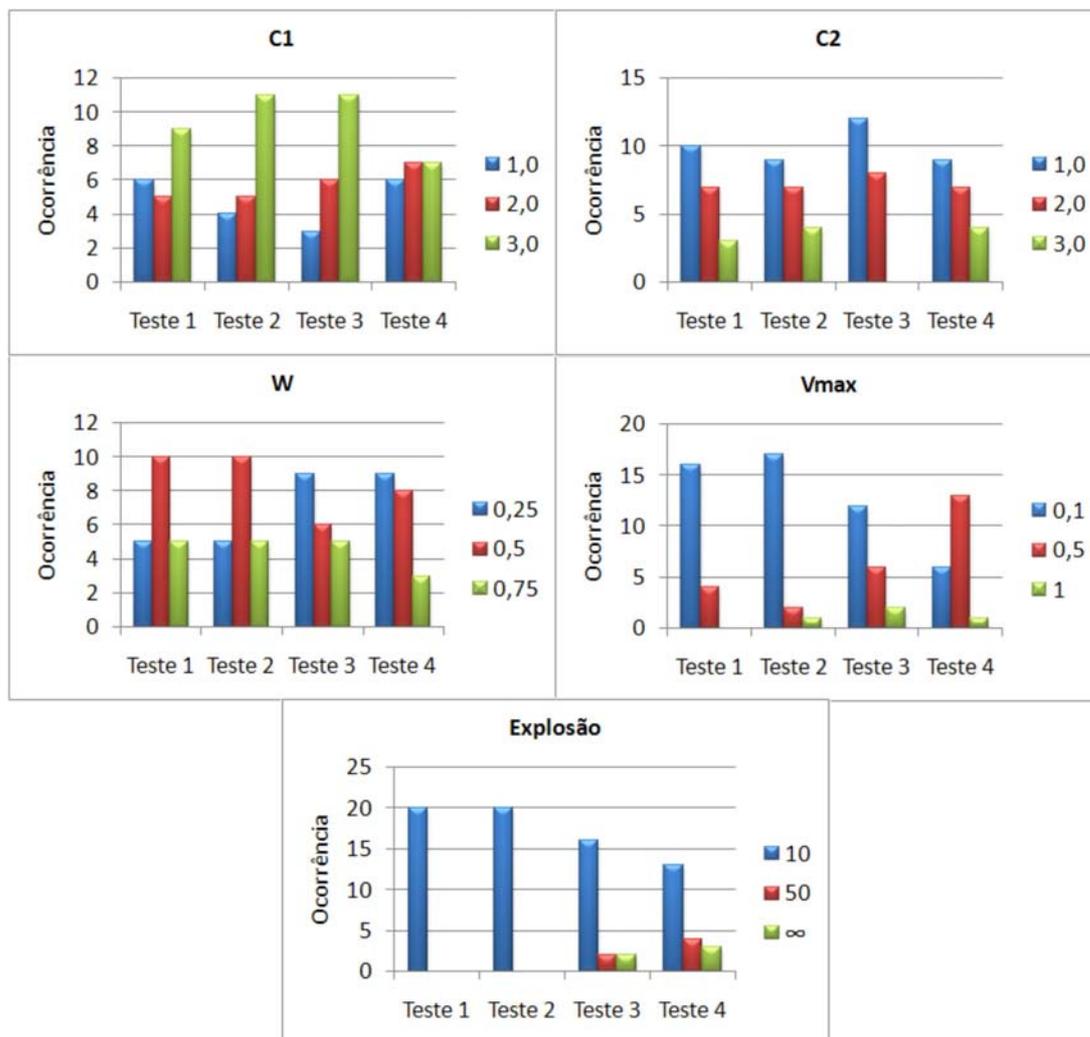


Figura 20: Distribuição da ocorrência de valores dos parâmetros do PSO para 4 problemas diferentes.

Analisando-se os gráficos é possível perceber que para o parâmetro c_1 , o valor 3 tem o maior número de ocorrências em qualquer um dos 4 testes, mostrando que provavelmente esse seja o valor adequado para c_1 . O mesmo vale para c_2 com o valor 1 e para *Explosao* com 10

iterações sem melhorias. Para os testes com imagens coloridas (Teste 1 e Teste 2) o valor que parece ser adequado para w é 0,5, e para imagens em níveis de cinza (Teste 3 e Teste 4) o valor de w é 0,25. A diferença entre os dois valores para o parâmetro w pode ter relação com o fato de que as imagens em níveis de cinza apresentam vários mínimos locais, fazendo com a opção por um baixo valor de inércia ajude no processo de busca. Verificou-se também que o valor adequado para o parâmetro V_{Max} é 0,1 para imagens coloridas. No entanto para imagens em níveis de cinza não existe uma clara definição para V_{Max} , assim foi empregado o valor 0,5.

As imagens de paisagem usadas para o ajuste de parâmetros do PSO são relativamente menores quando comparadas com aquelas que foram utilizadas nos experimentos de verificação. Assim, é esperado que o PSO enfrente maior dificuldade em encontrar soluções para imagens com grandes dimensões, considerando que o espaço de busca cresce consideravelmente com a modificação do tamanho da IP.

Experimentos aplicando o PSO, com uma população de 30 partículas e um máximo de 300 iterações, na busca de objetos em imagens de grande porte, mostraram que o método não forneceu boas soluções. Na tentativa de melhorar a eficácia do método, duas estratégias de modificações foram seguidas: aumentar o tamanho da população de partículas ou aumentar a quantidade máxima de iterações de busca, permitindo assim que o método consiga realizar um maior número de avaliações de possíveis soluções.

Dois conjuntos de testes foram formados, o primeiro com 30 partículas e 600 iterações, e o segundo com 60 partículas e 300 iterações. O esforço computacional dos dois conjuntos de teste é o mesmo, pois a quantidade de avaliações de *fitness* (multiplicação do tamanho da população pela quantidade de iterações) é exatamente a mesma.

Realizou-se um total de 100 rodadas independentes usando as duas estratégias. Foi verificado que as duas estratégias forneceram resultados cerca de 15% melhores que a configuração com 30 partículas e 300 iterações. Entre as estratégias, a configuração de 30 partículas e 600 iterações foi cerca de 5% melhor que a de 60 partículas e 300. Sendo assim o tamanho da população de indivíduos do PSO foi 30 partículas com 600 iterações no processo de busca.

O ajuste dos parâmetros de execução do PSO nesta aplicação é de grande importância, e possivelmente, também seja para outros problemas similares. Foi observado que o uso dos parâmetros sugeridos na literatura gerou resultados insatisfatórios. Mesmo que um experimento para encontrar o conjunto ideal de parâmetros seja uma tarefa de alto custo computacional, é um passo muito importante para o sucesso da implementação do método. Os resultados mostraram que, para diferentes classes de imagens, conjuntos diferentes de parâmetros devem ser utilizados para que o método obtenha um melhor desempenho.

3.4 CONJUNTO DE TESTES

Para a verificação da eficácia do PSO aplicado ao problema de reconhecimento de objetos foi definida uma série de testes. Tais testes incluem a qualidade das soluções fornecidas pelo PSO com diferentes combinações de transformações na imagem. Além disso, deseja-se saber se o PSO é capaz de reconhecer um objeto quando existem condições adversas da iluminação da IP ou na presença de ruídos.

Um banco de imagens sintéticas foi criado para permitir a avaliação da qualidade das soluções fornecidas pelo método. A partir deste banco pode-se conhecer *a priori* qual o ótimo global para uma determinada instância do problema de reconhecimento. Isso permite verificar a qualidade das soluções fornecida pelo PSO com alto grau de confiança.

Este conjunto de imagens possui transformações geométricas e a maioria de problemas comuns encontrados na aquisição de imagens, como modificação na iluminação, ruído e oclusão. O banco foi gerado com a ajuda do software *MODS* (PARKS; LEVINE, 2006). As imagens de paisagem foram obtidas do banco de imagens *McGill Calibrated Colour Image Database* (OLMOS; KINGDOM,), onde tais imagens possuem dimensões variadas. Por este motivo todas as imagens utilizadas foram ampliadas/reduzidas para uma dimensão padrão de 1000x1000 pixels.

A figura 21 mostra alguns exemplos de IP que podem ser encontradas no banco. As IR foram retiradas da biblioteca *Columbia University Image Library (COIL-100)* (NENE; NAYAR; MURASE, 2009). Todas as imagens da biblioteca *COIL-100* tem a mesma dimensão (128x128 pixel). A figura 22 mostra algumas miniaturas de objetos contidos na biblioteca. A figura 23 mostra uma imagem sintética que foi gerada como instância do problema de reconhecimento de objetos e à qual adicionou-se ruído Gaussiano, o que pode ser considerado um agravante no processo de reconhecimento.

3.5 COMPARAÇÃO COM OUTRA ABORDAGEM EVOLUCIONÁRIA

O método proposto neste trabalho foi comparado com um trabalho similar utilizando uma técnica de computação evolucionária. Em Silva, Lopes e Lima (2008) um algoritmo genético compacto com elitismo e mutação (emCGA) foi utilizado no reconhecimento de objetos em imagens digitais.

Um algoritmo genético compacto (CGA) é um algoritmo de estimação de distribuição, que utiliza dados estatísticos para a geração de descendentes, ao invés dos operadores genéticos da recombinação e mutação, como é o caso dos AG comuns. Várias versões do CGA podem ser encontradas na literatura, as quais aplicam diversos tipos de operadores para melhorar a qualidade de busca. O emCGA emprega o conceito de elitismo, que significa preservar certa

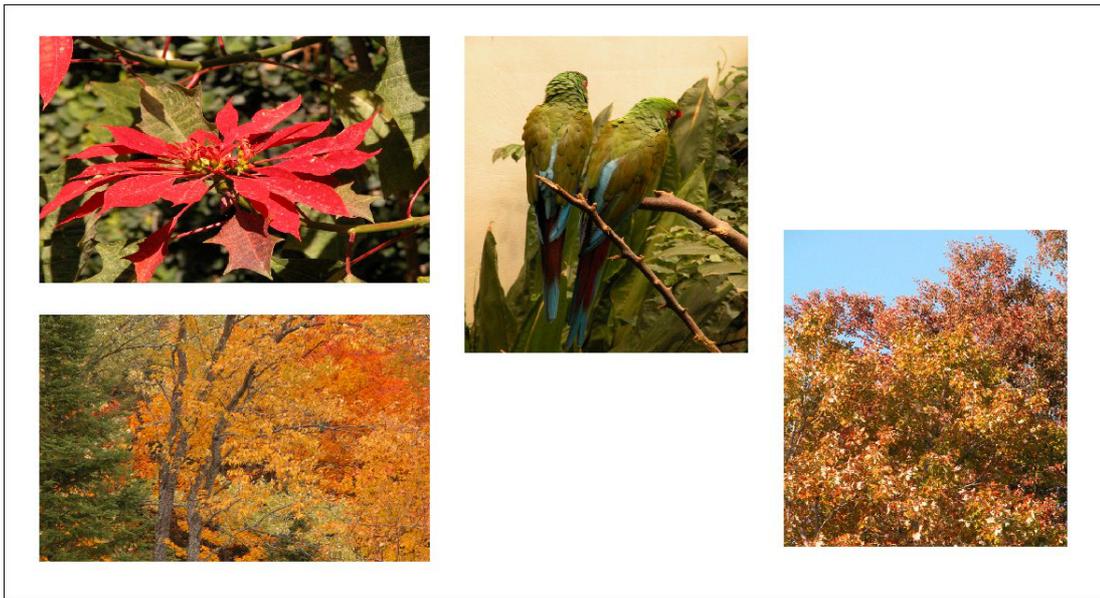


Figura 21: Exemplos de imagens de paisagem contidas no banco *McGill Calibrated Colour Image Database*.



Figura 22: Exemplos de objetos contidos no banco *COIL-100*.

quantidade de indivíduos na população entre uma geração e outra, juntamente com a mutação, que é a operação de modificar uma parte da solução aleatoriamente, de forma a introduzir diversidade na população (SILVA; LOPES; LIMA, 2008).

Existem algumas diferenças de modelagem entre os métodos. No emCGA uma solução é codificada de forma binária, introduzindo assim a discretização do espaço de busca bem como a necessidade de codificar/decodificar uma solução. O PSO permite que os dados que compõem uma solução sejam todos contínuos, o que facilita a formação de uma solução, bem como permite uma precisão maior na determinação da solução.

Outro aspecto importante é o tamanho do espaço de busca, que é diferente nos dois métodos, mesmo considerando o mesmo conjunto de imagens. Isso se deve ao fato de que uma solução encontrada pelo PSO é composta de quatro dimensões (x, y, s, θ) , enquanto que o



Figura 23: Uma instância do problema de reconhecimento de objetos gerada sinteticamente. Esta imagem possui uma série de objetos de interesse. Nesta imagem foi introduzido ruído Gaussiano com sigma igual a 0,5, simulando uma situação que pode ocorrer na aquisição da imagem.

trabalho que emprega o emCGA usa somente três dimensões (x, y, θ) .

Na construção do emCGA foi utilizado como ponto de origem do sistema de coordenadas o canto esquerdo inferior, enquanto que neste trabalho a origem foi considerada como sendo o ponto esquerdo superior da imagem. Sendo assim, foi necessário fazer uma conversão dos valores para a variável y das soluções encontradas pelo emCGA. Outra conversão necessária foi a transformação de graus para radianos no que diz respeito à variável θ . Cabe ressaltar que as duas conversões efetuadas não causam nenhuma interferência significativa na composição final da solução.

No próximo capítulo serão mostrados os experimentos realizados para a verificação da eficácia do PSO utilizando a função de *fitness* escolhida juntamente com o conjunto de parâmetros determinado, bem como os resultados obtidos.

CAPÍTULO 4

EXPERIMENTOS E RESULTADOS

4.1 EXPERIMENTOS PRELIMINARES

Para avaliar a viabilidade da utilização do PSO como solução para o problema de reconhecimento de objetos, uma série de experimentos preliminares foi realizada. O PSO padrão foi implementado, seguindo os passos descritos na seção 2.2. Estes primeiros experimentos foram realizados como parte de um trabalho prévio, antes de ser proposta a metodologia apresentada. Foi utilizada uma população de 40 partículas juntamente com o critério de parada sendo 800 iterações. Os parâmetros utilizados foram aqueles sugeridos pela literatura, que são $c_1 = c_2 = 2$ e $w = 0,5$ (EBERHART; SHI, 2001). O limite de velocidade V_{max} foi definido como sendo o limite superior de cada dimensão. Para evitar que o método sofra de convergência prematura, foram empregadas explosões (*Exp*) do enxame a cada 20 iterações sem melhora no *gbest*. Os experimentos relatados nas seções 4.1.1 e 4.1.1, foram publicados em (PERLIN; LOPES; CENTENO, 2008).

4.1.1 Experimento 1

Neste experimento o PSO foi testado usando uma IR branca-e-preta com 69×148 pixels, mostrada na figura 24(a). As figuras 24(b) e 24(c) mostram as IPs com 208×150 pixels, onde o objeto foi transladado e rotacionado. Além disso, foi adicionado às IPs ruído Gaussiano com média 0 e desvio padrão 0,5 e desvio padrão 2,0, respectivamente.

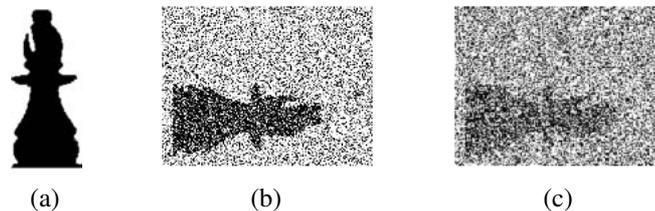


Figura 24: (a) IR. (b) IP com o objeto desejado transladado, rotacionado e degradado por ruído gaussiano com média 0 e desvio padrão 0,5. (c) Mesma IP degradada por ruído gaussiano com média 0 e desvio padrão 2,0.

Em ambos os casos, o PSO encontrou boas soluções, mostradas nas tabelas 6 e 7. O PSO facilmente encontrou as soluções, necessitando de somente 3440 e 3840 avaliações de *fitness*, respectivamente. Nestas tabelas, “Par.” denota os parâmetros da quádrupla que formam uma solução; “Ref.” são os valores “*ground truth*”; “PSO” denota os valores dos parâmetros

encontrados pelo PSO; e “Erro” é o erro relativo entre os valores encontrados pelo PSO e os valores da solução referência.

Tabela 6: Solução encontrada pelo PSO para a IR da figura 24(a) e IP da figura 24(b).

Par.	Ref.	PSO	Erro
y	84	84	0
x	106	107	1
s	1	1,006251	0,0062
θ	-1,57	-1,565450	0,0046

Tabela 7: Solução encontrada pelo PSO para a IR da figura 24(a) e IP da figura 24(c).

Par.	Ref.	PSO	Erro
y	84	84	0
x	106	107	1
s	1	0,986482	0,0136
θ	-1,57	-1,571803	0,0018

4.1.2 Experimento 2

Para testar a robustez do PSO em uma situação realista, foi utilizada uma IP em níveis de cinza, a qual contém vários objetos. A IR de 238 x 239 pixels é mostrada na figura 25(a), a qual se supõe que esteja presente na IP de 496 x 347 pixels mostrada na figura 25(b).

Analisando a IP visualmente, percebe-se que o objeto está transladado, rotacionado, reduzido e com perspectiva, ou seja, foi aplicada uma rotação no plano $\{y,z\}$. Além disso, o objeto desejado está parcialmente coberto por outro objeto (tesoura). Todas essas transformações configuram uma tarefa difícil para o método de reconhecimento automático de objetos. A solução encontrada é visualizada na figura 25(c), que é um recorte da IP.

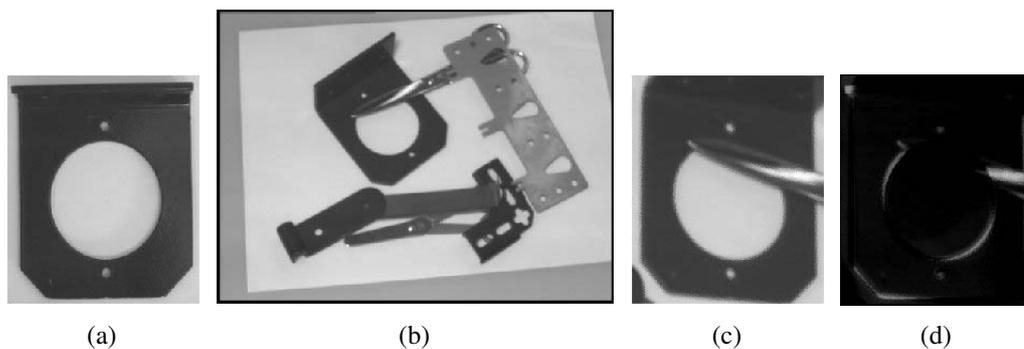


Figura 25: (a) IR. (b) IP. (c) Recorte da solução encontrada pelo PSO (c), quando foi utilizada a IR (a) e a IP (b). (d) Diferença entre a solução encontrada e a IR (a).

4.2 AVALIAÇÃO DO MÉTODO

Os experimentos relatados na seção 4.1 foram utilizados para determinar a viabilidade de empregar o PSO como possível solução para o problema de reconhecimento de objetos em imagens digitais. Os resultados preliminares mostraram que o método proposto é viável como uma solução para o problema. Sendo assim, os estudos para a determinação de uma

métrica adequada, como relatado na seção 3.2.3 foram realizados, bem como a aplicação de uma metodologia para a determinação de um conjunto adequado de parâmetros do PSO. Obtidos estes dados, foi realizada uma série de experimentos com o objetivo de avaliar o método, na questão de sensibilidade a degradações e transformações geométricas e também comparar o método proposto com outra abordagem evolucionária a fim de verificar o seu desempenho.

Os experimentos de sensibilidade relatados nas seções 4.2.1 a 4.2.5 utilizaram as IPs mostradas nas figuras 26(a), 26(b) e 26(c), que foram geradas como explanado na seção 3.4. Duas IR diferentes foram utilizadas contendo dois objetos com variações de formato, cor e tamanho. A primeira IR é mostrada na figura 26(d) e foi chamada de Obj 1 nos experimentos realizados. A segunda é mostrada na figura 26(e) e foi chamada de Obj 2. O fundo preto dos objetos foi removido através da aplicação de uma máscara fornecida pela biblioteca COIL-100.

Para verificar se o PSO teria o seu desempenho afetado pela quantidade de informação presente nas imagens onde o reconhecimento deveria ser realizado, os experimentos de sensibilidade foram realizados tanto com as imagens coloridas quanto com as imagens em níveis de cinza, como mostrado nas figuras 27(a), 27(b), 27(c), 27(d) e 27(e).

A implementação da norma L1 da OpenCV combina os valores de cada um dos três canais de cores para a obtenção da diferença de dois *pixels*. Como cada canal pode assumir 255 possibilidades e existem três canais, a maior diferença entre dois *pixels*, segundo a norma L1 é $765 = 255 \times 3$. Com base nisso, os valores da norma L1 podem ser de grande magnitude, gerando assim explosões numéricas dificultando a análise dos resultados.

Para evitar tais problemas os valores de *fitness* da melhor solução encontrada pelo PSO foram normalizados pelo maior valor possível que a norma L1 poderia alcançar para cada um dos objetos. O valor máximo que a norma L1 pode assumir para um determinado objeto, pode ser obtido multiplicando o valor 765 pela quantidade de *pixels* que compõem a imagem do objeto extraída pela máscara descrita anteriormente.

De posse deste valor, a normalização foi realizada utilizando a equação 14, onde $L1Max_{obj}$ é o maior valor da norma L1 para o objeto e *Fitness* é a solução final fornecida pelo PSO. O valor obtido após a normalização, os valores de *fitness* ficam no intervalo [0..1].

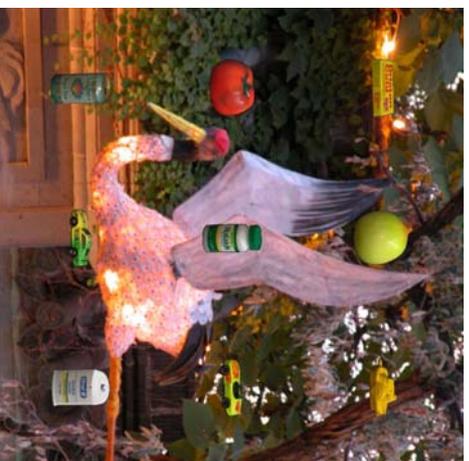
$$FitnessNormalizado = \frac{(L1Max_{obj} - Fitness)}{L1Max_{obj}} \quad (14)$$

4.2.1 Sensibilidade à Translação

A translação aqui é caracterizada pela inserção dos objetos em diferentes pontos da IP. Assim, a intenção aqui é verificar se o método consegue determinar quais são as coordenadas (x,y) do centro do objeto na IP sem nenhum outro tipo de transformação.



(a)



(b)



(c)

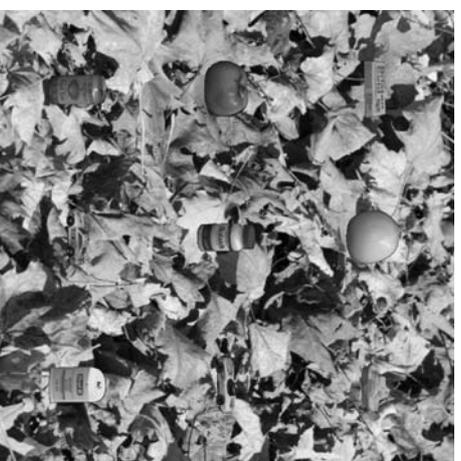


(d)



(e)

Figura 26: (a,b,c) Imagens de paisagem coloridas utilizada nos testes. (d,e) Imagens de referência que contem os objetos desejados.



(a)



(b)



(c)



(d)



(e)

Figura 27: (a,b,c) Imagens de paisagem em níveis de cinza utilizada nos testes. (d,e) Imagens de referência em níveis de cinza.

Os resultados do experimento são mostrados nas tabelas 8 e 9, sendo que as medidas foram calculadas com base nas diferenças entre os valores encontrados pelo PSO e o valor de “ground truth” (GT), fornecido pelo software de geração das imagens. As medidas são referentes à 100 execuções independentes do PSO.

Tabela 8: Média das estatísticas das diferenças entre os valores encontrados pelo PSO e o valor de “ground truth” para a IR Obj 1 sendo localizada nas três IPs, tanto colorida quanto em níveis de cinza, na ocorrência de translação.

Param.	Colorida			Cinza			Unid.
	Média	Desv. Pdr.	Mediana	Média	Desv. Pdr.	Mediana	
x	3,56	31,50	0,00	66,06	160,19	1,00	pixels
y	3,26	23,69	1,00	87,50	170,05	1,00	pixels
θ	0,01	0,03	0,00	0,10	0,25	0,01	rad.
s	0,01	0,03	0,01	0,11	0,18	0,01	-

Tabela 9: Média das estatísticas das diferenças entre os valores encontrados pelo PSO e o valor de “ground truth” para a IR Obj 2 sendo localizada nas três IPs, tanto colorida quanto em níveis de cinza, na ocorrência de translação.

Param	Colorida			Cinza			Unid.
	Média	Desv. Pdr.	Mediana	Média	Desv. Pdr.	Mediana	
x	76,23	123,59	1,00	164,97	150,13	15,00	pixels
y	102,28	152,89	1,00	170,69	162,57	2,00	pixels
θ	0,14	0,23	0,02	0,20	0,31	0,07	rad.
s	0,14	0,16	0,02	0,22	0,18	0,10	-

4.2.2 Sensibilidade à Escala e Rotação

Como mencionado anteriormente, a posição de um objeto na IP é definida como uma quádrupla. Em Perlin, Lopes e Centeno (2008) foi observado que o casamento entre duas imagens é bastante sensível à mudança de escala e rotação. Desta forma, o objetivo deste experimento é verificar o desempenho do método em relação a tais mudanças.

Os objetos introduzidos nas IPs foram submetidos à mudança de escala e rotação. A escala foi variada no intervalo de 0,7 (redução do objeto original) a 1,3 (ampliação do objeto original) com passos de 0,1, gerando 6 IPs diferentes, como mostradas na figura 28(a). O intervalo de variação da rotação foi de 15° a 90° com passos de 15°, formando assim um conjunto de 6 diferentes IPs, como mostradas na figura 29(a).

A figura 28(b) mostra a média dos erros relativos para os três conjuntos de IPs levando em consideração a mudança de escala. Da mesma forma, a figura 29(b) mostra a média dos erros relativos, para os três conjuntos de IPs, quando o objeto é rotacionado. Para cada fator de escala ou ângulo de rotação, foi calculada a média de *fitness* de 100 execuções independentes

do método. O erro relativo é calculado usando a média de *fitness* obtida da situação onde não existe nenhuma transformação (seção 4.2.1). A média dos três desvios padrão para cada fator de escala e para cada ângulo de rotação são mostradas nas tabelas 10 e 11, respectivamente.

Tabela 10: Média dos desvios padrão para cada fator de escala, tanto colorida quanto em níveis de cinza, para as duas IR.

Escala	IR Obj 1		IR Obj 2	
	Colorida	Níveis de Cinza	Colorida	Níveis de Cinza
0,70	0,05	0,03	0,06	0,03
0,80	0,03	0,03	0,06	0,04
0,90	0,02	0,03	0,06	0,05
1,00	0,01	0,02	0,05	0,05
1,10	0,01	0,02	0,05	0,06
1,20	0,01	0,02	0,06	0,05
1,30	0,01	0,02	0,06	0,05

Tabela 11: Média dos desvios padrão para cada fator de escala, tanto colorida quanto em níveis de cinza, para as duas IR.

Rotação	IR Obj 1		IR Obj 2	
	Colorida	Níveis de Cinza	Colorida	Níveis de Cinza
0	0,01	0,02	0,05	0,05
15	0,03	0,02	0,05	0,05
30	0,02	0,02	0,06	0,04
45	0,01	0,02	0,06	0,04
60	0,02	0,03	0,06	0,04
75	0,02	0,03	0,07	0,04
90	0,01	0,03	0,06	0,04

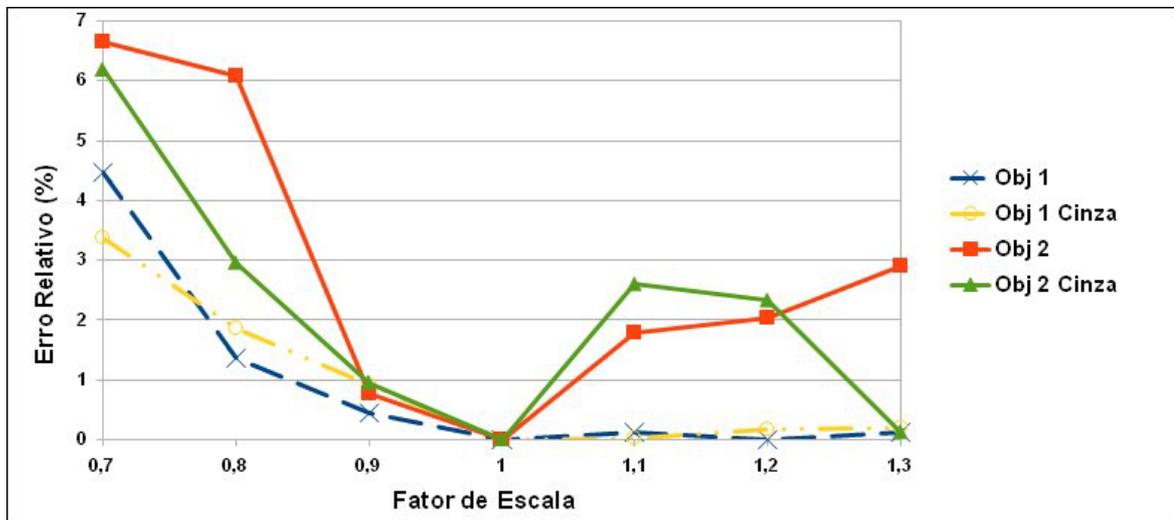
4.2.3 Sensibilidade a Ruído

Para verificar a robustez do método no reconhecimento de imagens com ruído, foram adicionados às IPs dois tipos de ruído: ruído Gaussiano e sal-e-pimenta. A quantidade de ruído sal-e-pimenta adicionado à IP variou no intervalo de 10% a 50%. Para o ruído Gaussiano, foi utilizada média 0 e desvio padrão variando no intervalo de 0,1 a 0,5.

Para cada um dos conjuntos de IPs, foi calculado o erro relativo entre a média de *fitness* de 100 execuções do PSO com a imagem sem presença de ruído e as imagens com presença de ruído. Após, para cada um dos 5 níveis de ruídos foi calculada a média para os erros relativos fornecidos por cada um dos três conjuntos de IPs, gerando as figuras 30(a) e 30(b). Nas tabelas 12 e 13 apresentam a média dos desvios padrão, fornecidos por cada um dos três conjuntos de IPs, para o reconhecimento das IR Obj 1 e IR Obj 2, respectivamente.



(a)

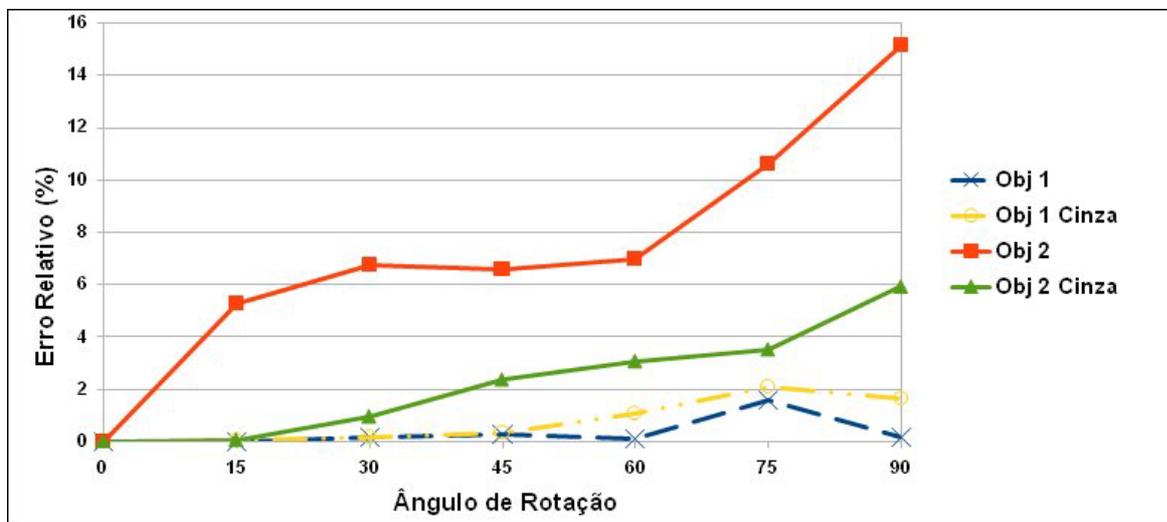


(b)

Figura 28: (a) Conjunto de IPs com variação no fator de escala aplicado nos objetos inseridos. (b) Média entre os erros relativos fornecidos pelos três conjuntos de testes, para os diferentes fatores de escala, no reconhecimento das IR Obj 1 e Obj 2, tanto coloridas quanto em níveis de cinza.

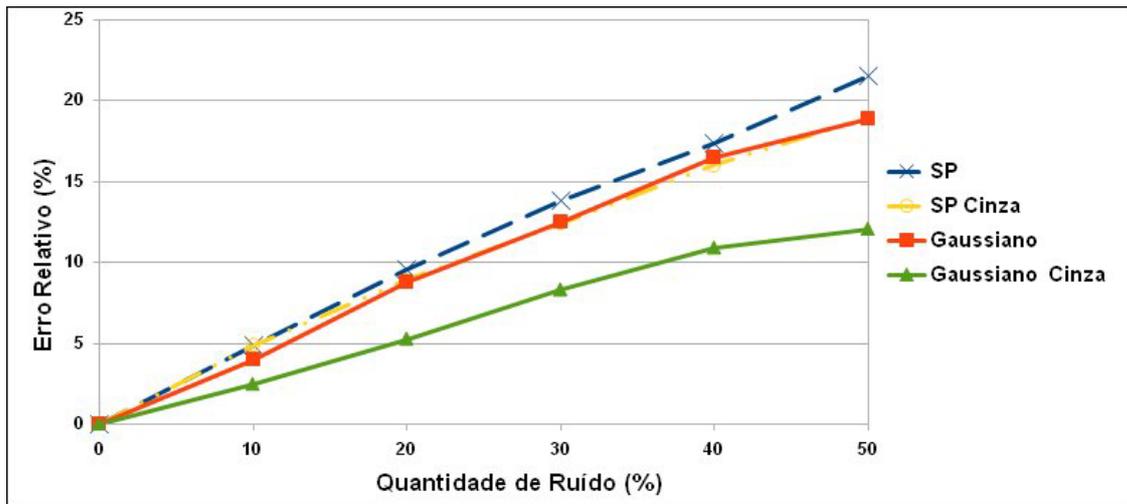


(a)

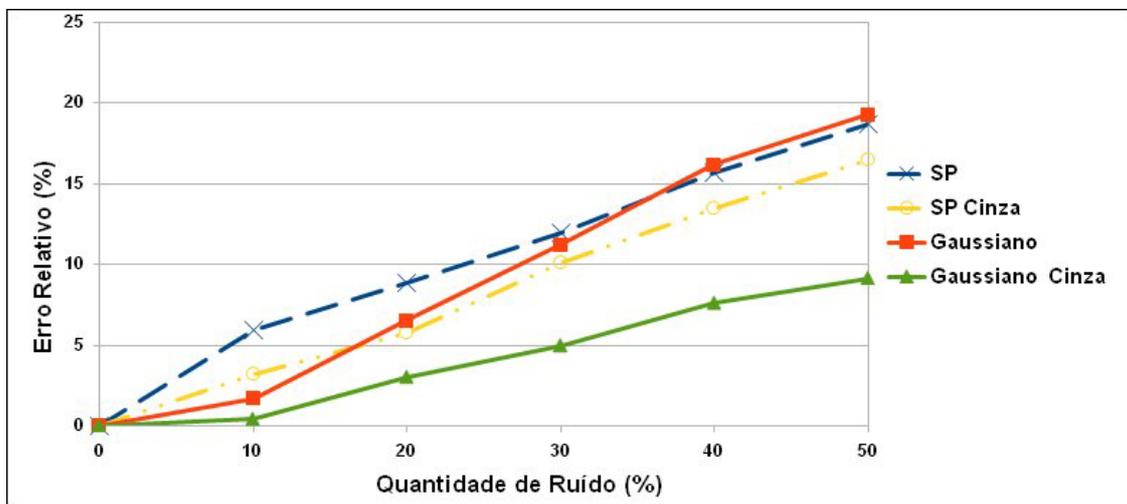


(b)

Figura 29: (a) Conjunto de IPs com variação no ângulo de rotação aplicado nos objetos inseridos. (b) Média entre os erros relativos fornecidos pelos três conjuntos de testes, para os diferentes ângulos de rotação, no reconhecimento das IR Obj 1 e Obj 2, tanto coloridas quanto em níveis de cinza.



(a)



(b)

Figura 30: (a) Média dos erros relativos para o reconhecimento da IR Obj. 1 colorida e em níveis de cinza. (b) Média dos erros relativos para o reconhecimento da IR Obj. 2 colorida e em níveis de cinza.

4.2.4 Sensibilidade à Oclusão

Em situações reais, o objeto a ser reconhecido na IP pode estar parcialmente ocluído por outro objeto. Este fato torna o reconhecimento difícil, pois somente uma parte do objeto pode ser utilizada para a comparação. Este experimento objetiva a avaliação do comportamento do método proposto no reconhecimento de um objeto ocluído.

Ao invés de sobrepor dois objetos para simular a oclusão, foi empregada uma técnica similar àquela utilizada por (NENE; NAYAR; MURASE, 2009). Naquele trabalho parte do objeto foi substituído por ruído aleatório, porém o contorno do objeto ainda é mantido. Foi decidido então mesclar parte do objeto com o fundo simulando um *clipping*, como mostrado na figura 31(b), de forma a não manter as informações relativas do contorno.

Tabela 12: Média dos desvios padrão para os níveis de ruído S&P e Gaussiano inseridos nas IPs, tanto coloridas quanto em níveis de cinza, para o reconhecimento da IR Obj 1.

Nível de Ruído	S&P		Gaussiano	
	Colorida	Néis de Cinza	Colorida	Níveis de Cinza
0	0,01	0,01	0,02	0,02
10	0,01	0,02	0,02	0,02
20	0,01	0,02	0,02	0,01
30	0,02	0,02	0,01	0,01
40	0,01	0,01	0,01	0,01
50	0,01	0,02	0,01	0,02

Tabela 13: Média dos desvios padrão para os níveis de ruído S&P e Gaussiano inseridos nas IPs, tanto coloridas quanto em níveis de cinza, para o reconhecimento da IR Obj 2.

Nível de Ruído	S&P		Gaussiano	
	Colorida	Níveis de Cinza	Colorida	Níveis de Cinza
0	0,05	0,05	0,05	0,05
10	0,05	0,03	0,04	0,05
20	0,04	0,03	0,03	0,04
30	0,03	0,04	0,03	0,03
40	0,03	0,03	0,03	0,02
50	0,02	0,03	0,02	0,01

As IPs contêm objetos com 25% de oclusão, como mostrado na figura 31(a). O desempenho do PSO é mostrado nas tabelas 14 e 15, onde são apresentadas as medidas obtidas a partir da diferença entre os valores fornecidos pelo método e os valores de “*ground truth*”, para a IR Obj. 1 e IR Obj. 2, respectivamente. As medidas são referentes à 100 execuções independentes do PSO.

Tabela 14: Média das estatísticas das diferenças dos valores encontrados pelo PSO e o valor de “*ground truth*” para a IR Obj 1 ser localizada nas três IPs com oclusão, tanto colorida quanto em níveis de cinza, com oclusão parcial.

Param.	Colorida			Cinza			Unid.
	Média	Desv. Pdr.	Mediana	Média	Desv. Pdr.	Mediana	
x	13,29	23,81	9,50	151,46	203,13	129,00	pixels
y	11,07	18,87	10,00	214,64	181,74	187,30	pixels
θ	0,21	0,11	0,21	0,37	0,39	0,18	rad
s	0,25	0,13	0,25	0,34	0,18	0,35	-

4.2.5 Sensibilidade a Contraste e Brilho

Em determinadas situações podem ocorrer mudanças de brilho e contraste durante a aquisição das imagens. Assim como no caso de ruído, técnicas de processamento de imagens

Tabela 15: Média das estatísticas das diferenças entre os valores encontrados pelo PSO e o valor de “*ground truth*” para a IR Obj 2 ser localizada nas três IPs com oclusão, tanto colorida quanto em níveis de cinza, com oclusão parcial.

Param.	Colorida			Cinza			Unid.
	Média	Desv. Pdr.	Mediana	Média	Desv. Pdr.	Mediana	
x	112,55	93,96	101,5	176,05	181,53	123,74	pixels
y	142,23	116,28	20,33	174,64	29	117,38	pixels
θ	0,2	0,27	0,11	0,24	0,28	0,23	rad
s	0,13	0,14	0,06	0,23	0,18	0,16	-

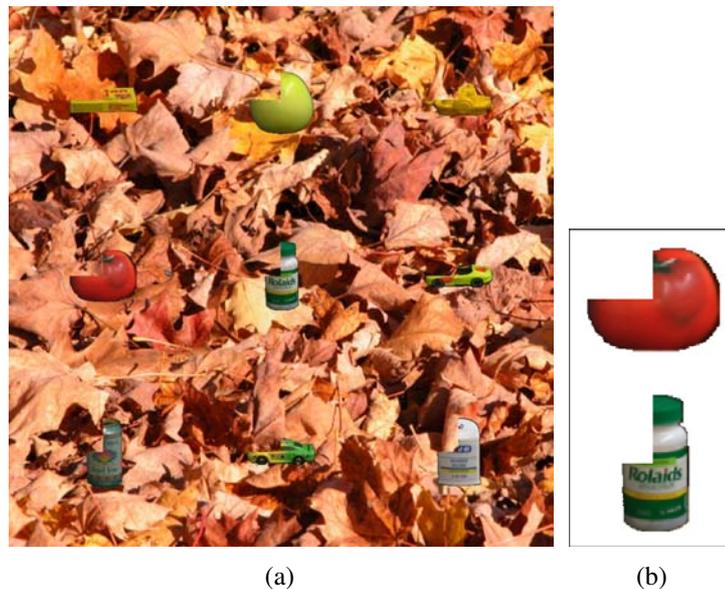


Figura 31: (a) Uma das imagens de paisagem usada no experimento com oclusão de 25% da superfície dos objetos. (b) Detalhe da simulação de oclusão nos objetos.

podem ser aplicadas às imagens para a correção de brilho e contraste, mas existe a necessidade de escolher a técnica a ser utilizada, bem como os parâmetros adequados. Assim, se o método for capaz de reconhecer os objetos, sem a necessidade de correções, pode-se poupar tempo e esforço.

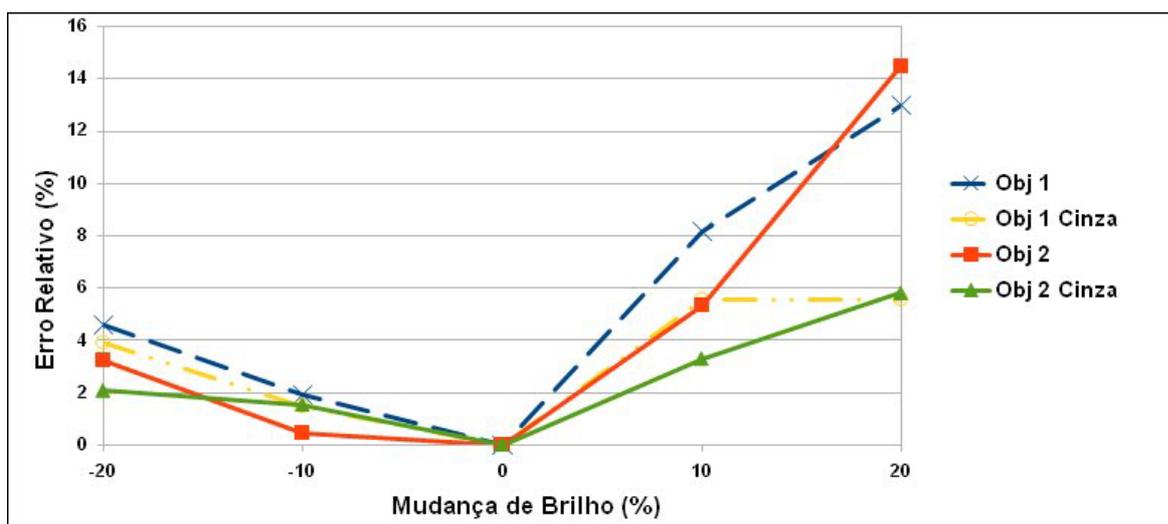
Foram realizados experimentos para determinar a eficácia do método, quando simuladas mudanças de brilho e contraste na IP. A mudança de brilho variou de -20% a +20%, com passos de 10%, ou seja, o valor de cada pixel é adicionado ou subtraído a uma constante de acordo com a porcentagem, gerando conjuntos de IPs como mostrado na figura 32(a). As mesmas condições de variação foram feitas para o contraste, gerando conjuntos de IPs como mostrado na figura 33(a). A modificação é feita através da técnica de alargamento/estreitamento de contraste, assim um valor de -10% significa um estreitamento na distribuição das cores.

Para cada conjunto de IPs, foram calculados os erros relativos entre a média de *fitness* de 100 execuções independentes do experimento padrão (condições de brilho e contrastes normais)

e a média de *fitness* das execuções com mudança de brilho e de contraste. O desempenho do PSO é mostrado através dos gráficos das figuras 32(b) e 33(b), onde cada ponto da curva representa a média dos erros relativos de cada um dos conjuntos de IPs. Para os testes de mudança de brilho e contraste são apresentadas nas tabelas 16 e 17, as médias dos desvios padrão obtidos para cada conjunto de IPs.



(a)

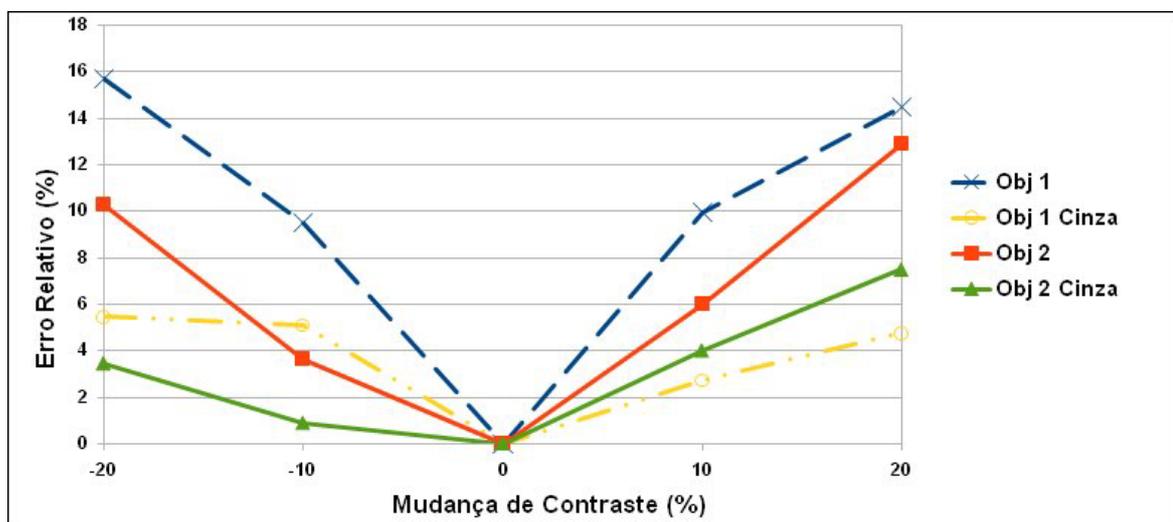


(b)

Figura 32: (a) Conjunto de IPs com variação de brilho. (b) Média entre os erros relativos fornecidos pelos três conjuntos de testes, para os diferentes níveis de brilho, no reconhecimento das IR Obj 1 e Obj 2, tanto coloridas quanto em níveis de cinza.



(a)



(b)

Figura 33: (a) Conjunto de IPs com variação de contraste. (b) Média entre os erros relativos fornecidos pelos três conjuntos de testes, para os diferentes níveis de contraste, no reconhecimento das IR Obj 1 e Obj 2, tanto coloridas quanto em níveis de cinza.

4.3 COMPARAÇÃO COM OUTRA ABORDAGEM EVOLUCIONÁRIA

Como já explanado na seção 3.5, o PSO foi comparado com uma implementação de um emCGA para o reconhecimento de objetos. Três conjuntos de IR e IP foram utilizados na comparação dos dois métodos.

Tabela 16: Média dos desvios padrão para os testes de mudança de brilho, tanto coloridas quanto em níveis de cinza.

Mudança de Brilho	IR Obj 1		IR Obj 2	
	Colorida	Níveis de Cinza	Colorida	Níveis de Cinza
-20	0,01	0,01	0,02	0,03
-10	0,02	0,02	0,04	0,04
0	0,01	0,02	0,05	0,05
10	0,01	0,01	0,04	0,03
20	0,02	0,02	0,02	0,02

Tabela 17: Média dos desvios padrão para os níveis de ruído S&P e Gaussiano inseridos nas IPs, tanto coloridas quanto em níveis de cinza, para o reconhecimento da IR Obj 2.

Nível de Ruído	IR Obj 1		Ir Obj 2	
	Colorida	Níveis de Cinza	Colorida	Níveis de Cinza
-20	0,01	0,02	0,02	0,01
-10	0,01	0,01	0,03	0,03
0	0,01	0,02	0,05	0,05
10	0,02	0,01	0,05	0,03
20	0,01	0,01	0,04	0,01

O primeiro conjunto utilizado no Teste 1, é composto pela IR mostrada na figura 34(a) e pela IP mostrada na figura 34(b). Observa-se que existem vários objetos de mesma tonalidade que o objeto desejado, indicando a presença de vários ótimos locais. Além disso, o objeto desejado sofreu uma rotação no plano $\{y,z\}$, dificultando assim o reconhecimento.

O segundo conjunto utilizado no Teste 2 é composto pela IR mostrada na figura 35(a) e pela IP mostrada na figura 35(b). Pode-se verificar que este conjunto possui características semelhantes ao conjunto do Teste 1.

O Teste 3 utilizou a IR mostrada na figura 36(a) e a IP mostrada na figura 36(b), o que caracteriza este experimento como sendo de alto grau de dificuldade, levando em conta que existem vários ótimos locais na IP. O retângulo em amarelo presente nas IPs é utilizado para destacar a solução encontrada pelo PSO.

Nas tabelas 18, 19 e 20, são mostrados os resultados para a melhor solução encontrada pelo PSO, pelo emCGA e pelo algoritmo de busca exaustiva (ABE) implementado por Silva, Lopes e Lima (2008), juntamente com as informações para cada dimensão da solução.

Além da qualidade da solução encontrada pelo método, é necessário verificar o esforço computacional despendido pelo método para conseguir encontrar tal solução. Uma medida que pode ser utilizada para comparação de esforço computacional, sem depender de detalhes de implementação, bem como ambiente de execução dos testes, é a quantidade total de avaliações de *fitness*. O gráfico da figura 37 mostra uma comparação do total de avaliações de *fitness* entre os três métodos para os três conjuntos de imagens testadas.

Tabela 18: Soluções fornecidas pelos métodos para encontrar a IR mostrada na figura 34(a) na IP mostrada na figura 34(b).

Métodos	x	y	θ	s	Número de Avaliações
<i>PSO</i>	326	215	32,62	0,98	5,16E+04
<i>emCGa</i>	327	216	32,41	-	9,58E+04
<i>ABE</i>	327	216	32,41	-	4,03E+08

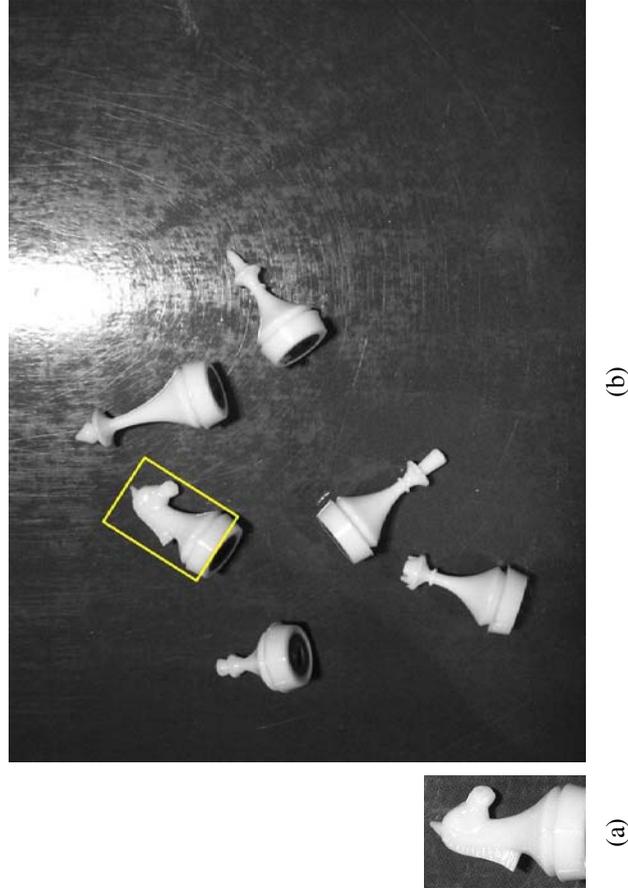


Figura 34: IR (a) e IP (c) utilizada em um dos testes de comparação. O retângulo destaca a solução encontrada pelo PSO.

Tabela 19: Soluções fornecidas pelos métodos para encontrar a 35(a) na IP 35(b).

Métodos	x	y	θ	s	Número de Avaliações
<i>PSO</i>	293	313	346,24	1,05	1,15E+04
<i>emCGa</i>	292	313	345,91	-	8,07E+04
<i>ABE</i>	292	313	345,91	-	4,03E+08

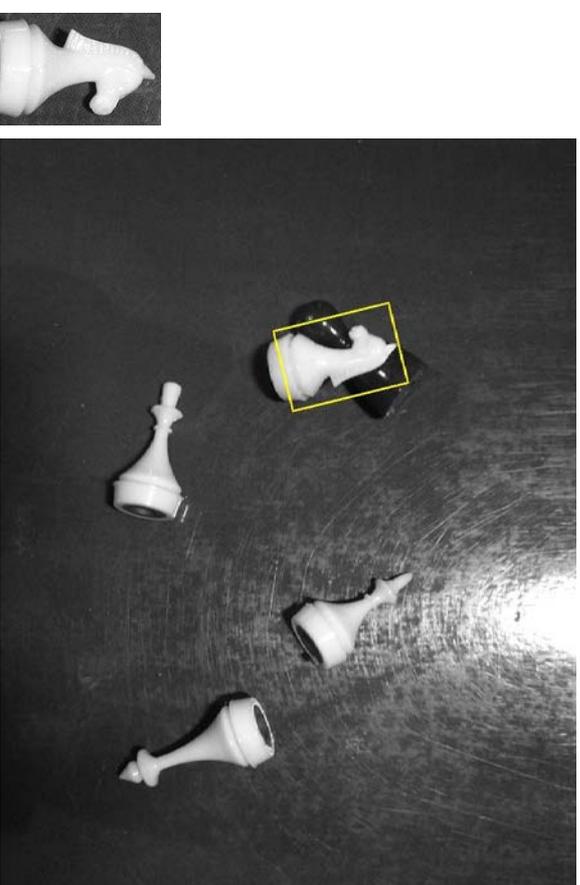


Figura 35: IR (a) e IP (c) utilizada em um dos testes de comparação. O retângulo destaca a solução encontrada pelo PSO.

Tabela 20: Soluções fornecidas pelos métodos para encontrar a IR mostrada na figura 36(a) na IP mostrada na figura 36(b).

Métodos	x	y	θ	s	Número de Avaliações
<i>PSO</i>	792	340	0	1	1,20E+04
<i>emCGA</i>	792	340	0	-	1,08E+05
<i>ABE</i>	792	340	0	-	4,03E+08

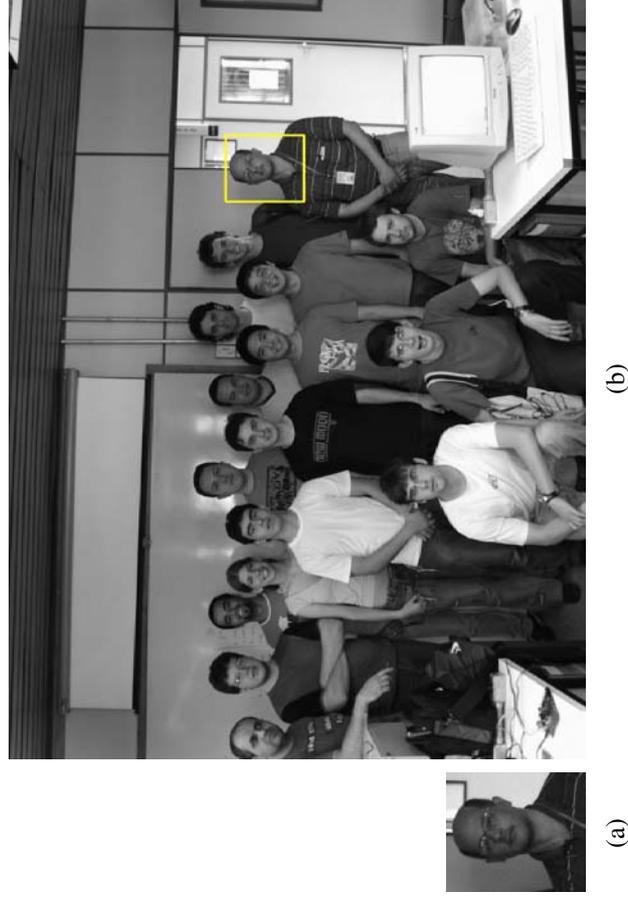


Figura 36: IR (a) e IP (c) utilizada em um dos testes de comparação. O retângulo destaca a solução encontrada pelo PSO.

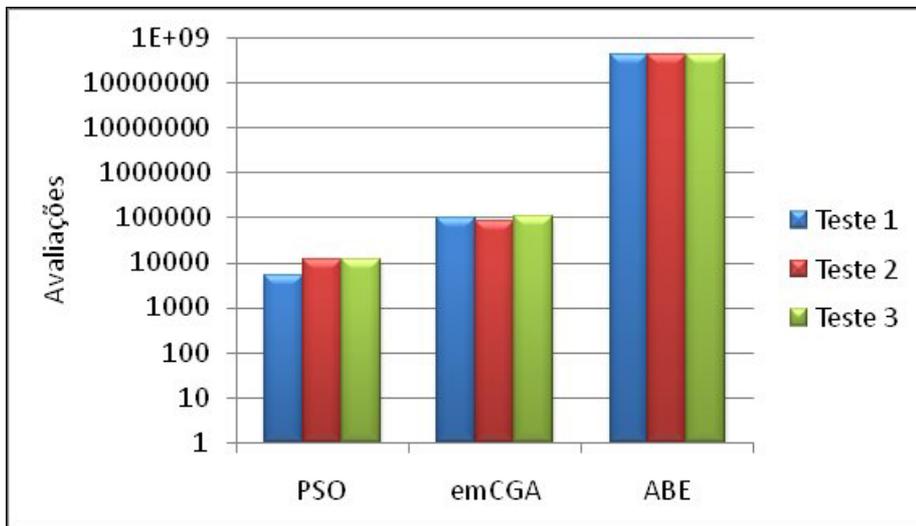


Figura 37: Quantidade de avaliações de *fitness* necessárias para os três métodos determinarem a melhor solução.

CAPÍTULO 5

DISCUSSÃO E CONCLUSÕES

5.1 ANÁLISE DOS RESULTADOS

5.1.1 Experimentos Preliminares

A determinação da eficácia do PSO para solução do problema de reconhecimento de objetos em imagens digitais foi comprovada através da realização de uma série de testes, mostrando que o método é viável para a solução de algumas instâncias do problema. Os resultados dos experimentos da seção 4.1.1 mostraram que o PSO é robusto, mesmo para imagens com ruído.

Na seção 4.1.2 o PSO foi aplicado a uma situação de reconhecimento de objetos muito próxima do real, especialmente no ambiente industrial. O método provou a sua eficiência, fornecendo uma solução de alta qualidade, mesmo sendo submetido a fatores degradantes, como: translação, rotação, redução, oclusão e também variação da perspectiva da IR.

5.1.2 Avaliação do Método

No experimento de sensibilidade à translação, para o objeto da figura 26(d) (pág.50) o método obteve bons resultados, atingindo, na maior parte das vezes, resultados muito próximos do mínimo global. No entanto, em algumas execuções o PSO ficou preso em mínimos locais prejudicando a média dos resultados, que pode ser verificado com base nos valores de desvio padrão. Por este motivo é que juntamente com a média foi apresentada a mediana dos valores dos testes, pois a presença de *outliers* influenciou negativamente a média, o que não acontece com a mediana.

O experimento de sensibilidade à escala e rotação, mostrou a robustez do PSO no reconhecimento de objetos que sofreram transformações como translação, rotação e mudança de escala. A figura 28(b) (pág.53) mostra que, ao diminuir o fator de escala, os objetos tendem a ser confundidos com outros objetos na IP, levando assim a ocorrência de erros no reconhecimento. A explicação para este fato é que a IR, possui dimensões pequenas, e ao ser aplicado um fator de escala menor que 1, as suas dimensões são ainda mais reduzidas, levando o PSO a confundir os objetos. Do contrário, quando o fator de escala é maior que 1, o PSO consegue facilmente encontrar os objetos.

Na segunda parte do experimento foi observado que o erro médio tende a aumentar conforme o objeto é rotacionado. Devido à simetria do objeto da figura 26(d), o método mostrou-se

menos sensível à rotação quando comparado com o objeto da figura 26(e). Mesmo com esta constatação, o método foi capaz de encontrar uma solução muito próxima do mínimo global, o que foi verificado em todos os testes realizados.

A comparação dos gráficos das figura 30(a) (pág.55) e figura 30(b) (pág.55) no experimento de sensibilidade a ruído sugere que o PSO encontrou maior dificuldade para encontrar o objeto da figura 26(d) quando comparado com o objeto da figura 26(e). Possivelmente, isso se deve ao fato de que o primeiro objeto possui uma coloração mais uniforme, o que prejudica o método durante o reconhecimento. Além disso, para os dois objetos, o método mostrou melhor desempenho com o ruído Gaussiano do que com o ruído sal-e-pimenta. A explicação para isso é que o ruído sal-e-pimenta modifica totalmente um pixel, ou seja, ele torna-se branco ou preto. No caso do ruído Gaussiano, o pixel é modificado dentre várias tonalidades, possibilitando assim que a diferença entre a IR e o recorte da IP não seja tão grande.

Em muitos casos de reconhecimento de objetos, podem ocorrer oclusões parciais dos objetos de interesse, tornando a tarefa muito mais difícil. Sendo assim, qualquer método deve ser capaz de tratar também o reconhecimento com possibilidade de oclusão. Os resultados do experimento de sensibilidade à oclusão mostraram que, mesmo com um decréscimo no desempenho do método, o PSO foi capaz de encontrar os dois objetos desejados com acurácia razoável. A mesma observação sobre o problema de ficar preso em um mínimo local que foi feita sobre o experimento de sensibilidade a translação é válida também para este experimento.

A mudança de luminosidade durante a aquisição das IP é um problema comum e difícil de ser controlado em ambientes abertos. Assim, o método deve ser capaz de realizar o reconhecimento mesmo quando mudanças de brilho ou contraste ocorrerem. Os resultados do experimento de sensibilidade à mudança de brilho e contraste demonstraram que o PSO obteve desempenho com erros abaixo de 20%, sugerindo assim que o método conseguiu tratar pequenas mudanças de brilho e contraste.

Os mesmos experimentos feitos com imagens coloridas foram repetidos com imagens em níveis de cinza. Quando é feita a conversão de cores para níveis de cinza, existe uma perda de informação, pois de três canais de informação que existiam sobra apenas um. Este fato aumenta a dificuldade no problema de reconhecimento. Analisando os resultados para a maioria dos experimentos, verifica-se que houve decréscimo no desempenho do PSO. No caso dos experimentos de sensibilidade a ruído e sensibilidade à mudança de brilho ou contraste, pode-se observar que mesmo com a redução de informação, o método obteve melhores resultados com imagens em níveis de cinza do que com imagens coloridas. Desta forma supõe-se que as imagens coloridas são mais degradadas por má iluminação e ruídos, influenciando assim o desempenho do método. Contudo, pode-se concluir que o método proposto pode ser usado tanto em

imagens coloridas como em imagens em níveis de cinza.

5.1.3 Comparação com outra Abordagem Evolucionária

A comparação do PSO com outro método de computação evolucionária de modelagem similar indicou que ambos os métodos conseguiram a mesma qualidade de solução, para os problemas comparados.

Como os métodos de computação evolucionária utilizam uma população de soluções durante o processo de busca, juntamente com o conceito de gerações ou iterações, é ideal que tanto a população quanto a quantidade de gerações necessárias para a obtenção da solução sejam pequenas, pois assim o número de avaliações de *fitness* será baixo.

Analisando a quantidade de avaliações de *fitness* utilizadas pelo PSO para a obtenção da mesma qualidade de resultados que o emCGA necessitou, verificou-se que o PSO precisa, no pior caso, de cerca de 5 vezes menos avaliações de *fitness* do que o emCGA. Aqui o pior caso refere-se à situação na qual o PSO encontre a solução utilizando o máximo de iterações permitidas.

Pelos dados mostrados no gráfico da figura 37 (pág.64) que relatam a quantidade de avaliações de *fitness* no melhor caso, ou seja, na iteração em que o PSO encontrou a melhor solução, é possível verificar que o PSO necessitou de cerca de 18 vezes menos avaliações que o emCGA. Isso mostra que o método proposto aqui é bastante eficaz quando comparado com a outra estratégia evolucionária.

5.2 CONCLUSÕES

O problema de reconhecimento automático de objetos em imagens é constantemente encontrado na indústria, robótica, medicina, engenharia, ciência da computação e outras áreas. Sendo assim, métodos eficientes para tratar deste problema são sempre bem-vindos.

O PSO é um método bastante robusto para tratar de problemas complexos como é o de reconhecimento de objetos. Dentre as vantagens deste método estão a simplicidade e a facilidade de implementação, bem como a necessidade de poucos recursos computacionais.

Por ser um método aproximado, durante a exploração do espaço de busca o método pode ficar preso em ótimos locais que podem estar muito longe do ótimo global. Este fato degrada o desempenho do PSO, pois as soluções providas pelo mesmo podem ser de baixa qualidade em alguns casos.

Outro ponto contrário é a necessidade de que os parâmetros que compõem o modelo sejam ajustados de maneira muito precisa para cada classe de problema, o que demanda uma grande quantidade de tempo. O ideal, como já relatado nas seções anteriores, é que o método

possa ajustar seus parâmetros de forma automática e dinâmica, o que, provavelmente forneceria ao método a possibilidade de aumentar em muito o seu desempenho.

O método de reconhecimento proposto aqui possui vantagem sobre outros métodos, pois utiliza apenas um exemplo do objeto que se deseja encontrar em uma imagem de paisagem, permitindo assim que seja realizado reconhecimento genérico de objetos. Outro ponto forte do método é o de realizar o reconhecimento sem a necessidade de pré-processar e segmentar a imagem de paisagem, ou seja, da forma que ela foi adquirida esta pode ser apresentada ao método.

Pesando as vantagens e as desvantagens, e também observando a vasta literatura existente sobre a aplicação do PSO e de outros métodos de computação evolucionária, como solução para diversos problemas, pode-se concluir que o método proposto aqui é plausível de ser aplicado como solução para o problema de reconhecimento de objetos em imagens.

5.3 TRABALHOS FUTUROS

O trabalho realizado aqui atingiu bons resultados, mas algumas falhas foram identificadas. Um ponto a ser melhorado é a realização do reconhecimento em imagens onde o objeto a ser detectado está ocluso, pois o método só obteve sucesso com apenas 25% de oclusão.

Como mencionado, a determinação de um conjunto de parâmetros ideal é complexa e demanda grande esforço computacional. Assim, é interessante desenvolvimento de uma metodologia de ajuste de parâmetros de forma dinâmica, ou seja, durante o processo de busca o método realiza o auto-ajuste de seus parâmetros.

O tempo necessário para que o PSO realizasse o reconhecimento foi elevado para algumas aplicações. Desta forma, outro ponto a ser melhorado é a redução do tempo de resposta do método. Uma possibilidade para esta redução é a utilização de técnicas como *GPUProgramming*.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABRAHAM, A.; GUO, H.; LIU, H. Swarm intelligence: foundations, perspectives and applications. In: NEDJAH, N.; MOURELLE, L. (Ed.). *Swarm Intelligent Systems*. Germany: Springer-Verlag, 2006. v. 26/2006, p. 3–25.
- ACHARYA, T.; RAY, A. K. *Image Processing: Principles and Applications*. Hoboken, N.J.: John Wiley & Sons, 2005. ISBN 0471719986.
- AMIT, Y. *2D Object Detection and Recognition: Models, Algorithms, and Networks*. Cambridge, MA, USA: MIT Press, 2002. ISBN 0262011948.
- BA-KARAIT, N. S.; SHAMSUDDIN, S. Handwritten digits recognition using particle swarm optimization. In: *Proceedings of 2nd Asia International Conference on Modeling & Simulation*. Los Alamitos, CA, USA: IEEE, 2008. p. 615–619.
- BANKS, A.; VINCENT, J.; ANYAKOHA, C. A review of particle swarm optimization. Part I: background and development. *Natural Computing: An International Journal*, Kluwer Academic Publishers, Hingham, MA, USA, v. 6, n. 4, p. 467–484, 2007. ISSN 1567-7818.
- BERGH, F. van den; ENGELBRECHT, A. P. A study of particle swarm optimization particle trajectories. *Information Sciences*, v. 176, n. 8, p. 937–971, April 2006.
- BONCELET, C. Image noise models. In: BOVIK, A. C. (Ed.). *Handbook of Image and Video Processing*. Philadelphia, PA, USA: Elsevier, 2005.
- CASTRO, L. N. Immune, swarm, and evolutionary algorithms. Part I: basic models. In: *Proceedings of the 9th International Conference on Neural Information Processing*. Singapore: IEEE, 2002. p. 1464–1468. ISBN 981-04-7524-1.
- CHEN, Z. Learning from nature: Natural computing meets virtual learning. In: *International Journal of Computers, Communications & Control*. Oradea, Romania: Agora University Editing House - CCC Publications, 2007. v. 3, n. 1, p. 33–40.
- CLERC, M.; KENNEDY, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 1, p. 58–73, 2002.
- DARWIN, C. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. London: John Murray, Albemarle Street, 1859.
- EBERHART; SHI, Y. Particle swarm optimization: developments, applications and resources. In: *Proceedings of the Congress on Evolutionary Computation*. Seoul, Korea: IEEE, 2001. v. 1, p. 81–86.
- EBERHART, R. C.; SHI, Y. Comparison between genetic algorithms and particle swarm optimization. In: *Proceedings of the 7th International Conference on Evolutionary Programming VII*. London, UK: Springer-Verlag, 1998. p. 611–616. ISBN 3-540-64891-7.

- EBERHART, R. C.; SHI, Y.; KENNEDY, J. *Swarm Intelligence*. San Francisco: Morgan Kaufmann, 2001. ISBN 1558605959.
- FELISBERTO, M. K.; SILVÉRIO, H. S.; CENTENO, T. M.; ARRUDA, L. V. R. An object detection and recognition system for weld bead extraction from digital radiographs. *Computer Vision and Image Understanding*, Elsevier Science Inc., New York, NY, USA, v. 102, n. 3, p. 238–249, 2006. ISSN 1077-3142.
- FONSECA, C. M.; FLEMING, P. J. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, v. 3, p. 1–16, 1995.
- FORSYTH, D. A.; PONCE, J. *Computer Vision: A Modern Approach*. New Jersey: Prentice Hall, 2002. ISBN 0130851981.
- GARNETT, R.; HUEGERICH, T.; CHUI, C.; WENJIE, H. A universal noise removal algorithm with an impulse detector. In: *IEEE Transactions on Image Processing*. Piscataway, NJ: IEEE Signal Processing Society, 2005. v. 14, p. 1747–1754.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento de Imagens Digitais*. São Paulo: Editora Edgard Blucher, 2000.
- HAUPT, R. L.; HAUPT, S. E. *Practical Genetic Algorithms (2nd Edition)*. Hoboken, NJ: John Wiley & Sons, 2004. ISBN 978-0-471-45565-3.
- HEMBECKER, F.; LOPES, H. S.; JUNIOR., W. G. Particle swarm optimization for the multi-dimensional knapsack problem. In: *Adaptive and Natural Computing Algorithms*. Germany: Springer Berlin / Heidelberg, 2007. v. 4431, p. 358–365.
- HOOS, H. H.; STÜTZLE, T. *Stochastic Local Search: Foundations and Applications*. San Francisco: Morgan Kaufmann, 2005. ISSN 1-55860-872-9.
- JANG, J.-S.; HAN, K.-H.; KIM, J.-H. Face detection using quantum-inspired evolutionary algorithm. In: *Proceedings of the Congress on Evolutionary Computation*. Piscataway, NJ: IEEE, 2004. v. 2, p. 2100–2106.
- JIAO, J.; YE, Q.; HUANG, Q. A configurable method for multi-style license plate recognition. *Journal of Pattern Recognition*, Elsevier Science Inc., New York, NY, USA, v. 42, n. 3, p. 358–369, 2009. ISSN 0031-3203.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*. Piscataway, NJ: IEEE, 1995. p. 1942–1948 vol.4.
- KOBAYASHI, T.; NAKAGAWA, K.; IMAE, J.; ZHAI, G. Real time object tracking on video image sequence using particle swarm optimization. In: *Proceedings of the International Conference on Control, Automation and Systems*. Seoul: IEEE, 2007. p. 1773–1778.
- KROTOSKY, S. J.; TRIVEDI, M. M. Mutual information based registration of multimodal stereo videos for person tracking. *Comput. Vis. Image Underst.*, Elsevier Science Inc., New York, NY, USA, v. 106, n. 2-3, p. 270–287, 2007. ISSN 1077-3142.

- LAIKA, A.; STECHELE, W. A review of different object recognition methods for the application in driver assistance systems. In: *Proc. Eight International Workshop on Image Analysis for Multimedia Interactive Services*. Washington, DC, USA: IEEE Computer Society, 2007. p. 10. ISBN 0-7695-2818-X.
- MCHUGH, S. *Cambridge in Colour*. 2009. Disponível em: <<http://www.cambridgeincolour.com/tutorials/histograms1.htm>>. Acesso em: Março de 2009.
- NENE, S. A.; NAYAR, S. K.; MURASE, H. *Columbia object image library (COIL-100)*. 2009. Disponível em: <<http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php>>. Acesso em: Março de 2009.
- NIEVERGELT, J. Exhaustive search, combinatorial optimization and enumeration: Exploring the potential of raw computing power. In: *Proc. 27th Conference on Current Trends in Theory and Practice of Informatics*. London: Springer-Verlag, 2000. p. 18–35. ISBN 3-540-41348-0.
- OLMOS, A.; KINGDOM, F. *McGill Calibrated Colour Image Database*. Disponível em: <<http://tabby.vision.mcgill.ca>>. Acesso em: Março de 2009.
- OMRAN, M. G. H. *Particle Swarm Optimization Methods for Pattern Recognition and Image Processing*. Tese (Doutorado) — University of Pretoria, 2004.
- OWECHKO, Y.; MEDASANI, S. Cognitive swarms for rapid detection of objects and associations in visual imagery. In: *Proceedings of IEEE Swarm Intelligence Symposium*. Piscataway, NJ: IEEE, 2005. p. 420–423.
- PARKS, D. H.; LEVINE, M. D. The McGill object detection suite. In: *Proceedings of the The 3rd Canadian Conference on Computer and Robot Vision*. Washington, DC, USA: IEEE Computer Society, 2006. p. 47. ISBN 0-7695-2542-3.
- PAUN, G. Bio-inspired computing paradigms (natural computing). In: *Unconventional Programming Paradigms*. Germany: Springer Berlin / Heidelberg, 2005, (Lecture Notes in Computer Science, Volume 3566/2005). p. 155–160.
- PEREZ, C.; VALLEJOS, J. Face detection using PSO template selection. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*. Taipei, Taiwan: IEEE Systems, Man and Cybernetics Society, 2006. v. 5, p. 4220–4224.
- PERLIN, H. A.; LOPES, H.; CENTENO, T. Particle swarm optimization for object recognition in computer vision. In: *New Frontiers in Applied Artificial Intelligence*. Germany: Springer Berlin / Heidelberg, 2008. (Lecture Notes in Computer Science, v. 5027), p. 11–21. ISSN 0302-9743.
- PINTER, J. D. Global optimization: Software, test problems, and applications. In: PARDALOS, P.; ROMEIJN, H. (Ed.). *Handbook of Global Optimization*. Germany: Springer Berlin / Heidelberg, 2002. v. 2, cap. 15, p. 515–569.
- PLUIM, J. P. W.; MAINTZ, J. B. A.; VIERGEVER, M. A. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, v. 22, n. 8, p. 986–1004, 2003.

- POLI, R. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, Hindawi Publishing Corp., New York, NY, United States, v. 2008, n. 1, p. 1–10, 2008. ISSN 1687-6229.
- POLI, R.; KENNEDY, J.; BLACKWELL, T. Particle swarm optimization. *Swarm Intelligence*, v. 1, n. 1, p. 33–57, June 2007.
- RAIDL, G. R. Evolutionary computation: An overview and recent trends. In: *ÖGAI Journal*. Austria: Austrian Society for Artificial Intelligence, 2005. v. 24, p. 2–7.
- RUSSELL, R.; SINHA, P. Perceptually-based comparison of image similarity metrics. In: *MIT AIM*. Cambridge, MA, USA: MIT Press, 2001.
- SAISAN, P.; MEDASANI, S.; OWECHKO, Y. Multi-view classifier swarms for pedestrian detection and tracking. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2005. p. 18. ISBN 0-7695-2372-2-3.
- SHANNON, C. E. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM, New York, NY, USA, v. 5, n. 1, p. 3–55, 2001. ISSN 1559-1662.
- SILVA, R. R.; LOPES, H. S.; LIMA, C. R. E. A compact genetic algorithm with elitism and mutation applied to image recognition. In: *Proceedings of the 4th international conference on Intelligent Computing*. Berlin, Heidelberg: Springer-Verlag, 2008. p. 1109–1116. ISBN 978-3-540-85983-3.
- TSANG, E. *Foundations of Constraint Satisfaction*. New York: Academic Press, 1993.
- VERNON, D. *Machine Vision: Automated Visual Inspection and Robot Vision*. New York: Prentice Hall, 1991. ISBN 0-13-543398-3.
- VIOLA, P.; WELLS III, W. M. Alignment by maximization of mutual information. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 24, n. 2, p. 137–154, 1997. ISSN 0920-5691.
- WANG, L.; SHI, J.; SONG, G.; SHEN, I.-F. Object detection combining recognition and segmentation. In: *Computer Vision*. Germany: Springer Berlin / Heidelberg, 2007. p. 189–199.
- WILKE, D. N. *Analysis of the particle swarm optimization algorithm*. Dissertação (Mestrado) — Department of Mechanical and Aeronautical Engineering, University of Pretoria, 2005.
- XIANG, L.; SHUQIAN, L.; KUNCHENG, L. Registration of CT and MR images using maximization of mutual information. In: *Proceedings of the 1st Joint Engineering in Medicine and Biology, 21st Annual Conference and the 1999 Annual Fall Meeting of the Biomedical Engineering Society*. Atlanta, GA, USA: IEEE, 1999. v. 2, p. 1164.
- XIE, X.-F.; ZHANG, W.-J.; YANG, Z.-L. Hybrid particle swarm optimizer with mass extinction. In: *International Conference on Communications, Circuits and Systems and West Sino Expositions*. Chengdu, China: IEEE, 2002. v. 2, p. 1170–1173. ISBN 0-7803-7547-5.

ZHANG, X.; HU, W.; MAYBANK, S.; LI, X.; ZHU, M. Sequential particle swarm optimization for visual tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage, Alaska, US: IEEE, 2008. p. 1–8. ISSN 1063-6919.

ZIMMERMANN, K. *Licence plate and text detection*. 2009. Disponível em: <<http://cmp.felk.cvut.cz/~zimmerk/lpd/index.html>>. Acesso em: Março de 2009.

ZUECH, N. *Understanding and Applying Machine Vision*. New York: CRC Press, 2000. ISBN -13: 9780824789299.

RESUMO:

O reconhecimento automático de objetos tem como objetivo fazer com que dispositivos computacionais tenham a capacidade de extração de informação visual do ambiente onde estão inseridos. Uma das formas de tratar o problema de reconhecimento é através da técnica de *Template Matching* visualizada como um problema de otimização global. Neste caso se deseja determinar a posição de um objeto em uma imagem, bem como seu ângulo de rotação e qual o fator de escala utilizado. A otimização por enxame de partículas (*Particle Swarm Optimization - PSO*) é um método bastante difundido na literatura e amplamente empregado em problemas de otimização global e foi utilizado como parte da solução. As medidas de similaridade são de grande importância para o funcionamento do método, assim foram realizados estudos para a escolha de uma boa função. Também foram realizados experimentos para determinar um conjunto de parâmetros adequados para o método de otimização, com o objetivo de melhorar a qualidade das soluções fornecidas. Para determinar a sua eficácia, o método foi testado com imagens coloridas e em níveis de cinza, com translação, rotação, mudança de escala e oclusão parcial dos objetos, bem como presença de ruído e simulação de mudança de luminosidade. Ainda foi feita uma comparação com outra abordagem evolucionária para o reconhecimento de objetos, a fim de verificar o desempenho do método proposto. Os resultados mostraram que o PSO foi capaz de encontrar os objetos predefinidos com bom desempenho, sugerindo que o método proposto é bastante promissor, especialmente no que diz respeito a robustez e simplicidade. Sendo assim, esta solução pode ser uma opção a ser empregada no reconhecimento de objetos em aplicações reais de visão computacional.

PALAVRAS-CHAVE

Visão Computacional, Reconhecimento de Objetos, Otimização, Particle Swarm Optimization..

ÁREA/SUB-ÁREA DE CONHECIMENTO

1.03.03.04 – 9 Sistemas de Informação

1.03.03.05 – 7 Processamento Gráfico

2009

Nº: 501

Livros Grátis

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

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

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

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

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