UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Atenção Visual Bottom-up Guiada por Otimização via Algoritmos Genéticos

Eanes Torres Pereira

Campina Grande, Paraíba, Brasil
Março de 2007

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Atenção Visual Bottom-up Guiada por Otimização via Algoritmos Genéticos

Eanes Torres Pereira

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande – Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação (MSc).

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Modelos Computacionais e Cognitivos

Herman Martins Gomes
Orientador

Campina Grande, Paraíba, Brasil
Março de 2007

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

P436 Pereira, Eanes Torres

2007 Atenção visual bottom-up guiada por otimização via algoritmos genéticos/
Eanes Torres Pereira. – Campina Grande, 2007
118fs.: il.

Referências

Dissertação (Mestrado em Ciência da Computação) -

Universidade Federal de Campina Grande,

Centro de Engenharia Elétrica e Informática.

Orientador: Herman Martins Gomes.

1– Visão Computacional 2– Atenção Visual Bottom-up 3–

Algoritmos Genéticos 4– Otimização 5– Grades Computacionais I–

Título CDU 004.932

Dissertação de Mestrado sob o título "*Atenção Visual Bottom-up Guiada por Otimização via Algoritmos Genéticos*", defendida por Eanes Torres Pereira e aprovada em Março de 2007, em Campina Grande, Estado da Paraíba, pela banca examinadora constituída pelos doutores:

Prof. Ph.D. Herman Martins Gomes

DSC / CEEI / UFCG

Orientador

Prof. Ph.D. Francisco Vilar Brasileiro

DSC / CEEI / UFCG

Examinador

Prof. Ph.D. Edson Costa de Barros Carvalho Filho

CIN / UFPE

Examinador

Resumo

A atenção visual é um mecanismo biologicamente inspirado, o qual corresponde à habilidade de selecionar e processar somente as regiões mais relevantes de uma cena visual. Para fins didáticos, a atenção visual pode ser dividida em duas categorias principais: bottom-up e top-down. A atenção visual bottom-up guia o foco de atenção a partir de características primitivas (como descontinuidades de intensidade em diferentes escalas e orientações) computadas diretamente na imagem, sem qualquer informação contextual. A atenção visual top-down, por outro lado, realiza uma busca por regiões de interesse a partir de características de alto nível, especificadas na forma de conhecimento prévio na forma ou modelos sobre o que se está buscando na cena. A principal questão de pesquisa que procuramos responder nessa dissertação é a seguinte: como seria possível agregar algum comportamento de alto nível a um mecanismo típico de atenção visual bottom-up (guiando dessa forma o foco de atenção para classes de regiões pré-estabelecidas)? O modelo mais conhecido de atenção visual bottom-up utiliza vários mapas de características primitivas para formar um mapa de saliência, o qual indica a importância do ponto de vista atencional das diferentes regiões de uma cena. Nesse trabalho, atribuímos pesos aos mapas de características e desenvolvemos um processo de otimização baseado em algoritmos genéticos simulados em uma grade computacional. Foram realizados experimentos com quatro classes de objetos (carros, faces de pessoas, objetos genéricos e pistolas). Os resultados utilizando atenção bottom-up com otimização foram comparados com os resultados de um mecanismo sem otimização de pesos e com um sistema existente que implementa o difundido modelo de atenção visual proposto por Itti et al. [Itti et al., 1998]. Os resultados mostraram ganhos de até 30% utilizando-se a abordagem proposta. Desta forma, este trabalho mostra que a atenção visual pode ser guiada para regiões pré-definidas, podendo ser utilizada como parte de sistemas de detecção de objetos.

Abstract

Visual attention is a biologically inspired mechanism, which corresponds to the ability of selecting and processing only the most relevant regions of a visual scene. For didactic purposes, visual attention can be divided into two main categories: bottom-up and top-down. Bottom-up visual attention guides the attention focus by using primitive visual features (such as discontinuities in intensity across different scales and orientations) computed directly from the input image, without the need of any context information. Top-down visual attention, on the other side, performs a search for interest regions from higher-level features, specified in the form of previous knowledge or models about what is being sought in the scene. The main research question that we intended to answer in this dissertation was the following: how it would be possible to incorporate some higher-level behaviour into a typical bottom-up visual attention mechanism (thus guiding the attention focus to pre-established classes of objects)? The most known bottom-up visual attention model uses several primitive feature maps to form a saliency map, which indicates the importance of the different scene regions. In this work, we assigned weights to the feature maps and developed an optimization process based on genetic algorithms running on a computational grid. Experiments involving four object classes (cars, human faces, generic objects and pistols) have been performed. The results of the optimized bottom-up mechanism have been compared with the results of a mechanism not using optimized weights and with an existing system that implemented the well known visual attention mechanism proposed by Itti et al. [Itti et al., 1998]. The results have shown an improvement of up to 30% when using the optimized mechanism. Thus, this work shows that visual attention can indeed be guided towards pre-defined regions and can be used as part of object detection systems.

Agradecimentos

Aos meus companheiros de laboratório (Bruno, Claudio, Eduardo, Einstein, Felipe, Luciana, Luana, Rodrigo, Thiago, Vinicius e Walter) pelas inúmeras discussões, filosóficas ou não, travadas durante nosso período de convivência. Agradeço também a vocês, companheiros de laboratório, pelas inúmeras vezes que viram o que eu não estava vendo (mesmo que tenha sido um ponteiro apontando para o nada e gerando falha de segmentação) e me mostraram o caminho do *delete* após o *new*.

Aos amigos alagoanos, pelas longas sessões de cinema (cine flamingo), pizza, batata e *counter-strike* que nos fizeram esquecer ao menos por algumas horas de todo o trabalho que tínhamos a fazer. Agradeço, também, a paciência que Fred, Milena e Xambinho tiveram comigo, principalmente durante o primeiro ano de mestrado. Em especial àqueles que me guiaram ao *TAO*, Xambinho (por meio de sua computação quântica) e Elthon (por meio de seus filmes e documentários intrigantes).

Ao professor Herman, pela orientação e acompanhamento constantes que foram fundamentais para a realização deste trabalho.

A Aninha e Vera por sempre estarem prontas a servir.

À Ludmila por ter me ajudado na finalização deste trabalho, desempenhando praticamente o papel de *coach*, oferecendo suas mãos quando não pude usar as minhas e, além disso, por trazer um pouco mais de sentido à minha existência.

Aos meus pais, que sempre me apoiaram em todos os meus empreendimentos, mesmo que em alguns momentos não entendessem o porquê de eu seguir determinados caminhos.

A Deus, Alá, Javé, Jeová, enfim, à *Força* que criou e domina o universo, por, apesar de me ter imposto a existência, ter me permitido esta vida maravilhosa e a realização deste trabalho.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Nesta dissertação, é investigado o uso de otimização via algoritmos genéticos para guiar um mecanismo de atenção visual *bottom-up* para regiões contendo objetos ou regiões de imagens com características pré-definidas. A otimização objetiva agregar conhecimento de alto nível a um mecanismo que utiliza apenas características primitivas, como é o caso da atenção visual *bottom-up*. Vários experimentos foram realizados visando comparar, identificar semelhanças e qualidades em relação a um sistema de atenção visual *bottom-up* amplamente utilizado. A seguir, a motivação para o desenvolvimento desse trabalho é apresentada, as principais características e limitações nas soluções existentes são descritos e os principais objetivos desta pesquisa são apresentados. O capítulo é concluído com uma breve descrição da estrutura da dissertação.

1.1 Motivação

A curiosidade é uma característica inerente ao ser humano. Tal característica tem impulsionado o desenvolvimento científico desde os primórdios da humanidade. A Ciência tem tentado explicar e entender os fenômenos que ocorrem na natureza. Entender o funcionamento do corpo humano é obviamente parte integrante desta constante busca da Ciência. Vários ramos científicos foram criados objetivando o estudo minucioso do corpo humano. Dentre eles, pode-se citar: a Anatomia, a Psicologia e a Fisiologia.

Porém, somente entender e explicar o funcionamento do próprio corpo não é o bastante para as mentes ávidas por conhecimento. Ao longo do tempo, a Ciência criou novos ramos

de estudo que envolvem a simulação de processos que ocorrem no corpo humano e buscam também criar modelos ou máquinas que simulem determinadas características e comportamentos humanos. Algumas áreas têm se destacado em tal empreendimento, como é o caso da Inteligência Artificial. A Inteligência Artificial consiste de esforços intelectuais e tecnológicos relacionados à construção de máquinas inteligentes, à formalização do conhecimento, à mecanização do raciocínio, e ao uso de modelos computacionais para compreender a Psicologia e o comportamento de pessoas e animais [Doyle and Dean, 1996].

Várias áreas da Ciência da Computação utilizam conhecimentos da Inteligência Artificial com o intuito de automatizar processos. É o que ocorre com a Visão Computacional, área na qual este trabalho se enquadra. A Visão Computacional tem como objetivo interpretação automática de cenas complexas [Jain and Dorai, 1997]. Alguns dos principais problemas da Visão Computacional são o reconhecimento e a aprendizagem de modelos visuais.

Um problema recorrente quando se deseja fazer reconhecimento ou aprendizagem de modelos visuais é a dificuldade de se encontrar um técnica robusta capaz de extrair regiões contendo objetos de imagens genéricas. A forma mais primitiva de se realizar a extração de tais regiões seria uma busca *pixel* a *pixel* na imagem. Porém, vários sistemas têm sido desenvolvidos utilizando atenção visual para agilizar o processo de busca por regiões importantes nas imagens.

A atenção visual é a habilidade que o sistema visual dos vertebrados superiores utiliza para selecionar e processar somente as regiões mais relevantes em uma cena visual. A atenção visual pode ser entendida como um mecanismo para lidar com a incapacidade de tratar de uma só vez uma grande quantidade de informação visual tanto em sistemas biológicos quanto em sistemas computacionais. Deste modo, somente as regiões mais importantes numa cena são escolhidas para processamento [Fischer and Weber, 1993]. Esta seleção das informações mais relevantes dos estímulos de entrada é uma das características mais importantes dos sistemas visuais biológicos que permite rápida detecção de predadores, perpetuação e evolução das espécies [Itti and Koch, 2001a].

Há dois métodos principais para obtenção da Atenção Visual. Os métodos *top-down* e *bottom-up*. O método *top-down* usa conhecimentos obtidos a priori para detectar regiões de maior interesse numa imagem. Esses conhecimentos podem ser obtidos de várias formas. Geralmente, utilizam-se ferramentas de aprendizagem baseadas em modelos estatís-

ticos como, por exemplo: redes neurais e máquinas de vetores de suporte. Porém, esses conhecimentos também podem ser fornecidos por um ser humano, selecionando-se manualmente regiões de maior interesse numa imagem. A atenção visual *bottom-up* é guiada por características primitivas da imagem como cor, intensidade e orientação. Além disso, ela atua de modo inconsciente, ou seja, o observador é levado a fixar sua atenção em determinadas regiões da imagem devido aos estímulos causados pelos contrastes entre características visuais presentes na imagem.

O sistema de atenção visual *bottom-up* proposto por Itti et al. [Itti et al., 1998] é o mais conhecido e utilizado atualmente para seleção de regiões salientes em imagens. No entanto, uma característica inerente a sistemas *bottom-up* é o fato de tais sistemas identificarem regiões importantes em áreas da imagem que não necessariamente contêm objetos bem definidos. Isto ocorre devido às características de tais regiões se sobressaírem em relação as suas vizinhas independentemente de tais regiões conterem objetos ou não. Vários experimentos têm demonstrado que a atenção visual pode ser guiada pelas características tidas como mais importantes para selecionar determinadas regiões [Wolfe, 2000]. A Figura 1.1 ilustra uma região que possui um alto valor de saliência, porém não contém nenhum objeto específico. O ponto saliente dessa figura é um resultado artificial e foi obtido pela aplicação do sistema de Itti et al. [Itti et al., 1998]. O ponto central da região circular indica o ponto mais saliente da imagem, deve-se ressaltar que, por se tratar de atenção visual *bottom-up*, esta *saliência* é inconsciente, sendo guiada apenas por características primitivas da imagem. Neste caso, a grande intensidade da iluminação na região destacada explica seu alto valor de saliência.

Figura 1.1: Exemplo de ponto de atenção em região genérica.

É com o intuito de guiar a atenção para regiões que contenham objetos de interesse que este trabalho propõe um mecanismo que otimiza pesos utilizando algoritmos genéticos, tais pesos são atribuídos aos diversos mapas utilizados para formar o mapa de saliências. Este mecanismo é descrito no capítulo 4.

1.2 Descrição do Problema

O problema que se pretende resolver é definido a seguir: dada uma imagem contendo, ou não, *background* complexo, como guiar um mecanismo de atenção visual *bottom-up* para que ele se atenha a regiões contendo objetos de interesse e não seja desviado para regiões que não contenham tais tipos de objetos? Este problema pode ser decomposto em três partes:

- Escolha de um método de ponderação que atribua pesos aos mapas que formam o mapa de saliências;
- Determinação de um mecanismo para otimizar os pesos de forma a ressaltar características específicas dos objetos que se deseja selecionar como mais salientes;
- Identificação de um meio eficaz para otimizar os pesos.

A seção a seguir, expõe os objetivos deste trabalho, bem como ressalta a relevância do mesmo. Além disso, descreve o que se propõe para resolver o problema exposto nesta seção.

1.3 Objetivos

O objetivo deste trabalho é desenvolver um sistema de atenção visual *bottom-up* que possa ser guiado para identificar regiões salientes em imagens de acordo com as preferências do usuário. Por exemplo, se o usuário do sistema deseja que em imagens contendo pessoas apenas as regiões das faces sejam ressaltadas como mais salientes, ele deve utilizar um conjunto de pesos que tenha sido previamente otimizado para ressaltar regiões de faces de pessoas em imagens. Um exemplo desse tipo de região saliente é o apresentado na Figura 1.2.

A seguir são apresentados os objetivos específicos nos quais este trabalho foi dividido:

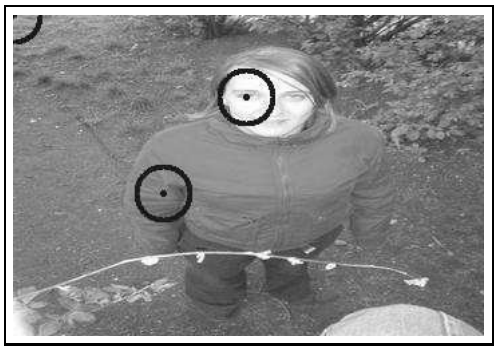


Figura 1.2: Exemplo de ponto de atenção em região específica (face).

- **Desenvolvimento de um módulo de atenção visual *bottom-up* -** Mecanismo de atenção visual que guia o processamento de forma que o sistema selecione apenas as principais regiões da cena. Neste objetivo se enquadra, também, o estudo de mecanismos de atenção visual existentes.

- **Desenvolvimento de um módulo de algoritmos genéticos -** Estudar algoritmos genéticos e selecionar uma biblioteca para implementação dos mesmos. Além disso, este objetivo busca criar o mecanismo de geração e otimização de pesos para os mapas de características e conspicuidades. O objetivo da otimização dos pesos dos mapas é determinar previamente que tipo de regiões deseja-se que sejam detectadas como mais salientes, guiando-se, desta forma, a busca por regiões.

- **Realização de experimentos -** Após a implementação do sistema, foram realizados diversos experimentos para validar e verificar as melhorias adquiridas pelo incremento das características aqui expostas à forma convencional de se determinar regiões salientes em imagens. Além disso, os experimentos objetivaram realizar um estudo comparativo do sistema proposto com o sistema de Itti et al. [Itti et al., 1998].

1.4 Relevância

Há vários trabalhos [Navalpakkam and Itti, 2002; Navalpakkam and Itti, 2003; Navalpakkam and Itti, 2006; Sun et al., 2003] que propõem métodos para integrar conhecimentos de alto nível a sistemas de atenção visual *bottom-up*, porém todos utilizam informação estatística ou conhecimento estruturado (como ontologias e grafos) para adicionar conhecimento de alto

nível à atenção visual *bottom-up*. Isto pode acarretar no uso de soluções que são, na verdade, locais ao problema tratado. Para evitar a parada em mínimos ou máximos locais, geralmente são utilizados métodos como algoritmos genéticos e *simulated annealing* (têmpera simulada). Como o uso de algoritmos genéticos como meio de atribuir informação de alto nível a sistemas de atenção visual *bottom-up* ainda não foi bem investigado pela literatura especializada, este trabalho se propõe a analisar a viabilidade do uso de algoritmos genéticos para resolver este problema.

O sistema proposto pode ser utilizado como um módulo em sistemas de detecção ou reconhecimento [Rodrigues,], podendo ser otimizado para guiar a atenção para determinadas classes de objetos. Ele pode servir como meio para agilizar a localização dos objetos mais importantes da cena. Como exemplos de aplicações práticas do sistema temos: filtragem web [Fong. and Hui, 2001] e segurança de ambientes [Lopez et al., 2006]. No primeiro caso, o sistema funcionaria acoplado a um navegador web e filtraria páginas que contivessem imagens com determinados tipos de objetos. Por exemplo, poderia-se evitar que o navegador mostrasse páginas que contivessem imagens de armas. No segundo caso, o sistema poderia ser integrado à rede de câmeras de segurança de algum estabelecimento comercial e ao sinal (emitido por um segurança) de algum indivíduo suspeito carregando um objeto estranho o sistema poderia rastrear as imagens das câmeras em busca do objeto e conseqüentemente do indivíduo.

A detecção de assunto em fotografia também é uma aplicação na qual o sistema proposto pode ser utilizado. Inclusive, este sistema foi desenvolvido com o intuito de melhorar o desempenho de um sistema de detecção de assunto implementado em um projeto de pesquisa do qual o autor participa. Neste sistema de detecção de assunto, a atenção visual foi otimizada para ser guiada para regiões contendo faces. Desta forma, o sistema de atenção visual serve como facilitador para um detector de faces.

Este trabalho se propõe a atribuir conhecimento de alto nível a um mecanismo de atenção visual que utiliza características primitivas (cor, intensidade e orientação). Este conhecimento de alto nível é atribuido utilizando-se algoritmos genéticos e seleção de regiões salientes por seres humanos na etapa de otimização dos pesos. A otimização dos mapas de características possibilitará que a atenção seja guiada para regiões contendo objetos específicos definidos pelo usuário. Na próxima seção, descrevemos como a apresentação das

atividades associadas à consecução dos objetivos do trabalho foram organizadas nos capítulos da dissertação.

1.5 Estrutura da Dissertação

Esta dissertação é dividida em seis capítulos. No Capítulo 2, são descritos os principais conceitos envolvidos nesta dissertação: atenção visual *bottom-up* e algoritmos genéticos. O modelo de atenção visual *bottom-up* descrito é o que utiliza mapas de saliências para identificar as regiões mais importantes de uma imagem e é baseado no modelo proposto por Itti et al. [Itti et al., 1998]. O modelo de algoritmo genético descrito é o Algoritmo Genético Canônico [DeJong, 1975].

O Capítulo 3 contém uma revisão bibliográfica sobre métodos que inserem conhecimentos de alto nível em sistemas de atenção visual *bottom-up* e métodos que utilizam algoritmos genéticos para aumentar o desempenho de sistemas de detecção de objetos em imagens. São descritos cinco trabalhos, três sobre utilização de conhecimentos de alto nível em sistemas de atenção visual *bottom-up* e dois sobre o uso de algoritmos genéticos em sistemas de detecção de objetos.

A arquitetura do sistema é descrita no Capítulo 4. Esta arquitetura é composta por três módulos: otimização de pesos, atenção visual e verificação de regiões. O módulo de otimização de pesos é responsável por otimizar os pesos que são utilizados para ponderar os mapas de características e saliências. O módulo de atenção visual detecta as regiões mais salientes de imagens de acordo com os obtidos obtidos pelo módulo de otimização. O módulo de verificação calcula estatísticas com base nos pontos obtidos pelo módulo de atenção visual e em regiões de imagens selecionadas manualmente.

Os experimentos e seus resultados são apresentados e analisados no Capítulo 5. Foram realizados experimentos nos quais os objetos de interesse eram: faces de pessoas, objetos genéricos, automóveis e pistolas ou revólveres. Além disso, foram realizados experimentos comparativos com a implementação de Itti et al. [Itti et al., 1998].

O Capítulo 6 conclui a dissertação apresentando um resumo dos principais pontos estudados, as contribuições da pesquisa desenvolvida e algumas sugestões de trabalhos futuros.

Capítulo 2

Fundamentos de Atenção Visual e Algoritmos Genéticos

Neste capítulo descrevemos o que é atenção visual, apresentamos o modelo de atenção visual *bottom-up* mais conhecido e utilizado, além de algumas técnicas de combinação dos mapas que formam os mapas de saliências. Além disso, como este trabalho utiliza otimização por meio de algoritmos genéticos, apresentamos também neste capítulo conceitos fundamentais de algoritmos genéticos.

2.1 Atenção Visual

A todo instante, os olhos humanos se deparam com uma carga de estímulos visuais enorme. No entanto, é impossível processar toda a informação que chega aos olhos de uma só vez [Tsotsos, 1990]. O cérebro humano lida com este problema de várias formas. Em primeiro lugar, os olhos não captam toda a informação que está a sua frente. Apenas alguns pontos por segundo são tratados. Por meio de movimentos rápidos dos olhos, conhecidos como movimentos sacádicos (do inglês *saccadic eye movements*), o cérebro recebe somente parte da informação visual a cada instante. Portanto, para lidar com o excesso de informação, o sistema visual possui mecanismos para selecionar apenas um subconjunto de estímulos para um processamento rigoroso e executar apenas uma análise limitada sobre o restante das informações visuais.

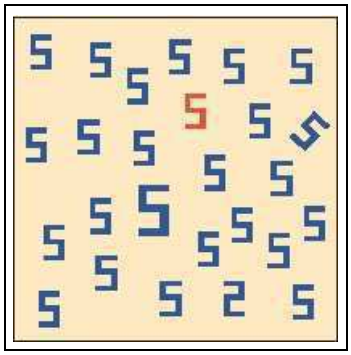
2.1.1 Inspiração Biológica

A atenção visual é a habilidade que o sistema visual dos vertebrados superiores utiliza para selecionar e processar somente as regiões mais relevantes em uma cena visual. A atenção visual pode ser entendida como um mecanismo para lidar com a incapacidade de tratar de uma só vez uma grande quantidade de informação visual tanto em sistemas biológicos quanto em sistemas computacionais. Deste modo, somente as regiões mais importantes numa cena são tratadas [Fischer and Weber, 1993]. Esta seleção das informações mais relevantes dos estímulos de entrada é uma das características mais importantes dos sistemas visuais biológicos que permite rápida detecção de predadores, perpetuação e evolução das espécies [Itti and Koch, 2001a]. Tsotsos [Tsotsos, 1990] analisou a complexidade computacional da análise visual e confirmou que a atenção visual é uma das mais importantes contribuições para otimizar a quantidade de computações em sistemas visuais.

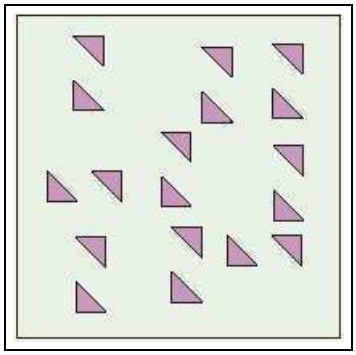
Em uma visão didática, podem ser identificados dois métodos principais para obtenção da Atenção Visual. Os métodos *top-down* e *bottom-up*. O método *top-down* usa conhecimentos obtidos a priori para detectar regiões de maior interesse numa imagem. Esses conhecimentos podem ser obtidos de várias formas. Geralmente, utilizam-se ferramentas de aprendizagem baseadas em modelos geométricos/relacionais (como redes semânticas ou grafos relacionais) ou modelos estatísticos (como redes neurais e máquinas de vetores de suporte). Porém, esses conhecimentos também podem ser fornecidos por um ser humano, selecionando-se manualmente regiões de maior interesse numa imagem. A atenção visual *bottom-up* é guiada por características primitivas da imagem como cor, intensidade e orientação. Além disso, ela atua de modo inconsciente, ou seja, o observador é levado a fixar sua atenção em determinadas regiões da imagem devido aos estímulos causados pelos contrastes entre características visuais presentes na imagem.

Wolfe e Horowitz demostraram que algumas características como cor, orientação ou tamanho dos objetos em uma imagem são responsáveis por guiar o mecanismo biológico de atenção visual [Wolfe and Horowitz, 2004]. A Tabela 2.1 mostra algumas destas características. Na Figura 2.1.1, há exemplos de tarefas de busca visual. Algumas destas tarefas são simples. Na Figura 2.1(a), o contraste entre o azul e o vermelho ressalta a existência de um numeral 5 (cinco) de cor diferente dos demais. No entanto, perceber um número cinco azul e maior é um pouco mais complicado. A Figura 2.1(a) também é um exemplo da importân-

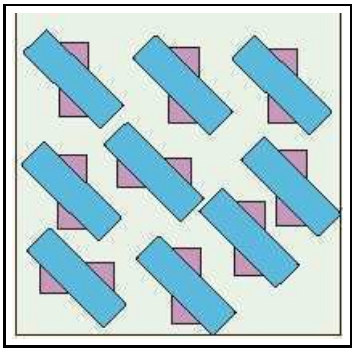
cia de conhecimento a priori para executar determinadas buscas visuais, pois dificilmente é possível identificar o número dois existente nesta imagem sem que alguém tenha dito que há um número dois. Isto demonstra o fato de que a atenção visual *top-down* é mais lenta e necessita de conhecimento prévio sobre o que se quer encontrar. As Figuras 2.1(b) e 2.1(c) demonstram a importância da orientação e do contraste de cores para ressaltar objetos diferentes em imagens. Na Figura 2.1(b) é difícil encontrar os pares de triângulos horizontais, mas esta tarefa é simplicada devido ao contraste de cores entre os retângulos azuis e os retângulos rosas. Na Figura 2.1(d), a busca por cruzes é ineficiente devido ao fato de que aqui a informação de intersecção não guia a atenção.



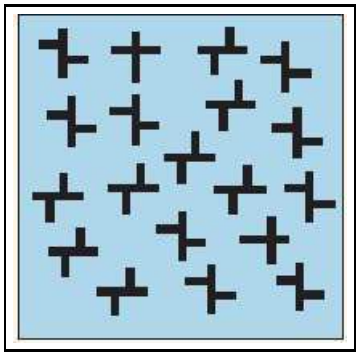
(a) Conhecimento a priori para executar buscas visuais.



(b) Contraste de cores.



(c) Contraste de orientações.



(d) Informação de intersecção não guia a atenção.

Figura 2.1: Exemplos de tarefas de busca visual.

Com certeza	**Provavelmente**	**Possivelmente**	**Talvez**
Cor	Luminância	Direção da Iluminação	Novidade
Movimento	Profundidade	*Aspect ratio*	Categoria alfanumérica
Tamanho	Terminação de Linha	Número	Tipo de Letra

Tabela 2.1: Características que podem guiar a atenção visual.

2.1.2 Modelo de Itti

Um dos métodos mais utilizados em atenção visual *bottom-up* é o que utiliza mapas de saliências. Itti et al. [Itti et al., 1998] propuseram um mecanismo de atenção visual *bottom-up* baseado em mapas de saliências, o qual é construido a partir de Pirâmides Gaussianas e operadores de vizinhança orientados localmente. A Figura 4.5 mostra um diagrama que representa o funcionamento deste mecanismo de atenção visual.

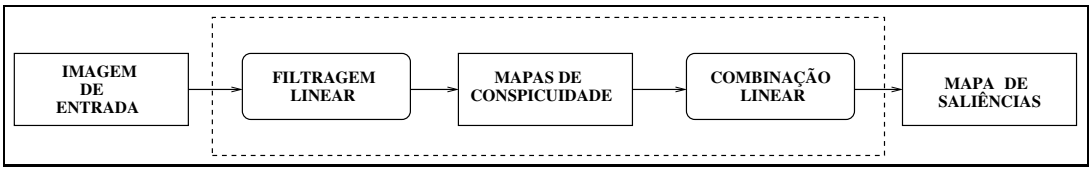


Figura 2.2: Mecanismo de atenção visual. A imagem de entrada passa por um processo de filtragem linear, gerando mapas de conspicuidade que são somados linearmente para gerar os mapas de saliências.

O modelo proposto por Itti implementado para os fins expostos neste trabalho é descrito a seguir. A implementação aqui descrita difere da apresentada por Itti em dois aspectos principais: quantidade de níveis das Pirâmides Gaussianas e método de movimentos sacádicos. No modelo de Itti, as Pirâmides Gaussianas possuem 9 níveis e no aqui implementado possuem 5 níveis. A justificativa para o uso de apenas 5 níveis nas pirâmides está relacionada à resolução das imagens utilizadas. Como a resolução das imagens é 352×240, em uma pirâmide de 5 níveis a imagem no menor nível terá sua resolução igual a 22×15 (equivalente a dividir a resolução do maior nível por 16). Desta forma, um maior número de níveis não incrementará informação importante ao sistema dado que a imagem num nível muito baixo seria insignificante. Itti et al [Itti et al., 1998] utilizam redes neurais para implementar movimentos sacádicos, no sistema aqui apresentado utiliza-se uma estratégia de deslocamento de

pixels. Porém, ambos os métodos podem ser divididos nas seguintes etapas: extração de características, filtragem linear, diferenças centro-vizinhaças, soma de mapas de características e seleção de regiões salientes (micro-sacadas).

2.1.3 Combinação de Mapas de Características

Para gerar um mapa de saliência, três tipos de características visuais primitivas são extraídas: cor, intensidade e orientação. Em seguida, quatro canais de cores são criados (*R* para vermelho, *G* para verde, *B* para azul e *Y* para amarelo). Sendo *r*, *g*, *b* os canais vermelho, verde e azul da imagem de entrada, os canais de cores são representados por:

$$R = r - (g + b)/2 \tag{2.1}$$

$$G = g - (r + b)/2 \tag{2.2}$$

$$B = b - (r + g)/2 \tag{2.3}$$

$$Y = (r + g)/2 - |r - g|/2 - b \tag{2.4}$$

A imagem de intensidades é representada por $I = (r+g+b)/3$, que define a imagem em tons de cinza. Para cada canal de cor e para a imagem de intensidades, são criadas Pirâmides Gaussianas: $R(\sigma)\ G(\sigma)\ B(\sigma)\ Y(\sigma)$ onde $\sigma \in \{0, 1, 2, 3, 4\}$. As Pirâmides Gaussianas são geradas utilizando um algoritmo proposto por Burt e Adelson [Burt and Adelson, 1983]. Informações de orientação local são obtidas pela aplicação de um algoritmo proposto por Freeman e Adelson [Freeman and Adelson, 1991], que trata de Pirâmides Direcionais.

Os canais de cores e a imagem de intensidades são submetidos a um processo de filtragem linear. Este processo é realizado por meio da geração de Pirâmides Gaussianas e Pirâmides Direcionais. A Pirâmide Gaussiana é composta por versões filtradas passa-baixa da convolução Gaussiana aplicada à imagem de entrada. A Pirâmide Direcional (*Steerable Pyramid*) é uma decomposição multi-escala e multi-orientação de uma imagem. Nesta decomposição linear, uma imagem é subdividida em um conjunto de sub-bandas localizadas em escala e orientação. A representação piramidal é usada para a obtenção de amostras da imagem sem detalhes indesejavéis. A seguir, os processos de geração das Pirâmides Gaussianas e Direcionais são detalhados.

A imagem de entrada é representada por uma matriz g_0, essa matriz contém C colunas e R linhas de *pixels*. Para cada nível da pirâmide é gerada uma imagem em uma escala menor que a escala no nível superior. A imagem de entrada é a base ou nível zero da Pirâmide Gaussiana. Cada nível inferior da pirâmide contém uma imagem que é uma redução ou uma versão filtrada passa-baixa da imagem da base da pirâmide. Os valores dos *pixels* de uma imagem num nível inferior são obtidos calculando-se uma média ponderada dos valores dos *pixels* num nível imediatamente superior dentro de uma janela 5×5. Este processo é realizado utilizando-se a função REDUZ.

$$g_k = REDUZ(g_{k-1}) \tag{2.5}$$

em que, para níveis $0 < l < N$ e nós $i, j, 0 \leq C_l, 0 \leq j < R_l$,

$$g_l(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) g_{l-i}(2i+m, 2j+n) \tag{2.6}$$

Na equação acima, N indica a quantidade de níveis da pirâmide, C_l e R_l indicam as quantidades de colunas e linhas do nível l, ou seja, as dimensões do nível l. Para que a imagem original seja adequada à construção de Pirâmides Gaussianas, devem existir os inteiros M_C, M_R e N de forma que $C = M_C 2^N + 1$ e $R = M_R 2^N + 1$. A multiplicação pela matriz de pesos w é equivalente à convolução da imagem por uma máscara gaussiana 5×5. Esta máscara é conhecida como núcleo (*Generating Kernel*) e seus valores são normalizados. A convolução da imagem pela máscara gaussiana é equivalente à aplicação de um *blur* ou filtro passa-baixa. A Figura 2.3 mostra um exemplo de Pirâmide Gaussiana.

As Pirâmides Gaussianas dos canais de cores e das imagens de intensidades de cada imagem de entrada são interpoladas. Para isso, utiliza-se a função EXPANDE que é definida como reversa de REDUZ. A função EXPANDE é utilizada com o objetivo de possibilitar a interpolação de imagens que estão em escalas diferentes. Por exemplo, a aplicação de EXPANDE a uma matriz da Pirâmide Gaussiana do nível 1 gera uma matriz que tem as mesmas dimensões de uma matriz do nível 0. A função EXPANDE é representada na Equação (2.7)

$$g_{l,n} = EXPANDE(g_{l,n-1}) \tag{2.7}$$

em que, para níveis $0 < l \leq N$ e $0 \leq n$ e nós $i, j, 0 \leq i < C_{l-n}, 0 \leq j < R_{l,n}$,

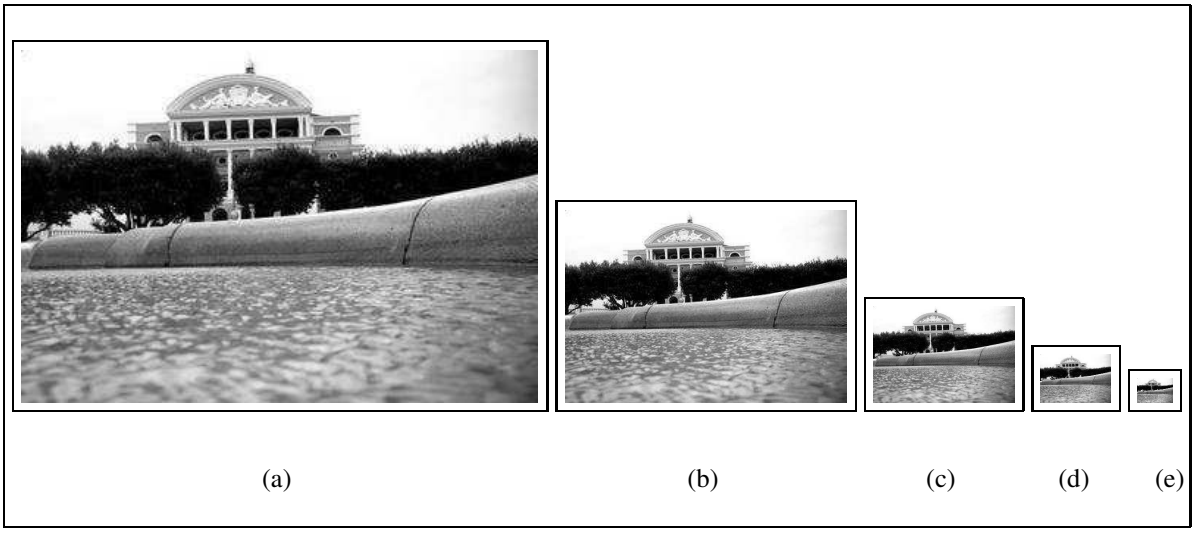


Figura 2.3: Pirâmide Gaussiana de 5 níveis.

$$g_{l,n} = 4 \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) \bullet g_{l,n-1}\Big(\frac{i-m}{2}, \frac{j-n}{2}\Big) \tag{2.8}$$

Neste trabalho, o termo *filtros direcionais* (do inglês *Steerable Filters*) é utilizado para descrever uma classe de filtros na qual um filtro de orientação arbitrária é sintetizado como uma combinação linear de um conjunto de *filtros base* [Freeman and Adelson, 1991]. A seguir, este conceito é demonstrado.

Seja uma função gaussiana circularmente simétrica bidimensional, G, escrita em coordenadas cartesianas, x e y:

$$G(x,y) = e^{x^2+y^2} \tag{2.9}$$

em que as constantes de normalização e escala são 1, por conveniência.

Aqui representaremos o operador de rotação por $(...)^\theta$, tal que $f^\theta(x,y)$ é a representação da função $f(x,y)$ rotacionada da origem por um ângulo θ. Representaremos, também, a enésima derivada de uma gaussiana na direção x por G_n. Desta forma, $f^\theta(x,y)$ é a representação da função $f(x,y)$ rotacionada de um ângulo θ da origem e a primeira derivada em relação a x de uma gaussiana, $G_1^{0^\circ}$, é

$$G_1^{0^\circ} = \frac{d\, e^{-(x^2+y^2)}}{d\, x} = -2xe^{-(x^2+y^2)} \tag{2.10}$$

Esta mesma função rotacionada 90° é

$$G_1^{90^\circ} = \frac{d\, e^{-(x^2+y^2)}}{dy} = -2ye^{-(x^2+y^2)} \tag{2.11}$$

A demonstração de que um filtro G_1 em uma orientação arbitrária θ pode ser sintetizado pela combinação linear de $G_1^{0^\circ}$ e $G_1^{90^\circ}$ é simples [Freeman and Adelson, 1991]:

$$G_1^\theta = cos(\theta)G_1^{0^\circ} + sen(\theta)G_1^{90^\circ} \tag{2.12}$$

Assim, $G_1^{0^\circ}$ e $G_1^{90^\circ}$ podem ser chamadas filtros base de G_1^θ. Os termos $cos(\theta)$ e $sen(\theta)$ são as funções de interpolação correspondentes para estes filtros base. Como a convolução é uma operação linear, pode-se sintetizar uma imagem filtrada em uma orientação arbitrária pela combinação linear das imagens filtradas com $G_1^{0^\circ}$ e $G_1^{90^\circ}$. Deste modo, representando a convolução pelo símbolo $*$ (asterisco), temos

$$R_1^{0^\circ} = G_1^0 * I \tag{2.13}$$

$$R_1^{90^\circ} = G_1^{90} * I \tag{2.14}$$

$$R_1^\theta = cos(\theta)R_1^{0^\circ} + sen(\theta)R_1^{90^\circ} \tag{2.15}$$

O exposto acima ilustra de forma simples como é possível extrair informações sobre orientação utilizando diferenciação de filtros gaussianos. A seguir, será feita uma análise da diferenciação de filtros direcionais no domínio de Fourier.

Como no domínio de Fourier a decomposição de filtros é polar-separável [Simoncelli and Freeman, 1995], a magnitude do i-ésimo filtro passa-banda será escrita em forma polar-separável:

$$B_i(w^\rightarrow) = A(\theta - \theta_i)B(w) \tag{2.16}$$

em que $\theta = tan^{-1}(w_y/w_x)$, $\theta_i = 2\pi/k$ e $w = |w^\rightarrow|$. As restrições sobre os componentes $A(\theta)$ e $B(w)$ são descritas abaixo.

Uma derivada direcional no domínio espacial corresponde a multiplicação por uma rampa linear no domínio de Fourier, assim, a porção angular da decomposição ($A(\theta)$) pode ser reescrita como:

$$-jw_x = -jw\ cos(\theta) \tag{2.17}$$

A função radial é implementada utilizando uma decomposição recursiva com um algoritmo de pirâmide. Desta forma, são necessários filtros passa-alta e passa-baixa ($H_0(w)$ e $L_0(w)$) para realizar o pré-processamento da imagem antes da recursão. As restrições sobre os filtros $H_0(w)$ e $L_0(w)$ são:

- Limitação de banda:

  $L_1(w) = 0$ para $|w| > \pi/2$

- Resposta do sistema flat (linear):

  $|H_0(w)|^2 + |L_0(w)|^2[|L_1(w)|^2 + |B(w)|^2] = 1$

- Recursão

  $|L_1(w/2)|^2 = |L_1(w/2)|^2[|L_1(w)|^2 + |B(w)|^2]$

A Figura 2.4 mostra uma Pirâmide Direcional com 3 níveis

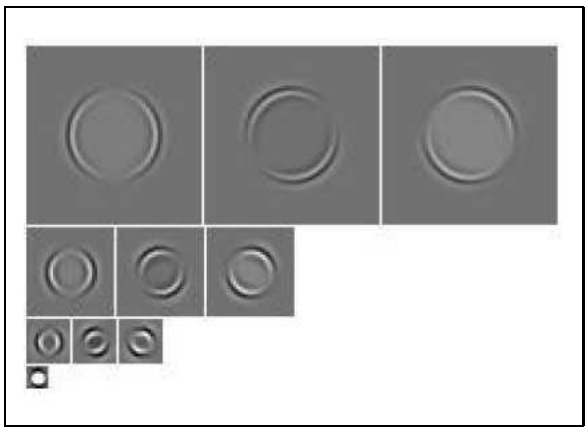


Figura 2.4: Pirâmide Direcional.

Os mapas de características são obtidos por meio da diferença entre canais de cores em diferentes escalas, este processo é conhecido como diferença centro-vizinhança. Nesta subtração de imagens, o centro é um *pixel* da imagem em uma escala $c \in \{1, 2\}$ e a vizinhança é o *pixel* correspondente de outra imagem em uma escala $v \in \{3, 4\}$ da pirâmide. Para que a diferença entre duas imagens em escalas diferentes seja realizada, aplicam-se interpolações

utilizando as funções EXPANDE e REDUZ. A partir da combinação das escalas c e v e da orientação θ, são produzidos 28 mapas de características. As Equações de (2.18) a (2.21) definem matematicamente as diferenças centro-vizinhanças.

$$\mathcal{I}(c,v) = |I(c)\Theta I(v)| \tag{2.18}$$

$$\mathcal{RG}(c,v) = |(R(c)-G(c))\Theta(G(v)-R(v))| \tag{2.19}$$

$$\mathcal{BY}(c,v) = |(B(c)-Y(c))\Theta(Y(v)-B(v))| \tag{2.20}$$

$$\mathcal{O}(c,v,\theta) = |O(c,\theta)\Theta O(v,\theta)| \tag{2.21}$$

em que $\theta \in (0°, 45°, 90°, 135°)$.

O processo de geração de todos esses mapas de características é inspirado biologicamente. A geração dos mapas de cores tem inspiração no sistema de cores oponentes do córtex visual [Itti and Koch, 2001a]. Os mapas de orientação são inspirados na propriedade que alguns neurônios do córtex visual possuem de responder a estímulos de orientação da cena [Itti and Koch, 2001a].

Uma vez que os mapas de características foram obtidos, eles são somados para a produção dos mapas de conspicuidades: $\overline{\mathcal{I}}$ para intensidade, $\overline{\mathcal{C}}$ para cor e $\overline{\mathcal{O}}$ para orientação, na escala $\sigma = 4$. A motivação para a criação de três canais separados $(\overline{\mathcal{I}}, \overline{\mathcal{C}}, \overline{\mathcal{O}})$ é a hipótese de que características similares competem pela saliência, enquanto que características diferentes contribuem independentemente para o mapa de saliência [Itti and Koch, 2001a]. O propósito do mapa de saliência é representar regiões salientes na imagem por quantidades escalares e guiar a seleção de regiões baseada na distribuição espacial da saliência. As Equações de (2.22) a (2.25) modelam matematicamente o processo de soma dos mapas de características.

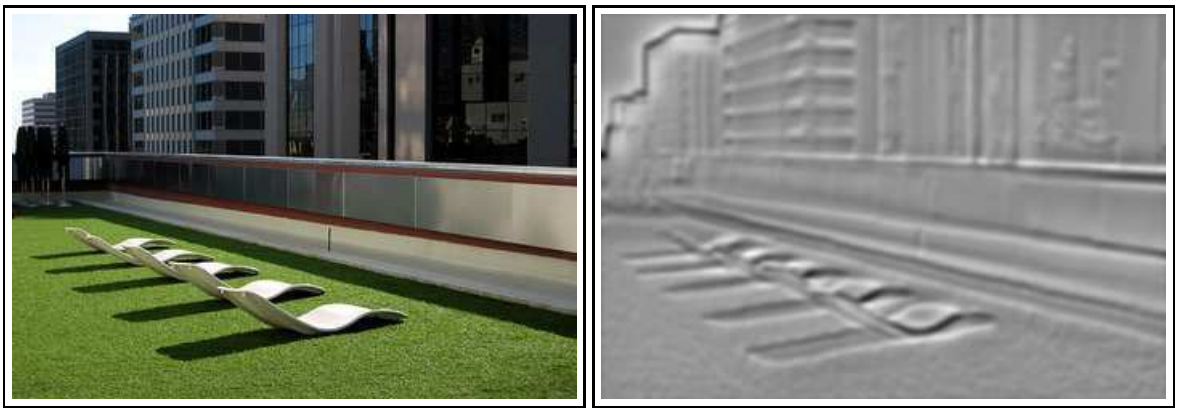
$$\overline{\mathcal{I}} = \bigoplus_{c=1}^{2}\bigoplus_{v=3}^{4}\mathcal{N}(\mathcal{I}(c,v)) \tag{2.22}$$

$$\overline{\mathcal{C}} = \bigoplus_{c=1}^{2}\bigoplus_{v=3}^{4}[\mathcal{N}(BY(c,v))] \tag{2.23}$$

$$\overline{\mathcal{O}} = \sum_{\theta\in\{0°,45°,90°,135°\}} \mathcal{N}\Big(\bigoplus_{c=1}^{2}\bigoplus_{v=3}^{4}\mathcal{N}(\mathcal{O}(c,v,\theta))\Big) \tag{2.24}$$

$$\mathcal{S} = \frac{1}{w_1+w_2+w_3}(w_1\overline{\mathcal{I}} + w_2\overline{\mathcal{C}} + w_3\overline{\mathcal{O}}) \tag{2.25}$$

Estas equações representam a soma e a normalização dos mapas de características nas escalas $\{1,2\}$ e $\{3,4\}$ resultando em mapas de conspicuidades e a soma normalizada dos mapas de conspicuidade para gerar o mapa de saliência. A Figura 2.5 mostra uma imagem e seu respectivo mapa de saliência.



(a) Imagem Original

(b) Mapa de Saliência Resultante

Figura 2.5: Exemplo de mapa de saliência.

O mapa de saliência resultante é uma imagem em tons de cinza em que as regiões mais salientes são representadas por *pixels* de altas intensidades. Desta forma, podem ocorrer regiões que possuem *pixels* com valores iguais. Para evitar que uma mesma região seja determinada como mais saliente mais de uma vez e para que seja possível determinar várias regiões salientes, mesmo que tais regiões possuam *pixels* de mesmo valor, utiliza-se o princípio da inibição de retorno.

No modelo de Itti, utilizam-se redes neurais *winner-takes-all* para inibir regiões previamente selecionadas. Porém, por simplicidade, a implementação realizada nessa dissertação utilizou um processo heurístico que funciona como descrito a seguir. Inicialmente, define-se um raio de inibição. Este raio de inibição indica um raio que é medido em quantidade de *pixels* e a partir dele é definida a área que será inibida na próxima iteração aplicando-se valores nulos aos *pixels* desta região.

Os *pixels* que serão utilizados como centro da região saliente são determinados por um processo de movimentos sacádicos, ou micro-sacadas. Biologicamente, movimentos sacádicos são os movimentos realizados pelo olho humano durante o processo de inspeção visual de uma cena. Em seres humanos, estes movimentos são executados de maneira bastante rápida (entre 4 e 6 por segundo).

No sistema desenvolvido nessa dissertação, os movimentos sacádicos foram implementados como descrito a seguir. Primeiro, obtém-se o *pixel* que possui o maior valor de intensidade. Em seguida, inibe-se toda a região vizinha de acordo com o raio de inibição previamente determinado. As micro-sacadas são determinadas deslocando-se as coordenadas do ponto de atenção 5 e 10 *pixels* em uma vizinhança de 8 *pixels*, gerando 16 variações de pontos de atenção.

A seguir, apresentaremos alguns dos principais métodos utilizados para combinação de características primitivas utilizadas para formar mapas de saliências nos processos de atenção visual *bottom-up*, semelhantes ao proposto por Itti et al. [Itti et al., 1998].

No método de detecção de objetos proposto por Itti et al. [Itti et al., 1998] os mapas de conspicuidade são normalizados e somados. Há outras formas de se combinar os mapas de conspicuidade. Em Itti e et al. [Itti and Koch, 2001b] quatro estratégias são comparadas:

- Somatório normalizado;
- Combinação linear dos mapas utilizando pesos resultantes de processo de aprendizagem;
- Normalização não-linear;
- Competição não-linear entre localizações salientes seguida de somatório.

A abordagem mais simples para combinação dos mapas de conspicuidade é o somatório normalizado. Tal somatório pode ser expresso pela seguinte equação:

$$S = w_1\mathcal{M}_1 + w_2\mathcal{M}_2 + \ldots w_3\mathcal{M}_n \tag{2.26}$$

em que os $\mathcal{M}$ representam mapas de conspicuidade e os w representam os pesos aprendidos. No caso de um somatório normalizado (como aquele usado em [Itti et al., 1998]), cada $w = \frac{1}{n}$, em que n é igual ao número de mapas de conspicuidade.

Um dos modos para se detectar objetos específicos é utilizar aprendizagem supervisionada. Essa estratégia consiste em usar uma técnica de aprendizagem para determinar pesos que serão atribuídos aos mapas de conspicuidade. No processo de aprendizagem proposto por Itti e et al. [Itti and Koch, 2001b], uma região alvo é delimitada manualmente e, em seguida, o seguinte procedimento é realizado:

1 Computa-se o mínimo global M_{glob} e o mínimo global m_{glob} do mapa $\mathcal{M}$;

2 Computa-se o mínimo dentro M_{in} e o mínimo fora M_{out} da região alvo.

3 Atualiza-se o peso seguindo-se a seguinte regra:

$$w(\mathcal{M}) \leftarrow w(\mathcal{M}) + \mathcal{N}\frac{(M_{in} - M_{out})}{(M_{glob} - m_{glob})} \tag{2.27}$$

em que $\mathcal{N}$ determina a velocidade de aprendizagem.

Esse procedimento de aprendizagem promove, através de um aumento nos pesos, a participação dos mapas de conspicuidade, que aprensentam maior pico de atividade dentro das regiões de interesse, no mapa de saliência.

Quando não há supervisão disponível, utiliza-se um esquema de normalização simples. Esse esquema consiste em promover os mapas de conspicuidade que apresentam uma certa quantidade de topos de atividades, enquanto que suprime os mapas de conspicuidade que apresentam picos de respostas semelhantes em várias localizações da cena visual. Um operador de normalização não-linear é obtido da seguinte forma:

1 Normalizam-se todos os mapas de conspicuidade;

2 Para cada mapa, encontra-se o mínimo global M e a média $\overline{m}$ de todos os outros mínimos locais;

3 O mapa é multiplicado globalmente pelo seguinte fator: $(M - \overline{m})^2$

A quarta estratégia de combinação de características se baseia na simulação de competição local entre localizações salientes vizinhas. O princípio geral é prover auto-excitação e inibição induzida pelos vizinhos para cada localização no mapa de conspicuidade. Para isso, cada mapa de conspicuidade é iterativamente convoluido por um filtro de diferenças gaussianas $2D - DoG$ (*Difference of Gaussians*).

Em cada iteração do processo de normalização um dado mapa de conspicuidade $\mathcal{M}$ é submetido à seguinte transformação:

$$\mathcal{M} \leftarrow |\mathcal{M} + \mathcal{M} * DoG - C_{inh}| \tag{2.28}$$

em que C_{inh} é o termo de inibição constante.

Neste trabalho, apresentamos um método de combinação de mapas de características e conspicuidades que utiliza otimização por meio de algoritmos genéticos. Resultados experimentais mostram que o método proposto é capaz de guiar a atenção para regiões específicas de imagens, por exemplo: regiões contendo pessoas, ou faces de pessoas. Este método será descrito no Capítulo 4 e os resultados experimentais serão apresentados no Capítulo 5.

2.2 Algoritmos Genéticos

Há vários métodos para solucionar problemas de otimização não-lineares, alguns bastante utilizados são: *hill climbing*, *simulated annealing* e algoritmos genéticos. Outra forma seria por busca exaustiva, ou seja, gerar aleatoriamente todas as soluções possíveis e testar qual delas se aplica ao problema. No entanto, se o número de boas soluções para um problema é esparso em relação ao espaço de busca, então uma busca aleatória não é uma forma prática para resolver o problema [Whitley, 1994].

Um problema que pode ocorrer durante a otimização utilizando métodos como *hill climbing* e *simulated anneling* é a otimização convergir para ótimos locais. Algoritmos genéticos lidam com este problema através de mutações que permitem aumentar a variabiliade das soluções avaliadas a cada iteração. Porém, um problema relacionado aos algoritmos genéticos é a necessidade de um grande poder de processamento devido à grande quantidade de soluções que devem ser avaliadas a cada iteração.

Nesta seção, apresentaremos o conceito de algoritmos genéticos. Tal conceito surgiu dos trabalhos de Holland [Holland, 1975] e DeJong [DeJong, 1975]. Algoritmos genéticos são bastante utilizados na resolução de problemas de otimização, são métodos heurísticos baseados na teoria evolucionária das espécies de Darwin. Tais algoritmos buscam através da simulação da evolução gerar populações de soluções para determinados problemas. São úteis especialmente quando as variáveis envolvidas no problema não podem ser tratadas isoladamente. Além disso, aplicam-se a funções não diferenciáveis ou que possuem vários ótimos locais e realizam uma busca global que não usa informação de gradiente. Algoritmos genéticos podem ser classificados como um método de busca fraco (*weak method*), pois não fazem nenhuma suposição sobre o problema tratado.

2.2.1 Inspiração Biológica

Várias teorias de outras áreas do conhecimento humano tem servido como fonte de inspiração para a Ciência da Computação. Uma teoria das Ciências Biológicas que influenciou o pensamento humano sobre a origem dos seres vivos e conseqüentemente a Ciência da Computação foi a teoria da evolução das espécies de Charles Darwin [Darwin, 1909]. Segundo essa teoria, os indivíduos encontram-se em uma luta constante pela sobrevivência. Nesta luta, apenas aqueles que possuem as características favoráveis à adapatação ao meio sobrevivem, transmitindo essas características às gerações futuras.

Com base na teoria da evolução foi criado o conceito de algoritmos genéticos. Algoritmos genéticos são uma técnica de programação inspirada nos mecanismos de evolução natural e recombinação genética. Os algoritmos genéticos fornecem um mecanismo de busca adaptativa que se baseia no princípio Darwiniano de reprodução e sobrevivência dos mais aptos. Isto é obtido a partir de uma população de indivíduos (soluções), representados por cromossomos (palavras binárias), cada um associado a uma aptidão (avaliação do problema), que são submetidos a um processo de evolução (seleção e reprodução) por vários ciclos. Assim, os algoritmos genéticos funcionam como otimizadores de funções.

2.2.2 Algoritmo Genético Clássico

O tipo de algoritmo genético discutido nesta subseção é o clássíco (Algoritmo Genético Clássico - AGC) [DeJong, 1975]. Para que um problema possa ser resolvido utilizando algoritmos genéticos é necessário que seu conjunto de soluções seja passível de ser mapeado em cadeias de bits. Este mapeamento constitui a primeira etapa no processo de resolução de um problema utilizando algoritmos genéticos. Cada cadeia de bits, representando uma possível solução do problema, pode ser chamada de genótipo, indivíduo ou cromossomo.

O funcionamento deste método pode ser dividido em cinco passos: gerar uma população inicial, avaliar os indivíduos da população, selecionar os indivíduos aptos, realizar recombinação (*crossover*) e mutação, e gerar uma nova população. Geralmente, as populações iniciais são geradas aleatoriamente.

A avaliação dos indivíduos de cada população é realizada verificando-se a aplicabilidade de cada solução intrínseca em cada indivíduo para a resolução do problema. Dependendo de sua aplicabilidade, a cada indivíduo é atribuído um valor. Em seguida, verifica-se a aptidão de os indivíduos fazerem parte da próxima geração utilizando-se uma *função de aptidão*. A aptidão de um indivíduo é calculada pela divisão do valor que foi atribuído a ele pela função de avaliação pela média dos valores de todos os indivíduos da população. Por exemplo, se o indivíduo *a* obteve um valor de avaliação $f_a = 5$ e o valor médio de avaliação da população é $\overline{f} = 4$, seu valor de aptidão será $f_a/\overline{f} = 1{,}25$.

O valor de aptidão é utilizado no processo de seleção dos indivíduos que serão duplicados para fazerem parte dos processos de recombinação e mutação. A probabilidade de um indivíduo ser selecionado é proporcional à sua aptidão. Para selecionar os indivíduos mais aptos utilizam-se os métodos da roleta (*roulette wheel*) e de amostragem estocástica com reposição (*stochastic sampling with replacement*). Estes métodos de seleção podem ser vistos como uma roleta em que as regiões onde o marcador pára possuem áreas diferentes. Cada região representa um indivíduo da população, sendo a área de cada região proporcional à aptidão do indivíduo representado. Logo, quanto maior a aptidão do indivíduo, maior a probabilidade da sua seleção ser efetuada. Por exemplo, se um indivíduo possui valor de aptidão $f_i/\overline{f} = 1{,}37$, este valor indica que o mesmo será selecionado uma vez e que há $0{,}37$ de chances do indivíduo ser selecionado novamente. Por outro lado, se um indivíduo possui valor de aptidão $f_i/\overline{f} = 0{,}63$, este valor indica que há $0{,}63$ de chances do indivíduo ser

selecionado. A Figura 2.6 ilustra este método de seleção.

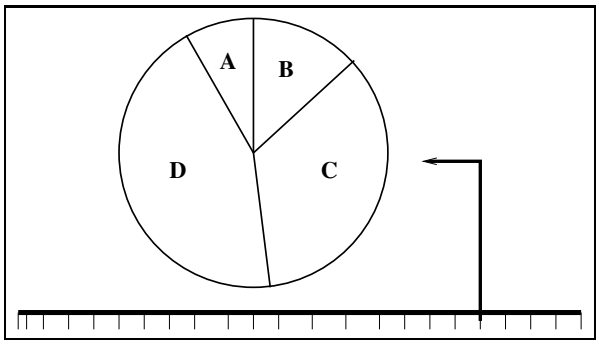


Figura 2.6: Ilustração do método de seleção. As regiões com áreas maiores possuem maiores probabilidades de serem selecionadas. Por exemplo, há uma maior probabilidade da região **D** ser selecionada do que a região **A**.

Após os indivíduos aptos terem sido selecionados, ocorre uma troca de informações conhecida como recombinação. A recombinação atua de maneira probabilística, trocando parte de uma cadeia de bits de um indivíduo por parte de uma cadeia de bits de outro indivíduo. Neste processo, duas cadeias de bits são emparelhadas aleatoriamente. Em seguida, escolhe-se em que ponto as cadeias serão quebradas segundo uma probabilidade previamente determinada. Por exemplo, supondo-se que as cadeias que sofrerão recombinação são 11000101 e $abbbaaaa$, e a probabilidade de recombinação determina que elas podem ser quebradas na sexta posição, as cadeias resultantes seriam: $110001aa$ e $abbbaa01$. Os indivíduos resultantes das recombinações irão compor uma nova população.

A fim de que haja uma maior variabilidade de indivíduos na nova população, tais indivíduos passam por um processo chamado de mutação. A mutação é uma mudança da disposição dos bits que compõem uma cadeia e ocorre segundo uma probabilidade pré-determinada muito baixa. Por exemplo, se a probabilidade de mutação do indivíduo $110001aa$ determina que ocorra uma inversão do primeiro e terceiro bits, o indivíduo mutante passa a ser $011001aa$. O conjunto de indivíduos mutantes comporá a nova população. Cada ciclo de seleção, recombinação e mutação é conhecido como geração. Este ciclo se repete até que seja atingida a melhor solução, a curva de evolução se estabilize ou o número máximo de gerações seja alcançado. A Figura 2.7 resume o processo de geração de populações de um algoritmo genético.

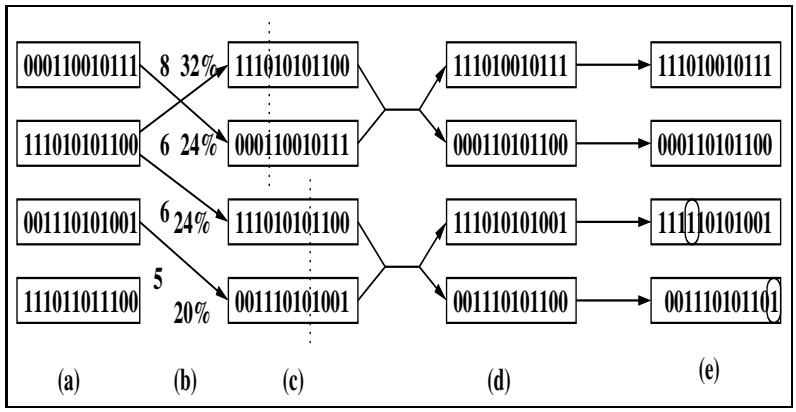


Figura 2.7: Ilustração da geração de uma nova população em um algoritmo genético. Em (a), temos uma população inicial de 4 indivíduos. A aptidão dos indivíduos é obtida pela função de aptidão em (b). Em (c), dois pares são selecionados. Em (d), vemos uma nova prole, gerada pelo cruzamento. Finalmente, em (e), ocorre mutação e uma nova população é gerada.

2.2.3 Breve Demonstração da Eficácia dos Algoritmos Genéticos

Uma maneira simples de demonstrar a eficácia dos algoritmos genéticos na solução de problemas de otimização foi apresentada por Holland [Holland, 1975]. Esta demonstração representa os espaços de busca como hiperplanos. Suponha que um espaço de busca analisado é constituído por todas as cadeias de 3 bits (8 cadeias). Este espaço de busca pode ser representado pelo cubo da Figura 2.8.

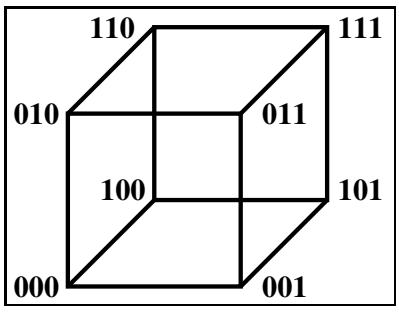


Figura 2.8: Representação de um espaço de busca como hiperplanos formadores de um cubo.

O plano frontal do cubo contém todos os pontos que começam com 0. Se $*$ é usado como um símbolo coringa, então este plano pode ser representado pela cadeia 0^{**} . Cadeias que contêm * são denominadas *schemata*, Cada *schema* corresponde a um hiperplano no espaço de busca. Desta forma, observa-se que uma população de cadeias provê informação sobre

vários hiperplanos e o número de hiperplanos amostrados é maior que o número de cadeias contido na população.

Assim, o algoritmo genético contém paralelismo intrínseco. Tal paralelismo é derivado do fato de que muitos hiperplanos são amostrados quando uma população de cadeias é avaliada [Holland, 1975]. Isto demonstra que os algoritmos genéticos atuam com amostras de todas as soluções possíveis e que seu paralelismo implícito resolve a competição entre os hiperplanos. Porém, são os efeitos cumulativos da avaliação de uma população que provêm informação estatística sobre qualquer subconjunto de hiperplanos.

2.3 Considerações Finais

Neste capítulo, foram apresentados os principais conceitos envolvidos nesta dissertação: atenção visual *bottom-up* e algoritmos genéticos. A atenção visual *bottom-up* funciona gerando mapas de saliências derivados de vários mapas de características visuais primitivas como cor, intensidade e orientação. Os algoritmos genéticos são um método geralmente utilizado para otimizar funções que foi criado com inspiração na teoria da evolução das espécies. O próximo capítulo apresenta uma análise de trabalhos relacionados com esta dissertação.

Capítulo 3

Revisão Bibliográfica

Este capítulo apresenta um levantamento e análise de trabalhos relacionados com esta dissertação, focando em trabalhos que investigam ou propõem métodos que utilizam atenção visual em sistemas de reconhecimento de imagens, ou que utilizam algoritmos genéticos como meio de otimização de sistemas de visão computacional.

A Seção 3.1 discute trabalhos que propõem métodos de integração de modelos de atenção visual *bottom-up* e *top-down*. A Seção 3.2 apresenta artigos que propõem sistemas de detecção e reconhecimento de objetos que utilizam atenção visual *bottom-up* como meio de aumentar o desempenho da busca por regiões a serem processadas. Na Seção 3.3, temos uma análise de trabalhos que utilizam algoritmos genéticos em sistemas de visão computacional. Finalmente, a Seção 3.4 apresenta as considerações finais sobre os trabalhos discutidos.

3.1 Integração de Modelos de Atenção Visual Top-down e Bottom-up

Uma arquitetura para estimar quais as regiões mais relevantes em uma cena foi proposta por Navalpakkam e Itti [Navalpakkam and Itti, 2002]. Nesta arquitetura, um grafo de regiões relevantes é construído utilizando uma ontologia que contém a descrição de entidades presentes na imagem e seus relacionamentos. A atenção é guiada por um mapa de atenção topográfico que codifica a saliência e a relevância de todas as regiões da cena.

O modelo é composto por quatro componentes: cérebro visual, memória de trabalho

(WM), memória de longo prazo (LTM) e agente. O cérebro visual mantém três mapas: mapa de saliências (SM), mapa de relevância (RM) e um mapa para guiar a atenção. O mapa para guiar a atenção é o resultado do produto entre o SM e o RM. A memória de trabalho cria e mantem o grafo que contém todas as entidades relevantes da cena. O papel do agente é transmitir informação entre o cérebro visual e a memória de trabalho.

A LTM atua como uma base de conhecimento. Ela contém as entidades e seus relacionamentos e é chamada de ontologia. Cada ontologia é representada como um grafo em que as entidades são os vértices e os relacionamentos são as arestas. Cada entidade possui uma lista de propriedades separada da lista de todos os seus vizinhos. Estas propriedades podem servir como guias para o módulo de reconhecimento. A WM estima a relevância de uma fixação para uma dada tarefa. O cálculo da relevância de uma fixação é uma função da natureza das relações que conectam uma entidade ao grafo e da relevância de seus vizinhos.

O modelo foi testado em cenas de ambientes naturais com muitos elementos dispersivos. Para verificar o modelo, o sistema foi executado com várias imagens com o mesmo objetivo e em uma mesma imagem com objetivos diferentes. Por exemplo, em cenas de ruas de cidades, o objetivo foi encontrar carros. Em outro experimento, utilizou-se uma cena com pessoas comendo e determinou-se que o sistema encontrasse as faces das pessoas e o que elas estavam comendo.

Apesar da análise dos resultados mostrar que o sistema apresentou bons resultados nos experimentos, o artigo não apresenta nenhum dado objetivo, como gráficos ou valores estatísticos. A análise é apenas subjetiva. Outro problema é que o artigo não mostra como as ontologias e seus atributos são criados, não especifica se foram criados para os fins do trabalho ou se foram obtidos de alguma base.

Navalpakkam e Itti [Navalpakkam and Itti, 2006] propuseram um modelo para sistema de atenção visual que integra os métodos *top-down* e *bottom-up*. O componente *bottom-up* do modelo computa a saliência visual da cena por meio de mapas de características extraidos de imagens em várias escalas. O componente *top-down* utiliza conhecimento estatístico acumulado das características visuais do objeto que é alvo da busca.

O principal conceito utilizado por este modelo para maximizar a velocidade de detecção é o SNR (Signal to Noise Ratio). O SNR é a razão entre a saliência do alvo de busca e a saliência dos objetos dispersivos do fundo da imagem. Para aumentar a velociade de detecção

deve-se maximizar o SNR. A saliência, S_j, de uma dada região, j, é calculada como uma combinação linear de saliências *bottom-up* s_{ij} para as características daquela região:

$$S_j(x, y, A) = \sum_{i=1}^{n} g_{i,j} s_{i,j}(x, y, A) \tag{3.1}$$

A saliência do alvo (S_T) é calculada em termos de sua saliência s_{iT}, $i \ \epsilon\{1, ..., n\}$, $j \ \epsilon \ \{1, ..., \ N\}$ para cada um dos *n* mapas de saliência dentro das *N* regiões das características. A saliência *bottom-up* é calculada utilizando o modelo de Itti et al. [Itti et al., 1998]. Foram utilizados os seguintes conjuntos de características visuais primitivas: 6 cores, 4 intensidades e 4 orientações ($0°$, $45°$, $90°$, $135°$). Os mapas de características são extraídos em 6 escalas espaciais diferentes. Tanto os mapas de caracerísticas quanto os de conspicuidade são ponderados por ganhos *top-down* e são combinados linearmente.

Para a realização dos experimentos, foram implementados 4 modelos: T0D0, T1D0, T0D1 e T1D1, em que T e D referem-se a alvo e distrator respectivamente. O 0 à direita da letra indica que o modelo não utiliza conhecimento sobre o elemento indicado pela letra, enquanto o 1 indica que o modelo utiliza tal conhecimento. Este conhecimento é obtido pelo cálculo da média dos SNR's para cada elemento. Por exemplo, T1D0 combina a saliência *bottom-up* apenas com conhecimento sobre o alvo. Os experimentos foram realizados utilizando tanto imagens com objetos artificiais (barras horizontais, verticais em diferentes cores, por exemplo) quanto com imagens de objetos reais (foto de vários objetos sobre uma mesa, por exemplo).

Foram realizados dois tipos de experimentos com imagens sintéticas e com imagens de ambientes naturais. O conjunto de imagens sintéticas continha 150 imagens e o de imagens naturais, 60. Todos os modelos obtiveram bons resultados nos testes em que o elemento alvo era muito diferente dos distratores. A busca era mais lenta quando havia algumas características semlhantes entre os elementos alvo e distratores. O artigo compara o modelo apenas entre variações do mesmo, não faz nenhuma comparação com outros modelos. Além disso, não mostra nenhum dado estatístico sobre o desempenho do sistema.

Fisher e MacKirdy [Fisher and MacKirdy, 1998] propuseram um sistema que utiliza processos *bottom-up* e *top-down* para reconhecer objetos. O processo *top-down* representa objetos como um ente inteiro (completo) no processo de reconhecimento. O processo *bottom-up* usa um conjunto de características relacionadas reconhecidas a priori.

O sistema utiliza coordenadas log-polar (R,θ) para foveamento. A representação polar é atrativa porque ela mapeia rotação e escala em translação e esta característica é usada no algoritmo de *matching* (correspondência). As principais representações são: o mundo (uma grande imagem estática), a pilha de imagens (42 imagens log-polar em 3 escalas diferentes), a base do modelo (um conjunto de modelos que podem ser comparados com a pilha de imagens atual), o mapa de interesse (seu conteúdo registra valores que representam o interesse de um dado ponto da cena).

O processo de comparação utiliza uma função de correlação cruzada modificada. A arquitetura do sistema foi extendida com cinco estruturas ou processos: representação estruturada do modelo (os modelos associados incluem subcomponentes tanto quanto objetos associados de maneira mais geral), registro de evidência de subcomponente (de acordo com a posição relativa dos subcomponentes, pode-se obter a posição do modelo), função de avaliação de *match* extendida, atualização do mapa de interesse (o mapa de interesse original é atualizado pelo cálculo de uma função de interesse em cada uma das 3 escalas de 14 características).

Nos experimentos, foi utilizado um conjunto de imagens contendo vistas frontais de faces. Em cada imagem de face, os olhos, o nariz e a boca representavam os modelos associados. Com o intuito de demonstrar que o uso da evidência de subcomponentes melhora a velocidade, a precisão posicional e a completude do reconhecimento, o sistema de reconhecimento icônico foi executado com e sem a habilitação do processo de evidência de subcomponentes. Todos os experimentos iniciavam com um foveamento no centro da imagem. O critério de parada era que todas as características fossem encontradas ou que o sistema tivesse executado 20 movimentos sacádicos. Os resultados experimentais demonstraram que a afirmação feita anteriormente de que a evidência de subcomponentes melhora o processo de reconhecimento é verdadeira.

O artigo mostra por meio de resultados experimentais que a integração de características estruturais a um modelo que utiliza características de baixo nível pode melhorar o processo de reconhecimento. No entanto, os testes são executados com apenas uma classe de objetos e uma quantidade muito pequena de imagens.

3.2 Uso de Atenção Visual na Melhoria do Desempenho de Sistemas de Reconhecimento de Padrões

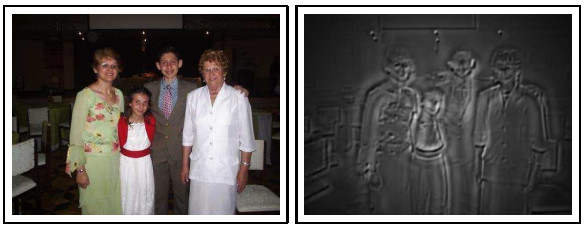
Walther et al [Walther et al., 2002] apresentam um sistema que realiza reconhecimento de objetos por meio da seleção prévia de regiões salientes. Esta seleção prévia é feita utilizando atenção visual *bottom-up*, segmentação e erosão de mapas de saliência. Desta forma, o sub-sistema de atenção visual funciona como um detector dos objetos mais importantes da imagem.

Inicialmente, a imagem é processada visando a obtenção de seus mapas de características e conspicuidades. São gerados vários mapas para cada classe de características. Por exemplo, para cor são gerados quatro mapas de características (para vermelho, azul, verde e amarelo). Para cada classe de característica é gerado um mapa de conspicuidade que representa a saliência para cada tipo de característica. Neste processo, são gerados mapas de conspicuidades para três características: cor, orientação e intensidade. O mecanismo de extração de mapas de características e conspicuidades empregado é semelhante ao proposto por Itti et al. [Itti et al., 1998]. Este mecanismo é construído utilizando Pirâmides Gaussianas [Burt and Adelson, 1983] e operadores de vizinhaça orientados localmente e é descrito no Capítulo 2.

Em seguida, verifica-se quais os pontos mais salientes da imagem e quais os mapas de características que mais contribuíram para que estes pontos fossem os mais salientes. A Figura 3.1 ilustra a escolha do mapa de conspicuidade mais importante para a saliência da imagem de exemplo. Os mapas de características não foram apresentados por que há uma grande quantidade dos mesmos (4 para cor, 4 para intensidade e 16 para orientação) e eles são semelhantes aos mapas apresentados.

O mapa de característica que mais contribuiu para a determinação do ponto mais saliente é segmentado utilizando-se um algoritmo de *flooding* com limiarização adaptativa. O mapa de característica segmentado é utilizado como modelo para inibição de retorno baseada em objeto do mapa de saliência. As regiões salientes obtidas pelo algoritmo exposto acima são apresentadas ao módulo de reconhecimento. O sistema de reconhecimento utilizado é baseado em um modelo hierárquico para reconhecimento de objetos HMAX [Fukushima, 1980].

(a) (b) (c)



(d) (e)

Figura 3.1: Ilustração da escolha do mapa de conspicuidade mais importante para a saliência. As Figuras 3.1(a), 3.1(b) e 3.1(c) representam os mapas de conspicuidade. A Figura 3.1(d) representa a imagem original e a Figura 3.1(e) o mapa de saliência.

Foram realizados experimentos com o intuito de avaliar qualitativamente a abordagem de segmentação de regiões salientes utilizando-se atenção visual. O método apresenta bons resultados para uma quantidade expressiva de imagens. Na maioria dos casos, as regiões selecionadas constituem, de fato, objetos ou partes de objetos. No entanto, o método apresenta problemas quando as regiões salientes dos objetos não são uniformes.

O desempenho apresentado pelo classificador HMAX não foi satisfatório para cenas naturais. Por isso, foi utilizado, também, um conjunto de imagens obtidas artificialmente. Para objetos claramente separados, o desempenho de reconhecimento usando atenção visual atingiu quase 100%, enquanto que o sistema sem utilizar atenção visual reconheceu apenas 50% dos objetos.

O sistema proposto por Walther et al [Walther et al., 2002] apesar de apresentar bons resultados para imagens artificiais, apresenta limitações quanto a robustez. Se a região saliente apresentar descontinuidades o sistema não consegue extrair corretamente o objeto saliente. A Figura 3.2 ilustra esse problema.

(a) (b)

Figura 3.2: Exemplo de imagem cuja segmentação da região saliente impõe dificuldades ao algoritmo.

Um mecanismo para localização e reconhecimento de placas de sinalização que utiliza atenção visual *bottom-up* foi proposto por Rodrigues [Rodrigues,]. Este sistema utiliza atenção visual como um meio de agilizar o processo de localização das regiões contendo placas de sinalização. Estas regiões são, em seguida, aplicadas a um sistema de redes neurais para que possam ser classificadas.

O sistema é composto por dois módulos principais: um módulo de detecção e um módulo de reconhecimento. O módulo de detecção é uma implementação simplicada do sistema de atenção visual *bottom-up* proposto por Itti et al. [Itti et al., 1998]. Enquanto que o módulo de reconhecimento é um classificador neural previamente treinado para reconhecer placas de sinalização de trânsito.

O módulo de detecção utiliza três características primitivas: cor, intensidade e orientação. Para cada característica são criadas Pirâmides Gaussianas e Pirâmides Direcionais que passaram por um processo de diferenças centro-vizinhaças (*center-surround differences*) para gerar mapas de conspicuidades. Esses mapas de conspicuidades são normalizados e somados para gerar o mapa de saliência final.

O classificador neural utiliza aprendizagem supervisionada. Ou seja, este classificador é treinado com um conjunto de imagens de placas de sinalização selecionadas por um ser humano. O tipo de rede neural utilizado é a Rede Neural MLP-BP (Multilayer Perceptron com algoritmo de treinamento Backpropagation).

Considerando-se que a rede neural tenha sido previamente treinada, o sistema funciona

da seguinte forma. Dada uma imagem de entrada, o sistema de atenção visual detecta as regiões mais importantes da imagem. Estas regiões passam por um pré-processamento (equalização de histograma e filtragem com *blur gaussiano*). Em seguida, as regiões salientes são apresentadas ao sistema de reconhecimento que as classifica de acordo com as classes que foram definidas durante o treinamento.

Foram realizados dois tipos de experimentos. O primeiro avaliou a acurácia do sistema de detecção na tarefa de selecionar regiões contendo placas de sinalização. O segundo tipo avaliou o desempenho do sistema de reconhecimento em classificar regiões obtidas pelo sistema de detecção. Para a construção da base de dados utilizada nos experimentos, foram extraídas imagens de um vídeo filmado a partir de um veículo em movimento durante uma viagem em dia claro entre duas cidades. Após a aquisição, o vídeo foi particionado em quadros e cada um deu origem a uma imagem colorida de resolução 352×240 *pixels*.

Para os experimentos com o Módulo de Detecção, foi selecionado um subconjunto de imagens a partir da base de imagens extraídas do vídeo. Apenas imagens com placas foram selecionadas, num total de 15 imagens com 16 placas, sendo 14 imagens com uma placa e uma com duas. Em todos os experimentos o raio de inibição foi fixado. Para o experimento em que o raio de inibição era de 20 *pixels* e foram utilizados 5 regiões, o sistema de detecção conseguiu localizar $93{,}75\%$ das placas.

Os experimentos de classificação obtiveram uma taxa média de reconhecimento de $84{,}40\%$. A melhor taxa foi de 100% (utilizando 11, 12 e 13 padrões de treinamento) e a taxa mais baixa foi de $56{,}41\%$ (utilizando 3 padrões de treinamento). Os experimentos mostraram que o número de padrões de treinamento tem um papel muito importante na tarefa de classificação.

O sistema proposto apresenta uma aplicação prática para a atenção visual. Os experimentos realizados obtiveram bons resultados, para alguns casos os resultados foram o máximo possível. No entanto, a forma de avaliar a complexidade do módulo de detecção apresenta algumas inconsistências. O autor afirma que se a quantidade de regiões salientes (com raio de inibição 20) utilizada é igual a 5, são utilizados $0{,}0059\%$ dos pontos de uma imagem 352×240. Porém, a quantidade de pontos realmente utilizada corresponde a multiplicação do número de regiões pela a área de cada região. Neste caso, a porcentagem de pontos utilizada seria $2{,}3674\%$ e não $0{,}0059\%$. Apesar disso, o resultado continua sendo muito bom.

Santos [Santos, 2005] propôs um mecanismo de atenção visual que integra mecanismos *bottom-up*, temporal e de profundidade para gerar um mapa de saliências em que as regiões mais importantes destacam objetos que despertam a atenção devido à influência tanto de características visuais primitivas quanto do movimento do mesmos. Ou seja, neste sistema, um objeto terá um valor alto de saliência se estiver em movimento, possuir uma distância menor que um valor *d* da câmera e apresentar alta saliência *bottom-up*.

O sistema considera a possibilidade de *n* câmeras que capturam de diferentes posições um mesmo vídeo a ser processado. O processamento de um vídeo consiste em gerar um novo vídeo em cujos quadros somente sejam visíveis as características *bottom-up* dos objetos móveis que estejam situados a, no máximo, uma distância *d* (pré-estabelecida) das câmeras. As demais regiões dos quadros são preenchidas com intensidades nulas.

Dois mapas intermediários são gerados durante o processamento. Um mapa de movimento, obtido do processamento dos quadros nos instantes t e $t + 1$ e um mapa de profundidade obtido do processamento dos n quadros no instante t. Utilizando os mapas de movimento e profundidade o quadro no instante t é segmentado. O quadro segmentado pelo movimento e pela profundidade é submetido ao módulo responsável pela atenção visual *bottom-up*.

O módulo de atenção visual *bottom-up* segue a arquitetura proposta por Itti et al. [Itti et al., 1998]. No entanto, o extrator de características implementado por Santos [Santos, 2005] recebe um quadro segmentado pelo movimento, ou seja, uma imagem em que apenas as regiões com algum nível de movimento são destacadas. Isso agiliza a extração das características *bottom-up*, pois o extrator só necessita trabalhar sobre as regiões não-nulas da imagem, que constituem uma pequena minoria.

Foram realizados experimentos com módulos separados (módulo de Atenção Temporal e módulo de segmentação de movimento) e com o sistema final. Para o módulo de atenção temporal foram realizados experimentos desde os protótipos iniciais, estes experimentos mostram a evolução de tal módulo. Além disso, também foi realizado um estudo de caso utilizando Atenção Temporal na detecção de transições em vídeo. Nenhum dos experimentos realizados ocorreu em tempo real.

Os experimentos que mostram a evolução do módulo de Atenção Temporal evidenciam o trabalho de implementação realizado no sentido de minimizar o ruído presente nos mapas de

movimento. Estes ruídos foram minimizados aplicando-se mudanças no algoritmo proposto por Wildes [Wildes, 1998], mais especificamente no que diz respeito à normalização de valores.

Além disso, foi realizado um estudo de caso da aplicação de Atenção Temporal na detecção de transições abruptas em vídeo. As estatísticas dos experimentos com o conjunto de treinamento mostram uma taxa relativamente baixa de falsas rejeições e taxa nula de falsas afirmações. Para o conjunto de teste as taxas foram mais altas do que as apresentadas pelo conjunto de treinamento, tanto para as falsas rejeições quanto para as falsas afirmações. No entanto, as taxas apresentadas pelo conjunto de teste são satisfatórias e promissoras.

Para o cálculo do desempenho dos experimentos de segmentação de objetos móveis, foi realizado uma contagem de *pixels*. Este cálculo mostrou que a redução da quantidade de *pixels* a serem analisados (o percentual de *pixels* completamente escuros) ultrapassa 97%. Por fim, foram realizados experimentos integrando atenção visual *bottom-up* e Atenção Temporal.

Os experimentos realizados separadamente com cada módulo apresentaram ótimos resultados, alguns com taxas de até 97%. No entanto, não são apresentadas estatísticas para os resultados obtidos pelo sistema global (todos os módulos integrados). Estas estatísticas poderiam ser, por exemplo, a porcentagem de regiões corretamente classificadas como estando em movimento.

3.3 Utilização de Algoritmos Genéticos como Métodos de Otimização em Sistemas de Visão Computacional

Bebis et al. [Bebis et al., 1999] propuseram um método para aplicar algoritmos genéticos na busca pela face de pessoas em imagens. Este problema foi dividido em duas partes: detecção de regiões contendo faces e comparação das regiões candidatas com a face a ser encontrada. Tanto a detecção de regiões contendo faces quanto o casamento das faces detectadas com a face buscada são realizados utilizando *eigenfaces* [Turk and Pentland, 1991].

Antes que as *eigenfaces* sejam calculadas, cada imagem passa por um pré-processamento que envolve normalização e equalização de histograma. Após o pré-processamento os *eigenspaces* são calculados e, para melhorar a detecção de faces, as características das faces

são salientadas pelo cálculo do gradiente de cada imagem utilizando um operador de Sobel.

Nos esquemas de codificação, cada indivíduo da população representa uma subjanela dentro da imagem de entrada. Foram utilizados dois esquemas. O esquema 1 é usado se o *aspect ratio* (razão entre a largura e altura) da imagem de entrada for maior que o *aspect ratio* das imagens do conjunto de treinamento, do contrário, o esquema 2 é utilizado. Um indivíduo codificado com o esquema 1 possui as coordenadas do canto superior esquerdo e a coordenada y do canto inferior direito da janela. Um indivíduo codificado usando o esquema 2 possui as coordenadas do canto superior esquerdo e a coordenada x do canto inferior direito. Em ambos os casos a coordenada que falta é calculada utilizando o *aspect ratio* da janela.

Os indivíduos de cada geração são avaliados utilizando o cálculo da distância do espaço de faces (*distance from face spaces* - dffs). Desta forma, uma imagem é considerada como face se o erro quadrático médio entre sua representação utilizando os auto-valores mais importantes e a imagem normalizada é pequeno. A função de aptidão possui dois termos: um para detecção e outro para verificação da face. O termo de detecção é calculado utilizando o *eigenspace* construído com faces de diferentes pessoas. O termo de verificação é calculado utilizando o *eigenspace* construído com várias imagens da face da pessoa pesquisada.

A aptidão de cada indivíduo é calculada com a seguinte equação:

$$aptid = MAX - dffs_{detec} - dffs_{verif} \tag{3.2}$$

em que MAX é um valor constante muito alto. Durante a evolução, o algoritmo genético procura maximizar esta função de aptidão.

Nos experimentos, foram utilizados dois conjuntos de treinamento. O primeiro possuia 38 imagens e foi utilizado para calcular o termo de detecção. O segundo continha 20 imagen do indivíduo a ser pesquisado e foi utilizado para calcular o termo de verificação. O algoritmo genético utilizou recombinação simples de dois pontos e mutação de um ponto. A probabilidade de recombinação foi $0{,}95$, de mutação $0{,}05$ e a constante MAX foi 18000.

O algoritmo foi testado em 10 cenas. Em média, foram necessárias 40 gerações para o algoritmo genético encontrar a face de interesse. Em todas as cenas, a face de interesse foi localizada corretamente. Os experimentos mostraram que o algoritmo genético reduziu o espaço de busca em dezenas de milhares de vezes em relação a uma busca exaustiva. Apesar

de apresentar taxa de detecção e reconhecimento de 100%, o artigo não apresenta os tempos necessários para realizar o processamento.

Um sistema que utiliza algoritmos genéticos para selecionar conjuntos de características mais relevantes em um processo de detecção de objetos foi proposto por Sun et al [Sun et al., 2003]. Como estudo de caso, eles realizaram extração de características utilizando um método *Principal Component Analysis* (PCA) [Jolliffe, 2002] e máquinas de vetores de suporte (SVM) [Vapnik, 1995] como classificador.

Cada imagem é representada como um conjunto de auto-valores. Embora muitos auto-valores sejam importantes para propósitos de reconhecimento, eles também podem confundir o classificador em outras aplicações, tal como a detecção. Desta forma, Sun et al [Sun et al., 2003] utilizaram algoritmos genéticos para selecionar um bom subconjunto de auto-valores a fim de aumentar o desempenho do sistema de detecção de objetos.

O método proposto pode ser dividido em quatro passos:

- extração de auto-valores utilizando PCA;
- seleção de subconjuntos de auto-valores utilizando algoritmos genéticos;
- treinamento das SVMs;
- classificação de novas imagens.

Cada imagem é representada como um vetor de auto-valores. Neste esquema de codificação, o cromossomo é uma cadeia de bits cujo comprimento é determinado pela quantidade de auto-valores. A função objetivo é modelada de modo a minimizar a quantidade de características necessárias para que se obtenha o melhor desempenho. Portanto, a avaliação da aptidão possui dois termos: precisão e quantidade de características utilizadas.

Em geral, a população inicial é gerada aleatoriamente. Como não há informação suficiente que determine se há dependência entre as características nos cromossomos, utiliza-se o cruzamento uniforme. A mutação é um operador de probabilidade muito baixa e apenas muda um bit específico.

Foram realizados experimentos com duas classes de imagens: imagens de automóveis e imagens de faces. As imagens de automóveis são proprietárias e foram extraídas manualmente de fotografias obtidas pelos autores. Dessas imagens, 1051 contêm veículos e 1051

não contêm veículos. As imagens de faces foram extraídas manualmente do CMU *face detection dataset* [Sim et al., 2003]. Para os experimentos com faces, foram utilizadas 616 imagens de faces e 616 imagens que não continham faces.

Para propósito de comparação, também foi implementado o método de seleção de características SFBS (*Sequential Floating Backward Selection*). O SFBS é uma versão do método *plus l - take away r* que primeiro enlarguece o subconjunto de características por l características utilizando seleção para frente e depois remove r características utilizando seleção para trás. O número médio de características selecionadas pelo SFBS em imagens de veículos foi 87, enquanto o método proposto selecionou 46 características. Com as imagens de faces, o SFBS selecionou 68 características e o algoritmo genético 34.

O método proposto por Sun et al [Sun et al., 2003] apresenta aplicações práticas e simples do uso de algoritmos genéticos (detecção de faces e de automóveis). No entanto, a comparação dos resultados com outros métodos existentes poderia ter sido realizada utilizando mais de um método de seleção de características e não apenas o SFBS. Além disso, os autores não deixam claro o esforço computacional aplicado no processo de otimização do algoritmo genético.

Um método adaptativo que utiliza algoritmos genéticos associados a um mecanismo de atenção visual para localizar olhos em imagens de faces foi proposto por Huang e Wechsler [Huang and Wechsler, 1999; Huang and Wechsler, 2000]. Este método procura, inicalmente, por regiões salientes e, em seguida, as classifica. O mapa de saliência é obtido utilizando consenso entre rotinas de navegação codificadas como um autômato de estados finitos (FSA - *Finite State Automaton*) que explora a imagem de face e evolui utilizando algoritmos genéticos.

A abordagem adaptativa de localização de olhos primeiro busca onde os objetos salientes estão e, em seguida, os classifica. Especificamente, esta abordagem envolve: geração do mapa de atenção e possível classificação de regiões como regiões contendo olhos. A etapa de classificação realiza uma seleção ótima de características e a criação de uma árvore de decisão (DT - *Decision Tree*) para confirmação da classificação de olhos utilizando algoritmos genéticos.

O mapa de saliência é obtido a partir das seguintes tarefas: extração de características, derivação dos mapas de conspicuidade e integração das saídas das várias rotinas visuais. O

modelo computacional correspondente envolve a média, o desvio padrão e a entropia como mapas de características. Um FSA evoluído por meio de AGs gera os mapas de características, enquanto que métodos de consenso os integram em um mapa de saliências final. A Figura 3.3 ilustra a criação do mapa de saliência.

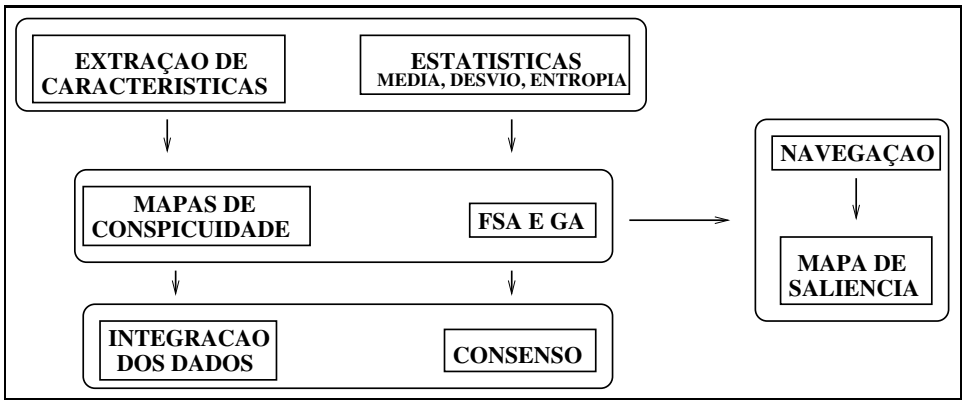


Figura 3.3: Criação do mapa de saliência.

Após a obtenção do mapa de saliência, o componente de reconhecimento deve decidir se as regiões mais salientes correspondem a regiões de olhos. O classificador implementado como uma árvore de decisão e algortimos genéticos utiliza o desempenho obtido pela árvore de decisão como guia para a obtenção de um subconjunto de características ótimas e para melhorar a aptidão média das populações de árvores de decisão.

Os experimentos foram realizados utilizando 10 imagens de faces para treinar 20 FSAs de olhos esquerdos e 20 FSAs de olhos direitos para derivar o mapa de saliência. Foram necessárias 2000 gerações do AG para que o FSA obtivesse um desempenho de 100% no reconhecimento de olhos. A abordagem proposta por Huang e Wechsler [Huang and Wechsler, 1999; Huang and Wechsler, 2000] apresentou bons resultados. No entanto, o artigo não descreve a técnica utilizada para extrair a localização dos olhos das imagens testadas para efeito de comparação com os resultados obtidos pelo sistema.

3.4 Considerações Finais

Neste capítulo, foram analisados trabalhos que tratam do uso de atenção visual como meio para agilizar o processo de detecção de objetos em imagens e do uso de algoritmos genéticos

como técnica de otimização para sistemas que utilizam atenção visual *bottom-up*. Os trabalhos de Santos [Santos, 2005] e Rodrigues [Rodrigues,] foram desenvolvidos por alunos do mesmo grupo de pesquisa que o autor desta dissertação. Inclusive, todos utilizam o mesmo sistema de atenção visual *bottom-up* com adaptações para cada caso.

Observa-se que nenhum destes trabalhos utiliza algoritmos genéticos para ponderar mapas de características em sistemas de atenção visual *bottom-up* como o sistema proposto neste trabalho, que é discutido no Capítulo 2, o faz. Além disso, esses trabalhos utilizam conjuntos de imagens muito pequenos e apresentam uma análise estatística muito restrita dos resultados.

O próximo capítulo apresenta o sistema proposto nesta dissertação. Este sistema utiliza algoritmos genéticos para otimizar pesos que são utilizados para ponderar os diversos mapas de características utilizados para formar mapas de saliências em sistemas de atenção visual *bottom-up*.

Capítulo 4

Sistema Proposto

Este capítulo apresenta o sistema proposto, sua arquitetura, implementação e a descrição de cada um de seus módulos. Adicionalmente, o capítulo também traz alguns aspectos da implementação referentes à escolha de uma biblioteca para implementação de algoritmos genéticos e a utilização de uma grade computacional para acelerar o processo de otimização dos algoritmos genéticos.

4.1 Arquitetura

Esta dissertação propõe uma nova estratégia para otimização de pesos de um mecanismo de atenção visual baseado em características. Esta estratégia utiliza algoritmos genéticos para otimizar um arranjo de pesos para os mapas que compõem o mapa de saliências de forma que o mapa de saliências resultante apresente melhores resultados quando comparado com resultados previamente otimizados. A estratégia proposta aplica pesos não somente aos mapas de características mas sim a todos os mapas que compõem os mapas de saliências. A otimização é seguida por fases de detecção e comparação. Na fase de detecção, um mapa saliente baseado em três características (cor, intensidade e orientação) é construido. Após a fase de detecção, as regiões salientes são comparadas com algumas regiões selecionadas manualmente em uma etapa anterior e os resultados da comparação são usados como função de avaliação do algoritmo genético para produzir as próximas gerações do processo de otimização.

O sistema de otimização é formado por três módulos: verificação de regiões salientes,

atenção visual e otimização de pesos. O módulo de atenção visual é baseado no que foi proposto em [Itti et al., 1998]. A verificação de regiões salientes é realizada utilizando regiões selecionadas manualmente e é usada para validação dos experimentos. O módulo de otimização de pesos é utilizado para gerar populações com pesos apropriados para a ponderação dos mapas. A Figura 4.1 ilustra a arquitetura do sistema de otimização.

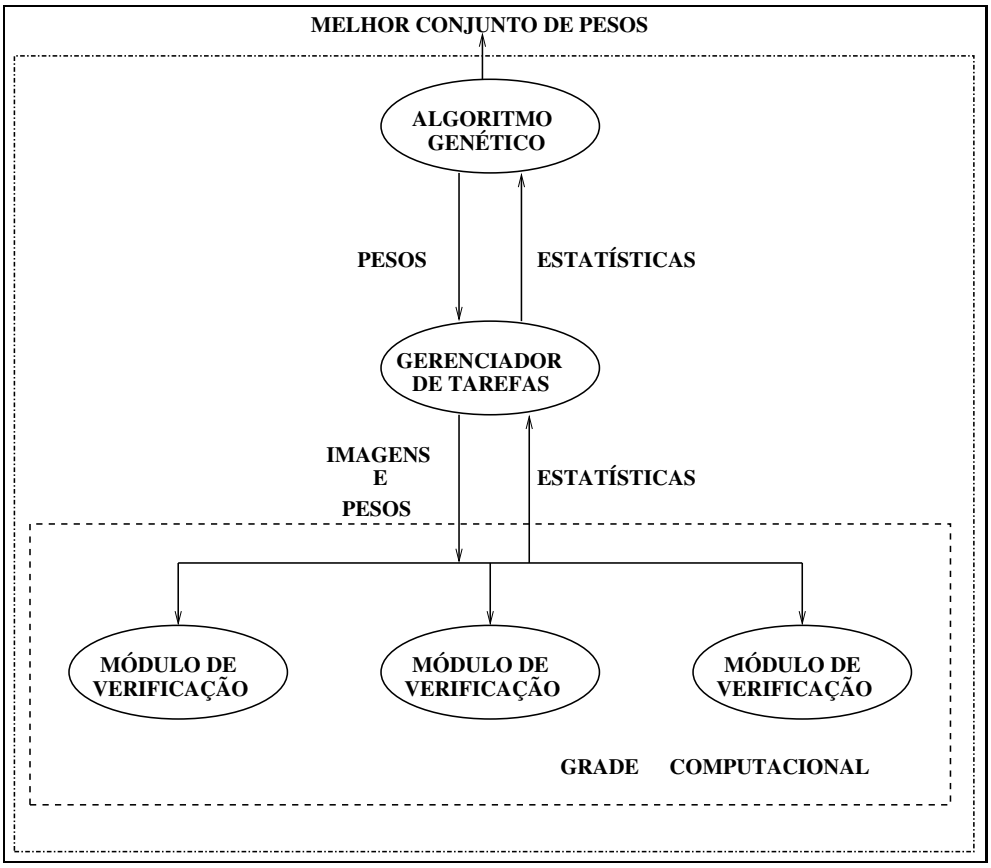


Figura 4.1: Arquitetura do sistema.

Em linhas gerais, o sistema descrito na Figura 4.1 funciona da seguinte forma. Após a seleção manual das regiões de interesse de um conjunto de imagens utilizadas para otimização, o módulo de otimização de pesos envia à grade o conjunto de imagens, bem como as coordenadas das regiões selecionadas, um conjunto de pesos e os programas executáveis que compõem o módulo de atenção visual. A grade computacional irá gerenciar o envio dessas informações e o recebimento dos resultados obtidos pelo módulo de atenção visual. A cada iteração, o módulo de otimização de pesos avalia os resultados do módulo de atenção visual e envia um novo conjunto de pesos à grade até que a otimização seja finalizada. A grade computacional foi utilizada devido à necessidade de se processar uma quantidade grande de

imagens (100 para cada classe de região) em um número muito grande de iterações (cerca de 1600 iterações). Se este processamento fosse executado em apenas um computador, levaria cerca de um mês para ser executado. O processamento em grade reduziu o tempo de processamento para um ou dois dias dependendo da disponibilidade de computadores na grade. A seguir, a descrição de cada módulo é apresentada.

O módulo de atenção visual utilizado é uma adaptação do sistema proposto em [Itti et al., 1998]. Ele usa um mecanismo de atenção visual *bottom-up* de mapas de saliências. Este mecanismo é construído utilizando-se Pirâmides Gaussianas e operadores de vizinhança localmente orientados. A Figura 4.2 mostra um diagrama do módulo de atenção visual.

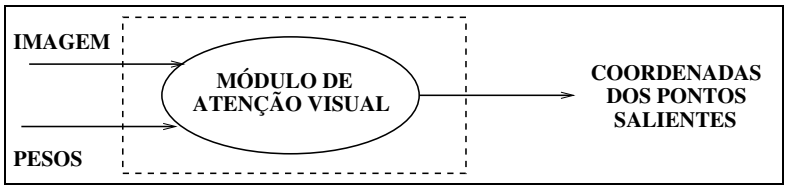


Figura 4.2: Módulo de atenção visual.

Os pesos são otimizados utilizando valores estatísticos obtidos pelo módulo de verificação de regiões. A Figura 4.3 ilustra resumidamente o módulo de verificação. Este módulo calcula a média de pontos necessária para que pelo menos um ponto saliente esteja presente em regiões selecionadas manualmente. O módulo de atenção visual recebe imagens e pesos, processa as imagens e envia as coordenadas dos pontos salientes para o módulo que calcula as estatísticas. O módulo que calcula as estatísticas recebe, também, as coordenadas das regiões selecionadas manualmente e dá como saída as médias e desvios-padrão das quantidades de pontos salientes presentes nas regiões selecionadas manualmente.

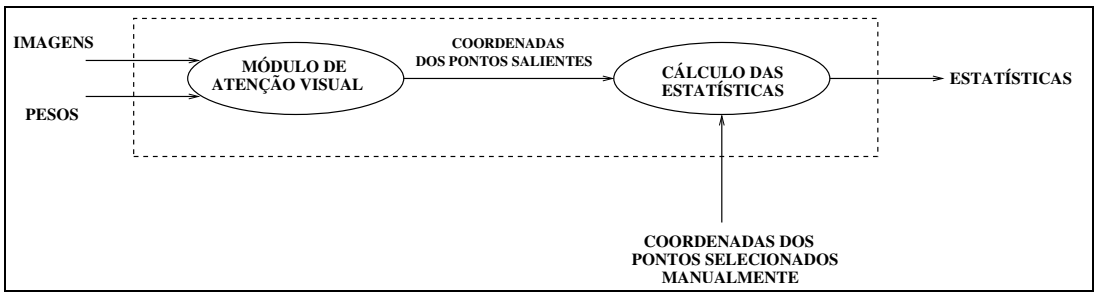


Figura 4.3: Módulo de verificação de regiões salientes.

4.2 Implementação do Sistema

Em um modelo de mapas de saliências, um conjunto de mapas é combinado para formar um único mapa que representa as regiões mais salientes na cena [Itti and Koch, 2000]. Uma região saliente é a região que mais atrai a atenção de um observador. Conforme discutido no Capítulo 2, Itti e Koch [Itti and Koch, 1999] compararam quatro estratégias de combinação de mapas de características: soma normalizada simples, combinação linear com pesos aprendidos, normalização global não-linear seguida por somatório e competição não-linear entre localizações salientes. Todas as estratégias comparadas por Itti e Koch [Itti and Koch, 1999] combinam os mapas de características utilizando processos de aprendizagem para ponderar os mapas. Porém, nenhum deles utiliza um processo de otimização com algoritmos genéticos como o apresentado neste trabalho.

Algoritmos genéticos pertencem à classe de técnicas de otimização global, se caracterizando por encontrar ótimos globais da função sendo otimizada, enquanto que os processos de aprendizagem mais populares (como redes neurais), tendem a utilizar métodos de otimização local, os quais possuem o risco de encerrar o processo de aprendizagem em mínimos locais da função sendo aprendida. Assim, pode-se considerar essa uma das vantagens do uso de algoritmos genéticos sobre aprendizagem no problema em questão. Contudo, algoritmos genéticos possuem alguns problemas como a necessidade de grande poder de processamento devido à grande quantidade de iterações necessárias durante cada evolução.

Como para cada mapa de saliência pode-se associar um peso, conhecimento de alto nível pode ser utilizado para guiar os tipos de regiões selecionadas [Itti et al., 2005]. Por exemplo, se alguém procura por flores vermelhas em uma imagem de jardim, a precisão da busca pode ser melhorada se os pesos relacionados a cores tiverem valores mais altos do que os pesos relacionados às outras características. É este tipo de conhecimento que o sistema aqui proposto utiliza para melhorar a qualidade da busca por regiões salientes e guiar a atenção para objetos semelhantes a objetos previamente conhecidos via um processo de otimização.

4.2.1 Módulo de Verificação de Regiões Salientes

Como o sistema aqui descrito necessita de uma etapa de otimização, é necessário que a seleção de imagens que contenham características semelhantes seja realizada. Nessa dissertação

essa seleção é realizado manualmente. O processo de otimização requer que as regiões mais importantes das imagens tenham sido indicadas pelo usuário. Isto é necessário devido ao fato de que o sistema irá otimizar os pesos que serão atribuídos aos mapas de características de acordo com as regiões que foram indicadas na etapa de seleção manual.

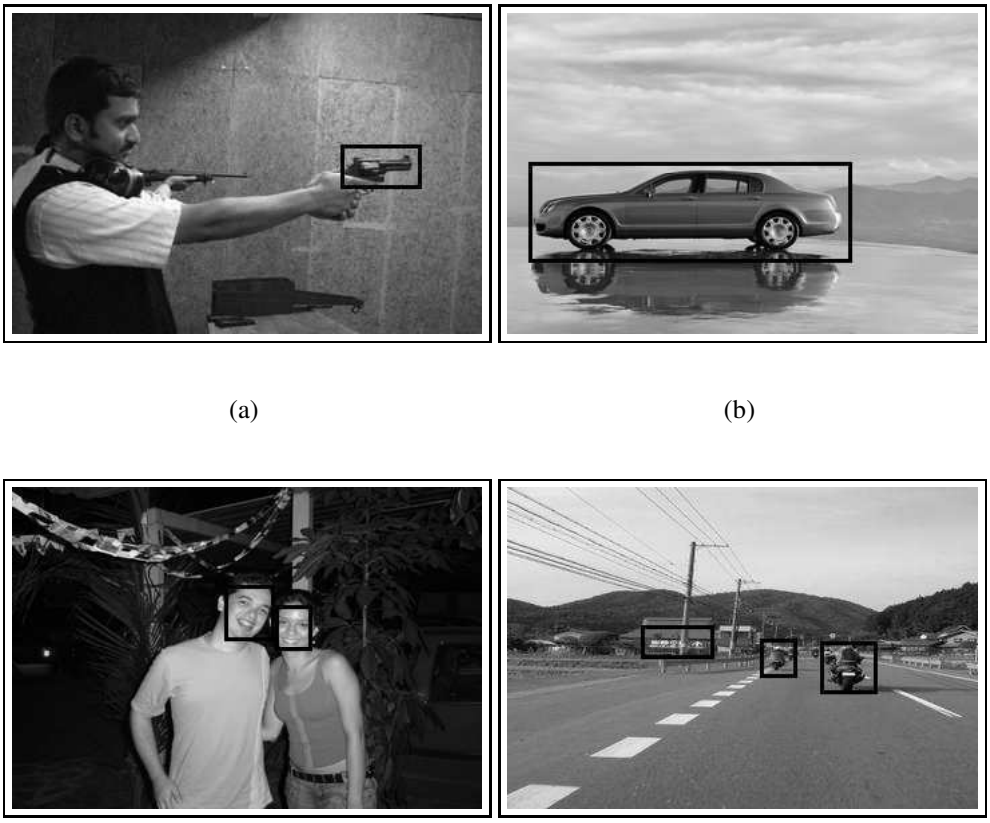
Foram realizadas quatro otimizações de pesos, uma para cada classe de região selecionada manualmente. Estas classes são: faces de pessoas, objetos genéricos, armas (pistolas ou revólveres) e carros. Para o conjunto de imagens contendo pessoas, selecionam-se manualmente as regiões das faces guardando-se as coordenadas dos retângulos que as contém. O mesmo processo é realizado para as imagens em que o assunto a ser selecionado são objetos genéricos. Para os casos de armas e carros, as regiões selecionadas são os menores retângulos que contém tais objetos. Em todos os casos, as regiões selecionadas manualmente são aquelas que despertam a atenção do observador com base em características primitivas como cor, intensidade e orientação, mas tendo em mente a classe de regiões definida previamente. Na Figura 4.4 temos exemplos de imagens utilizadas na etapa de otimização e as regiões selecionadas.

Estas regiões selecionadas manualmente são utilizadas pelo módulo de verificação de regiões salientes. O módulo de verificação calcula estatísticas sobre a presença de pontos salientes nas regiões selecionadas manualmente. O cálculo é feito como descrito a seguir. Ao aplicar o conjunto de pesos a uma imagem, calcula-se a quantidade mínima de pontos para que 100% das áreas de interesse sejam atingidas por no mínimo 1 ponto saliente. Após aplicar esse conjunto de pesos a todas as imagens, calculam-se a média e o desvio-padrão. Em seguida, essas medidas são enviadas ao módulo de otimização que utiliza estes valores para minimizar a quantidade média de pontos presentes nas regiões de interesse.

4.2.2 Módulo de Atenção Visual

A Figura 4.5 apresenta detalhes da implementação do módulo de atenção visual. O sistema de atenção visual tem como entrada a imagem a ser processada e o conjunto de pesos utilizado para ponderar os mapas. O resultado desse processamento são as coordenadas dos pontos da imagem ordenados por valor de saliência, do mais saliente para o menos saliente.

Antes da soma dos mapas de conspicuidades e dos mapas de características, tais mapas são ponderados utilizando-se os pesos obtidos pelo algoritmo genético. Cada combinação de



(a) (b)

(c) (d)

Figura 4.4: Exemplos de imagens utilizadas na otimização. Os retângulos indicam as regiões de interesse selecionadas manualmente.

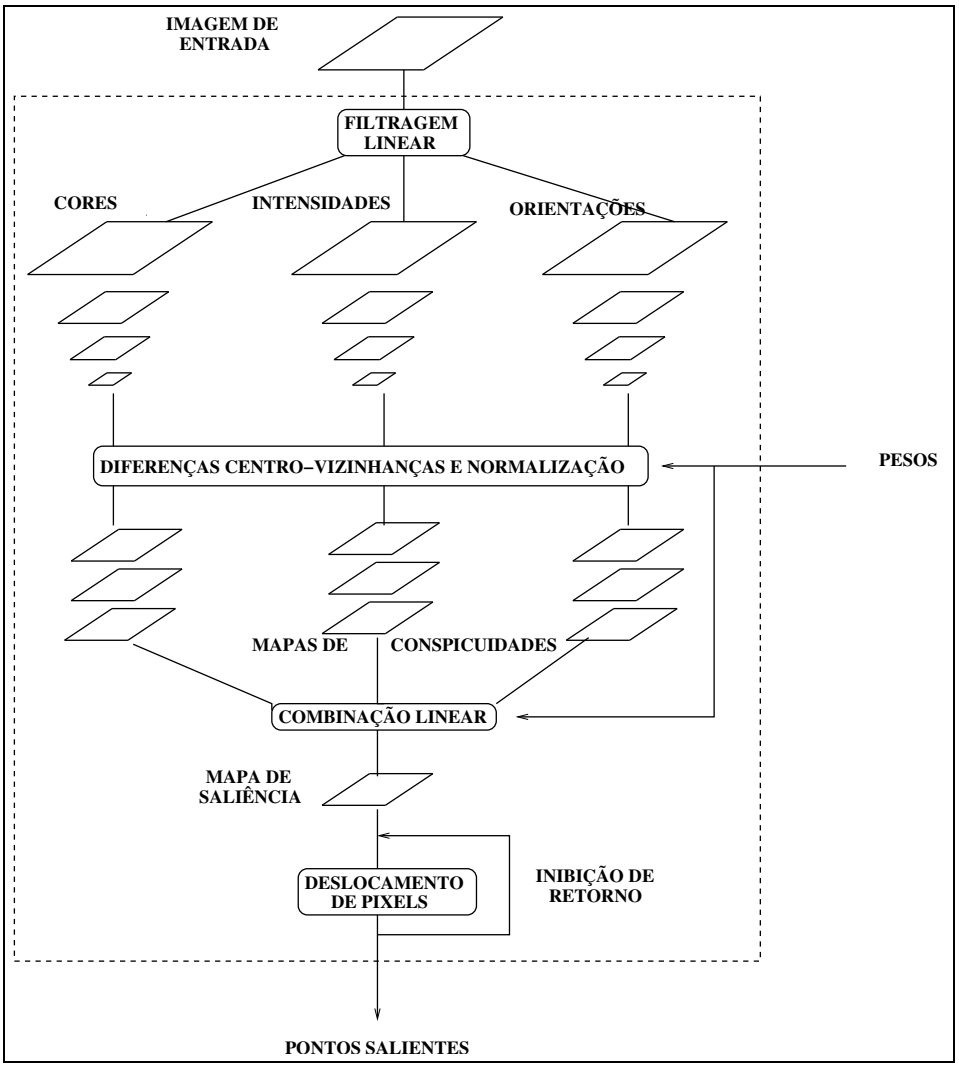


Figura 4.5: Ilustração do módulo de atenção visual.

pesos é aplicada a todas as imagens e os resultados são escritos em arquivo. Esses pesos são aplicados aos mapas de características e de conspicuidades a fim de verificar qual a melhor combinação para detectar as regiões salientes da imagem.

Foi implementada uma estratégia de ordenação de *pixels* para selecionar as regiões de interesse. É selecionada uma região ao redor da coordenada de interesse (que corresponde ao *pixel* com maior valor) no mapa de saliências. Além de selecionar a região de interesse, esta região é preenchida com valores de intensidade nula, posteriormente. Isso previne que a mesma região de interesse seja tratada mais de uma vez, correspondendo a uma variante simplificada para o mecanismo de inibição de retorno proposto no trabalho de Itti et al. [Itti et al., 1998]

Para prevenir que somente partes de objetos sejam tratadas, uma estratégia de movimentos sacádicos foi implementada. Para cada região de interesse, são implementados deslocamentos que mudam o foco de atenção para vários pontos vizinhos. Os focos de atenção são determinados deslocando-se as coordenadas do ponto de atenção 5 e 10 *pixels* em uma vizinhança de 8 *pixels*, gerando 16 variações de pontos de atenção.

Siagian e outros [Siagian and Ititi, 2004] empregaram a computação da média dos mapas para obtenção do mapa de saliências final, e para realizar a tarefa visual requerida usando este mapa (exemplo: localizar placas de trânsito ou localizar faces). Nesta dissertação, contudo, o sistema de atenção visual é ajustado pela mudança do conjunto de pesos que são usados para para produzir o mapa de atenção final, de forma que a tarefa visual seja melhor realizada. Estes pesos são obtidos por um processo experimental. Neste processo, pesos diferentes são atribuidos a cada mapa e os resultados são otimizados por um algoritmo genético. Na próxima subseção, o módulo responsável pela geração dos pesos que são atribuídos aos mapas é descrito.

4.2.3 Módulo de Otimização de Pesos

Após revisão bibliográfica sobre técnicas de ponderação de mapas de características para geração de mapas de saliências, optou-se por gerar os pesos utilizando-se algoritmos genéticos. O principal motivo para tal escolha é que não foi encontrado nenhum trabalho que mencionasse o uso de algoritmos genéticos para ponderação de mapas de características, e, portanto, decidiu-se verificar a viabilidade de seu uso no problema em questão.

Como o sistema que implementa algoritmos genéticos foi implementado na linguagem de programação *C++*, procuramos por uma biblioteca que se adequasse a esse requisito. Além disso, outro fator pelo qual algumas partes do sistema foram implementadas em *C++* é a questão do desempenho, visto que operações de processamento de imagens e evolução de algoritmos genéticos exigem um alto desempenho. Porém, alguns módulos do sistema foram implementados utilizando *Java*. A linguagem *Java* foi utilizada nos sistemas de comunicação com o *grid* e no sistema de seleção manual das regiões de interesse. Usou-se *Java* para comunicação com o *grid* devido às facilidades disponibilizadas em *Java*, tais como: bibliotecas e interfaces para visualização de *jobs* e *tasks*.

Os conjuntos de pesos aplicados ao sistema de atenção visual durante a otimização foram

mapeados em cromossomos da seguinte forma: cada peso do conjunto corresponde a um gene do cromossomo e um cromossomo é representado por um conjunto de pesos. Neste mapeamento não utilizamos cadeias de bits, ao invés disso, cada elemento da cadeia que compõe o cromossomo é um número entre 1 e 100. Este modo de construir os cromossomos foi utilizado para permitir um mapeamento mais claro entre os pesos e os elementos do algoritmo genético. No entanto, cada elemento da cadeia é transformado em uma cadeia de bits para que as mutações sejam realizadas.

Devido a necessidade de normalizar os valores dos pesos aplicados aos mapas, para cada mapa utilizado para formar o mapa de saliência foi atribuido aleatoriamente um valor entre 1 e 100 dividido pela soma dos valores de todos os mapas. Portanto, um cromossomo representa um conjunto de pesos e cada peso é representado por um gene. Como são necessários 27 mapas (3 de características, 4 de cor, 4 de intensidades e 16 de orientação) para formar um mapa de saliência, cada cromossomo contém 27 pesos cujos valores iniciais são obtidos aleatoriamente e em seguida são normalizados dividindo-se cada um pela soma de todos.

O tipo de algoritmo genético implementado evolui utilizando sobreposição de populações. A partir de uma porcentagem previamente estabelecida, o algoritmo cria uma nova população de uma porcentagem dos melhores indivíduos da população anterior e de uma porcentagem dos cruzamentos e das mutações da população anterior. A aptidão dos indivíduos para evolução é mensurada pela média de pontos presentes nas regiões de interesse que cada indivíduo (conjunto de pesos) obteve. Se a média de pontos é alta, o indivíduo é descartado e não estará presente na próxima geração, mas se a média é baixa há uma grande probabilidade do indivíduo evoluir para a próxima geração ou participar de cruzamentos e mutações com outros indivíduos aptos a evoluir.

Após a obtenção de um conjunto de pesos, enviam-se os pacotes contendo o material que será utilizado no processamento nas máquinas remotas. Cada pacote contém os programas de atenção visual e de cálculos estatísticos, os arquivos contendo as coordenadas dos retângulos que circunscrevem as regiões selecionadas manualmente e dez imagens em dois formatos diferentes (PPM, PGM). Os pesos são enviados por meio das tarefas do OurGrid e irão servir como parâmetros para o programa de atenção visual.

4.2.4 Biblioteca para Implementação de Algoritmos Genéticos

Neste trabalho, utilizamos uma biblioteca para criação de algoritmos genéticos, a biblioteca *GAlib* (*http://lancet.mit.edu/ga/*). A *GAlib* é uma biblioteca construída em *C++* por *Matthew Wall* no *Massachusetts Institute of Technology*. Além disso, a *GAlib* é gratuita e distribuída sob uma licensa estilo BSD (*BSD-style license*). Alguns fatores que influenciaram na escolha da *GAlib*: ela ser construída em *C++*, ser gratuita e dispor de uma boa documentação.

Esta biblioteca possui duas classes principais, uma representa genomas e a outra representa um tipo de algoritmo genético. Cada instância de genoma representa uma solução única para determinado problema. O objeto algoritmo genético define como a evolução deverá ocorrer. O algoritmo genético utiliza uma função objetivo definida pelo usuário que determina quão apto cada genoma está para sobreviver. Há também operadores de genoma e estratégias de seleção para gerar novos indivíduos.

Para utilizar esta biblioteca o usuário deve definir três coisas:

- uma representação;
- os operadores genéticos;
- a função objetivo.

A GAlib provê mecanismos para gerar de forma rápida e prática operadores e representações. Porém, o programador é totalmente responsável pela função objetivo. Uma vez que o programador tenha uma representação, os operadores e uma maneira de medir o objetivo da otimização, ele poderá aplicar as funções pré-definidas do GAlib para implementar seu sistema.

Há muitos tipos de algoritmos genéticos. A GAlib provê três tipos básicos: *simple*, *steady-state* e *incremental*. Estes algoritmos diferem no modo de criação de novos indivíduos e na forma como os indivíduos antigos serão substituídos durante a evolução. A GAlib provê dois mecanismos de extensão das capacidades dos objetos pré-definidos. Primeiro, o programador pode derivar suas próprias classes e definir novas funções membro. Se o programador necessita apenas de pequenos ajustes no comportamento de uma classe da GAlib, em muitos casos, ele pode definir uma única função e informar à classe da GAlib para usar a

nova função ao invés da padrão. Abaixo, há um trecho de código de um programa utilizando as classes da GAlib.

Código 4.1: Trecho de código exemplificando o uso da GAlib

```
float Objective(GAGenome&);
main(){
  // cria um genoma
  GA2DBinaryStringGenome genome(width, height, Objective);
  // cria o AG
  GASimpleGA ga(genome);
  // evolui o AG
  ga.evolve();
}
```

4.3 Descrição sobre o Uso do OurGrid

Detectamos a necessidade de utilização de alguma forma de paralelismo para execução do sistema, justificada pelos seguintes fatores:

- o processamento de uma imagem pelo módulo de atenção visual leva, em média, 20 segundos para ser executado em um computador com 512MB de memória RAM e 1GHZ de clock;
- havia um número muito grande de imagens para avaliar em cada geração do algoritmo genético (foram processadas 380 imagens);
- os parâmetros estabelecidos para o algoritmo genético determinavam o uso de 80 gerações de 40 indivíduos;

Considerando os fatores acima, seriam necessários mais de 100 dias para realizar a otimização utilizando apenas um computador. Das formas de paralelismo conhecidas (supercomputadores, *clusters* e grades computacionais, por exemplo), optou-se pela utilização de uma grade computacional pelos seguintes motivos: não havia a necessidade de comunicação entre os processos em execução em computadores diferentes, possibilidade de execução em

ambientes heterogêneos e baixo custo operacional. Utilizando-se uma grade computacional, o tempo de processamento foi reduzido para pouco mais de 24 horas. A grade computacional utilizada foi o OurGrid (*http://ourgrid.org*). O OurGrid é uma grade ponto-a-ponto (*peer-to-peer*) a qual qualquer pessoa pode se juntar e obter acesso. Ele tem sido desenvolvido desde dezembro de 2004. Em 8 de março de 2007 o OurGrid contava com 206 computadores conectados à grade.

O sistema de comunicação com o OurGrid foi acoplado ao módulo de otimização de pesos utilizando-se funções de chamada ao sistema de *C++* e de *Java*. Isto foi necessário devido ao fato de que tais módulos foram implementados em linguagens de programação distintas: o módulo de otimização de pesos em *C++* e o sistema de comunicação com o OurGrid em *Java*. Detalhes sobre o funcionamento do sistema de comunicação com o OurGrid são expostos a seguir.

Para cada indivíduo da população de genes é criado um *job* contendo uma quantidade de tarefas que está relacionada a quantidade de imagens que se deseja processar em cada computador remoto. Por exemplo, se há 100 imagens e se deseja processar cinco imagens em cada computador remoto, criam-se 20 tarefas. Observa-se que, desta forma, o número de tarefas também indica a quantidade de computadores remotos que serão utilizados para o processamento dos *jobs*. A escolha da quantidade ideal de tarefas foi realizada fazendo-se alguns testes na grade utilizando-se poucas imagens e observando-se a quantidade de computadores disponíveis no período em que o processamento ia ser iniciado. Além disso, como, às vezes, algumas tarefas falhavam ou algum computador remoto que estava disponível passava a ser usado causando a perda da tarefa, buscou-se estimar uma quantidade de tarefas que minimizasse o tempo de processamento levando-se em consideração tanto a quantidade de imagens que estariam sendo processadas paralelamente quanto a probabilidade da tarefa falhar durante o processamento.

O OurGrid provê dois mecanismos que facilitam o acompanhamento das tarefas que estão sendo processadas: uma interface gráfica (mygrid gui) e uma página de status. A interface gráfica mostra o estado de todas as tarefas bem como em quais computadores da grade as tarefas estão sendo executadas. A página de status (*http://status.ourgrid.org/*) exibe todos os *peers* que estão online e seus computadores disponíveis. As Figuras 4.6 e 4.7 mostram telas da interface gráfica e da página de status.

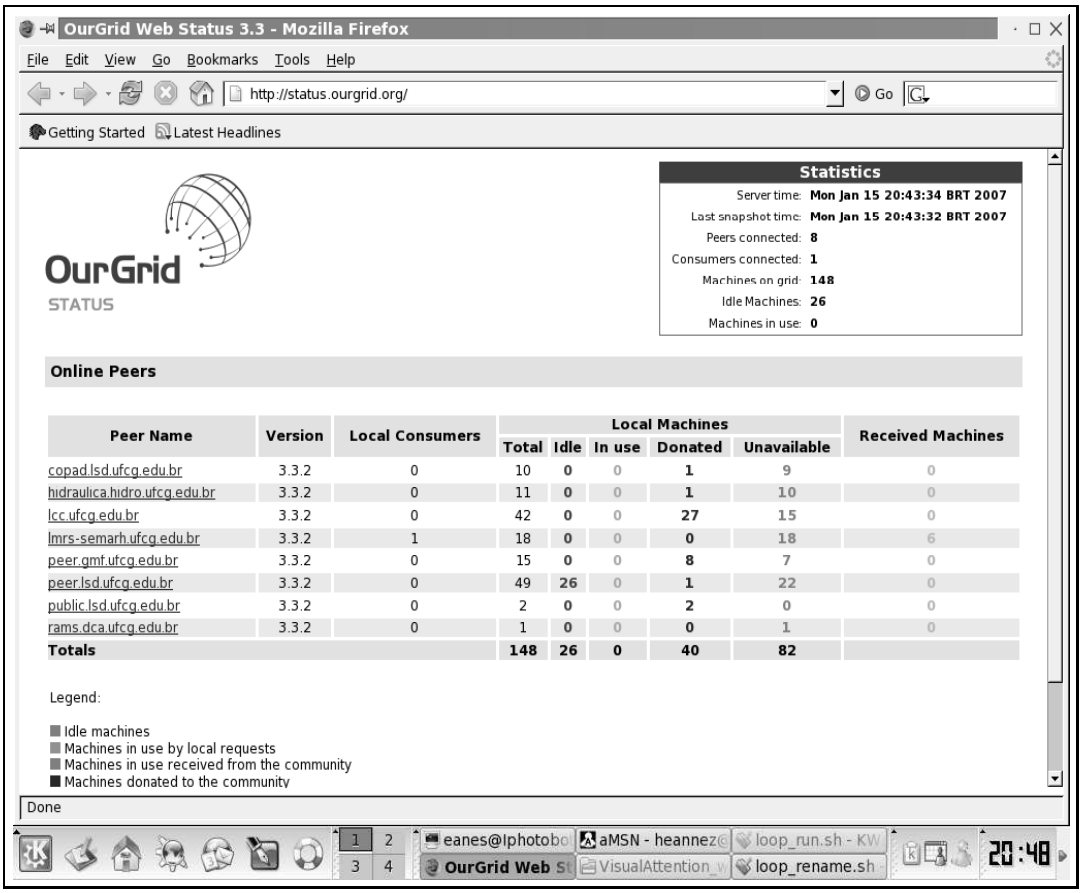


Figura 4.6: Página de status do OurGrid.

Cada *job* contém a descrição das tarefas que devem ser executadas nos computadores remotos bem como algumas exigências que os computadores remotos devem *cumprir* para que possam ser usados. No caso do sistema aqui exposto, os computadores remotos deveriam estar executando o sistema linux. Há outro requisito necessário pelo sistema que iria rodar remotamente mas que não podia ser especificado no início do programa: ter o programa de compactação *tar* instalado. A verificação da inexistência desse programa em computadores remotos era feita pela observação dos motivos de falha em algumas tarefas e pelo fato de que algumas tarefas sempre falhavam nos mesmos computadores. Quando se observava falha persistente de tarefas em determinados computadores ou em computadores sob certo domínio, alterava-se os requisitos dos *jobs* adicionando-se uma entrada que indicava ao gerenciador de *jobs* para não usar tais máquinas no processamento.

A especificação de cada tarefa contém a quantidade de imagens a serem processadas, o comando para descompactar o pacote contendo um conjunto de imagens e os executáveis, o

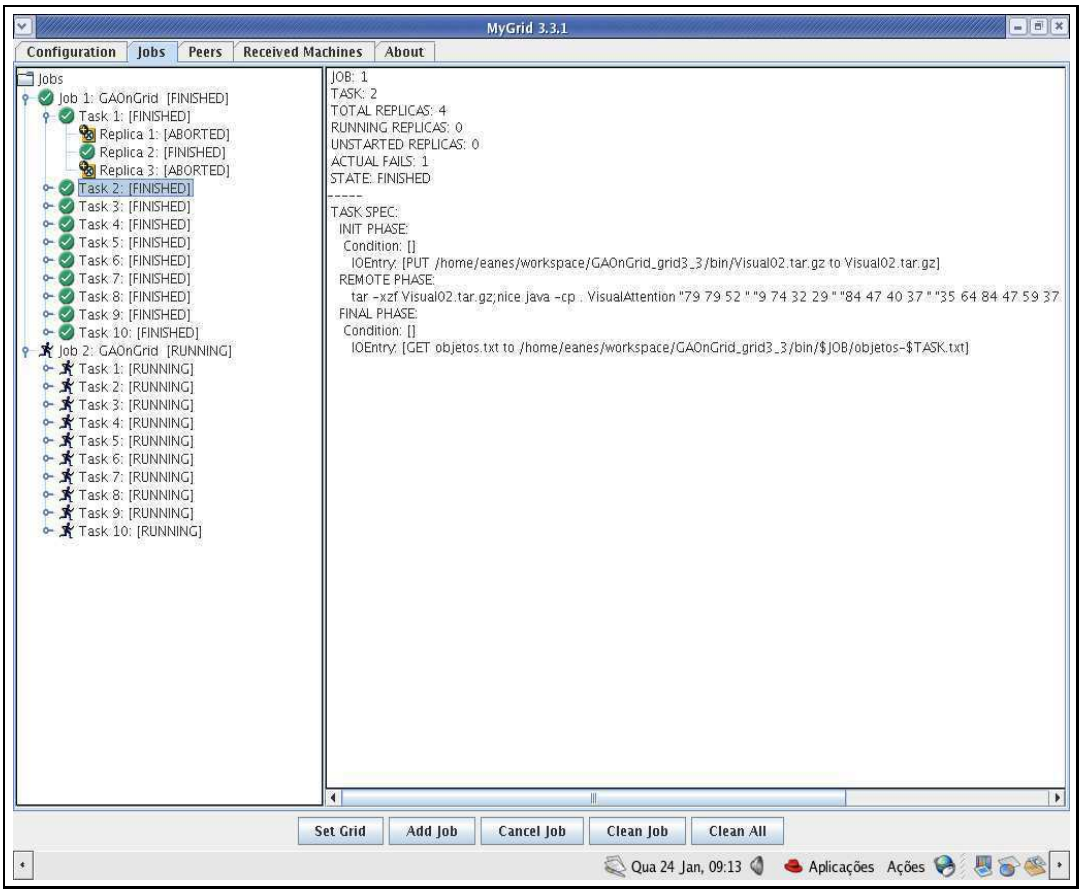


Figura 4.7: Interface gráfica do MyGrid em execução.

comando java responsável por executar o programa que gerencia o sistema de atenção visual e os cálculos estatísticos, e o comando para enviar os resultados para o computador onde está sendo executado o algoritmo genético.

4.4 Considerações Finais

Este capítulo apresentou a arquitetura do sistema e a descrição dos seus módulos. Esta arquitetura utiliza algoritmos genéticos para otimizar um arranjo de pesos para os mapas de características de forma que o mapa de saliências resultante apresente melhores resultados quando comparado com resultados previamente aprendidos. Nesta arquitetura, a otimização é seguida por fases de detecção e comparação. Na fase de detecção, é construído um mapa de saliência baseado em três características (cor, intensidade e orientação). Após a fase de detecção, as regiões salientes são comparadas com algumas regiões selecionadas manual-

mente em uma etapa anterior e os resultados da comparação são usados na próxima etapa de otimização.

No próximo capítulo, os experimentos realizados são relatados e analisados. Estes experimentos foram executados com quatro classes de imagens e os resultados obtidos pelo sistema aqui exposto são comparados com os resultados obtidos utilizando o sistema de Itti [Itti et al., 1998].

Capítulo 5

Resultados Experimentais

Este capítulo apresenta os experimentos realizados com o intuito de comparar sistemas de atenção visual *bottom-up* sem otimização dos pesos dos mapas de características com o sistema de atenção visual proposto, no qual os pesos dos diversos mapas que compõem o mapa de saliência são otimizados via algoritmo genético. Além disso, os problemas de gerenciamento de memória enfrentados com os experimentos no OurGrid são descritos.

5.1 Detalhes sobre a Obtenção das Imagens e Otimização dos Pesos

O principal propósito dos experimentos foi otimizar os pesos de modo que o sistema pudesse encontrar o assunto de interesse nas imagens utilizando a menor quantidade possível de pontos. Os experimentos foram realizados utilizando quatro tipos de imagens: imagens com pessoas, imagens de objetos genéricos, imagens de carros e imagens de pistolas. No conjunto de imagens que continha pessoas, as regiões das faces foram selecionadas como sendo as regiões mais atrativas. Nos outros conjuntos de imagens, as regiões contendo as classes de objetos definidas anteriormente foram selecionadas. O objetivo de usar diferentes tipos de imagens é verificar a capacidade de generalização do algoritmo genético.

Neste trabalho, foram utilizados computação em grade e algoritmos genéticos. A grade computacional *OurGrid (http://ourgrid.org/)* foi utilizada para processar remotamente e paralalemente o módulo de atenção visual. A biblioteca de algoritmos genéticos *GAlib*

(http://lancet.mit.edu/ga/) foi utilizada para implementar o algoritmo de otimização de pesos dos mapas de características e conspicuidades. Quase todo o código foi implementado utilizando a linguagem de programação *C++*, as excessões são o sistema de seleção manual e os métodos relacionados à comunicação com a grade os quais foram implementados utilizando a linguagem *Java*.

5.1.1 Obtenção de Imagens

No total, foram utilizadas 380 imagens nos processos de otimização: 100 de pessoas, 100 de carros, 100 de armas e 80 de objetos genéricos. Para teste, foram utilizadas 400 imagens, 100 para cada classe. Estas imagens foram obtidas por meio de *download* da Internet. Para executar esta tarefa, foram utilizados os seguintes programas: *FlickrDown*, *GoogleGrab* e *NeoDownloader*. O *FlickrDown* (*http://greggman.com/pages/flickrdown.htm*) é específico para obtenção de imagens do sítio *www.flickr.com*. O *GoogleGrab* (*http://www.sas21.de/apps/webimagegrab/*) automatiza o processo de *download* de imagens do sítio *http://images.google.com.br*. O *NeoDownloader* (*http://www.neowise.com/neodownloader/*) é um *webcrawler* que busca por imagens a partir de um sítio dado como entrada e de todos os seus *links*.

O sítio *www.flickr.com* é um sítio para armazenamento, pesquisa e organização de fotografias onde qualquer pessoa pode armazenar suas imagens. Ele utiliza um sistema simples, porém útil, para facilitar o agrupamento das imagens. Este sistema de rotulamento (*tags*) permite ao usuário que está fazendo *upload* rotular suas imagens de acordo com seu conteúdo. No entanto, este sítio não oferece nenhuma forma prática para *download* de um conjunto de imagens com determinado rótulo. O *FlickrDown* é um programa criado para solucionar este problema. Ele permite que o usuário faça *download* de até 500 imagens de determinado rótulo.

Problema semelhante é enfrentado quando alguém deseja fazer *download* de várias imagens retornadas pela busca por imagens do *Google*. O *Google* não fornece nenhuma ferramenta que permita o *download* automático de várias imagens retornadas por sua busca. O *GoogleGrab* é um programa que permite que este *download* seja realizado. No entanto, há limitações quanto a quantidade de imagens baixadas.

Uma maneira menos restrita, porém com uma probabilidade mais alta de obter imagens

irrelevantes (baixa resolução e conteúdo não fotográfico, por exemplo) da Internet, é utilizar um *webcrawler*. Um *webcrawler* é uma ferramenta que provê um meio rápido de encontrar recursos na Internet através da manutenção de um índice da Web que pode ser consultado sobre documentos de um assunto específico [Pinkerton, 1994]. Neste trabalho utilizamos o *webcrawler NeoDownloader* para obter parte das imagens. Os parâmetros básicos que ele necessita como entrada são: o sítio inicial, os tipos de imagens que devem ser obtidas, o tamanho mínimo de imagem e uma palavra, ou conjunto de palavras, chave. Ele busca por todo o sítio e segue os links presentes neste para outros sítios.

5.1.2 Otimização dos Pesos

O algoritmo genético utiliza sobreposição de populações. Utilizando uma porcentagem previamente estabelecida, o algoritmo cria uma nova população de uma porcentagem dos melhores indivíduos da população anterior e de uma porcentagem das recombinações e das mutações da população anterior. O objetivo do algoritmo é determinar a melhor média de pontos necessária para encontrar todas as regiões previamente selecionadas manualmente.

Após os pesos otimizados serem encontrados, um processo de verificação foi realizado. Este processo foi realizado utilizando-se para cada classe 100 imagens que não faziam parte dos conjuntos utilizados para otimização. Estas imagens foram obtidas de sítios da Internet. No conjunto de imagens com pessoas, as regiões das faces de pessoas foram manualmente selecionadas. Nos outros conjuntos de imagens, as regiões selecionadas foram aquelas que despertam a atenção segundo algumas características primitivas (cor, intensidade e orientação). Em seguida, foi realizada uma verificação para se obter a quantidade de pontos salientes contidos nas regiões selecionadas manualmente. A próxima seção detalha a evolução de cada algoritmo genético.

5.2 Processo de Otimização

Esta seção apresenta uma discussão sobre a evolução dos algoritmos genéticos de cada classe de imagens. Além de apresentar gráficos das melhores médias de pontos presentes nas regiões selecionadas manualmente, esta seção também apresenta os melhores conjuntos de

pesos, os parâmetros de entrada dos algoritmos genéticos e uma análise de quais características são mais importantes para cada classe de imagens.

5.2.1 Determinação dos Parâmetros para os Algoritmos Genéticos

Antes de iniciar os processos de otimização, foram realizados experimentos com o intuito de identificar quais os valores mais apropriados para a probabilidade de mutação. Para estes experimentos o valor de probabilidade de recombinação foi fixado em 60%. Foram realizados dez experimentos para cada conjunto de imagens, utilizando amostras de 20 imagens para cada conjunto. Em cada experimento, foram testados valores de mutação no intervalo de 1 a 10%, com passo de 1%. Estes experimentos mostraram que, para o problema aqui tratado, o valor de mutação mais apropriado para ser utilizado com uma probabilidade de recombinação de 60% é de 1%. O Apêndice B mostra os gráficos das evoluções dos algoritmos genéticos para cada valor de mutação.

5.2.2 Imagens de Objetos Genéricos

A Figura C.1 mostra a evolução do algoritmo genético para imagens contendo objetos genéricos. Nesta figura, podemos ver os valores das médias dos melhores indivíduos de cada geração. Para este experimento, os valores atribuídos para mutação, recombinação e substituição foram: $0{,}01$, $0{,}6$ e $0{,}5$ respectivamente. Cada geração continha 80 indivíduos e o algoritmo genético deveria evoluir até 40 gerações. No entanto, o algoritmo genético parou de evoluir na vigésima primeira geração. Isto ocorreu porque a curva de otimização estabilizou-se. O indivíduo que obteve a melhor média ($577{,}33$) de pontos foi o indivíduo 34 da vigésima primeira geração.

O indivíduo que obteve melhor média de pontos continha o conjunto de pesos mostrado na Tabela 5.1. Nesta tabela, a primeira coluna contém os três pesos que serão aplicados aos mapas de conspicuidades que formam o mapa de saliência, a segunda coluna contém os pesos para os mapas de intensidades, a terceira os pesos para os mapas de cores e a quarta os pesos para os mapas de orientação. Os pesos da primeira coluna correspondem a intensidade, cor e orientação, respectivamente. Os pesos dos mapas de intensidades correspondem às intensidades dos canais vermelho, verde, azul e amarelo, nesta ordem. Os pesos dos mapas

de orientação podem ser agrupados de quatro em quatro, de forma que cada grupo de quatro orientações corresponde a variações de orientação para uma mesma escala (as orientações são: $0°$, $45°$, $90°$ e $135°$). Esta mesma explicação vale para as Tabelas 5.2, 5.3 e 5.4). Da análise deste conjunto de pesos podemos observar que a característica mais importante para guiar a atenção no mapa de saliência para objetos genéricos é a orientação. E para os 4 níveis das pirâmides de orientação, as orientações mais importantes são $45°$ e $135°$ (segundos e quartos valores de cada grupo de quatro orientações), nesta ordem.

Saliência	Intensidades	Cores	Orientações
4 11 96	44 86 4 16	13 47 28 86	5 89 19 76 3 96 4 10 13 39 1 24 5 89 3 66

Tabela 5.1: Pesos para objetos genéricos.

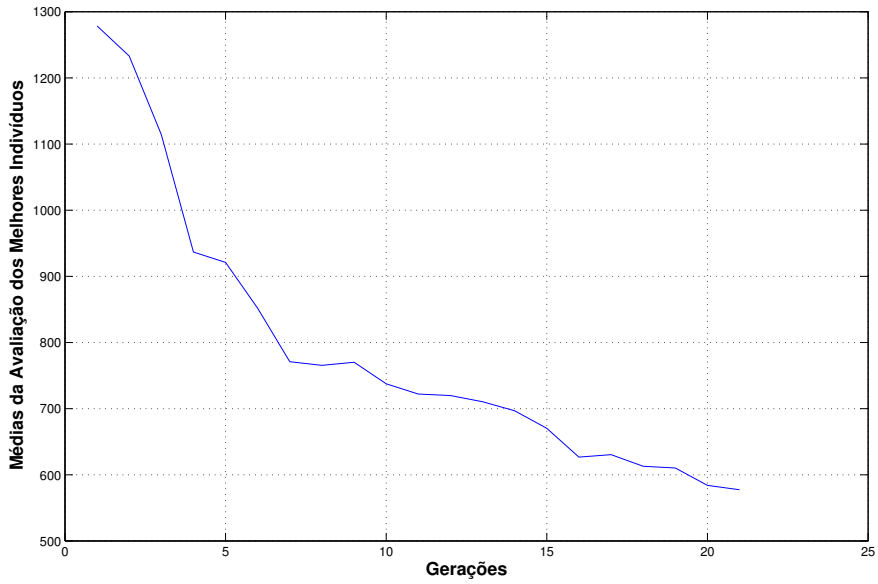


Figura 5.1: Melhores médias de cada geração para imagens contendo objetos genéricos.

5.2.3 Imagens Contendo Faces de Pessoas

As melhores médias de cada geração na evolução do algoritmo genético para imagens contendo faces de pessoas é mostrada na Figura C.2. Nesta figura, podemos observar que o melhor cojunto de pesos foi obtido por um indivíduo da quadragésima geração, que obteve

uma média de $40{,}29$ pontos presentes nas regiões selecionadas manualmente. Os valores das probabilidades de mutação, recombinação e substituição para este experimento foram respectivamente $0{,}01$, $0{,}7$ e $0{,}5$. O algoritmo genético deveria evoluir até 80 gerações de 40 indivíduos, mas a evolução estabilizou-se a partir da geração 40 e a otimização parou na geração 41. A Tabela 5.2 mostra o conjunto de pesos que obteve a melhor média de pontos salientes em regiões selecionadas manualmente. A partir desses valores podemos perceber que a característica mais importante para despertar a atenção do observador em regiões de faces (em um mapa de saliências que utiliza cor, intensidade e orientação) é a orientação. Pelo conjunto de pesos para os mapas de orientação podemos perceber que a orientação $45°$ é a mais importante para os três níveis mais altos da pirâmide. No entanto, para o quarto nível, a orientação mais importante é $90°$.

Saliência	Intensidades	Cores	Orientações
1 3 65	33 39 93 46	37 79 10 60	2 90 23 7 62 99 3 58 1 86 1 15 4 32 39 10

Tabela 5.2: Pesos para imagens de pessoas.

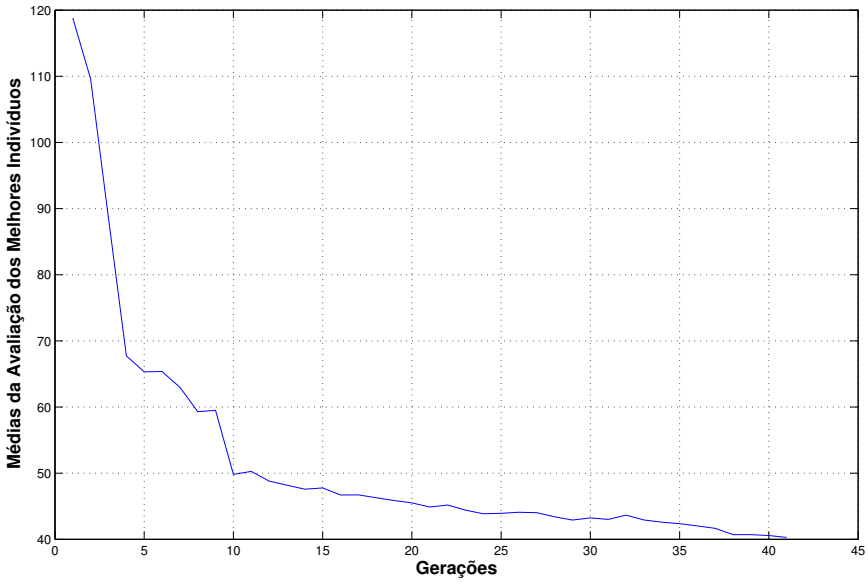


Figura 5.2: Melhores médias de cada geração para imagens de pessoas.

5.2.4 Imagens de Armas

No gráfico da Figura C.3 podemos observar os melhores resultados de cada geração na otimização de pesos para imagens de armas (pistolas ou revólveres). Os valores das probabilidades de mutação, recombinação e substituição foram: $0{,}1$, $0{,}6$ e $0{,}5$, nessa ordem. As gerações continham 40 indivíduos e deveriam evoluir até 80 gerações. Porém, a curva de otimização estabilizou-se na quadragésima primeira geração. A melhor média de pontos $(30{,}05)$ foi obtida pelo indivíduo 50 da décima oitava geração. O conjunto de pesos deste indivíduo é mostrado na Tabela 5.3. Analisando os pesos do mapa de saliência observa-se que os mapas de características para intensidade e cor têm pouca ou nenhuma influência para guiar a atenção para regiões onde há pistolas. Podemos observar que a orientação $45°$ é a mais importante em todos os níveis das pirâmides.

Saliência	Intensidades	Cores	Orientações
1 1 33	71 16 68 78	18 40 72 79	24 47 2 10 49 76 23 39 2 77 13 72 43 65 31 11

Tabela 5.3: Pesos para imagens de pistolas.

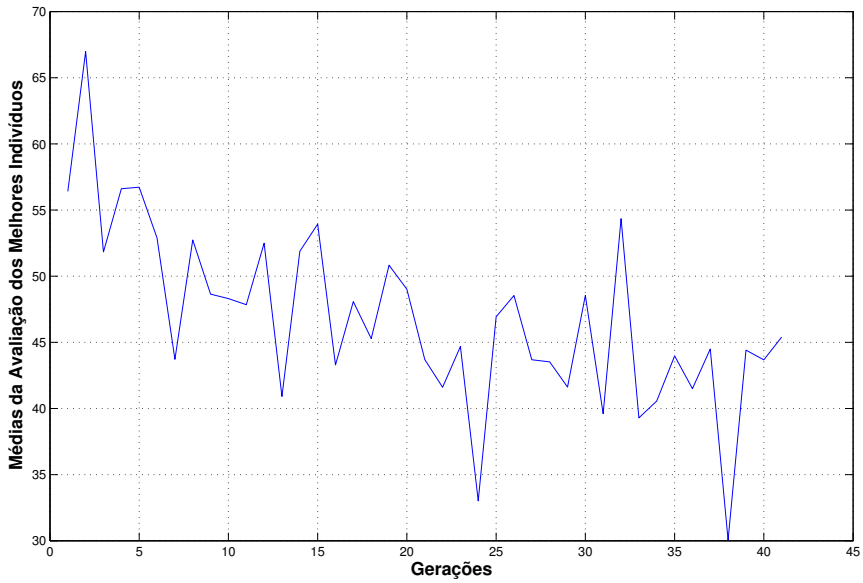


Figura 5.3: Melhores médias de cada geração para imagens contendo armas.

5.2.5 Imagens de Carros

Como nos outros casos, verificou-se que a orientação é a característica mais importante para guiar a atenção para regiões contendo carros. Isto pode ser observado no conjunto de pesos do indivíduo que obteve a melhor média de pontos $(4{,}15)$ na otimização para imagens contendo carros. Tal conjunto de pesos foi obtido pelo vigésimo indivíduo da quadragésima primeira geração. O algoritmo deveria evoluir até 80 gerações, mas a curva estabilizou-se a partir da geração 41. Seu conjunto de pesos é: "18 25 96" "80 32 26 4" "98 48 23 27" "58 97 6 51 5 72 2 12 3 51 4 82 16 96 1 38", como mostrado na Tabela 5.4. Com exceção do nível 2 das pirâmides, cujo peso mais importante está para orientação $135°$, todos os níveis têm como orientação mais importante $45°$. Para este experimento, os valores das probabilidades de mutação, recombinação e substituição foram: $0{,}01$, $0{,}7$ e $0{,}5$.

Saliência	Intensidades	Cores	Orientações
18 25 96	80 32 26 4	98 48 23 27	58 97 6 51 5 72 2 12 3 51 4 82 16 96 1 38

Tabela 5.4: Pesos para imagens de carros.

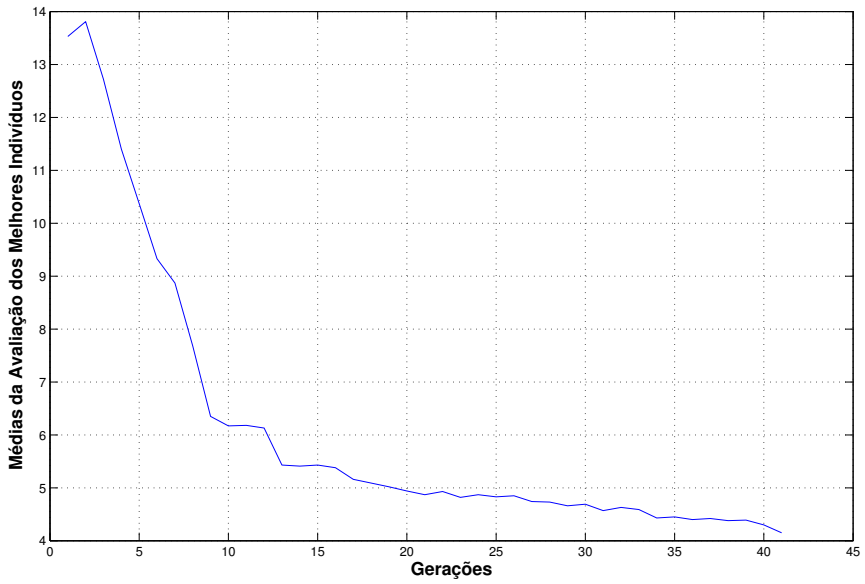


Figura 5.4: Melhores médias de cada geração para imagens contendo carros.

5.3 Descrição do Sistema Utilizado para Comparação

O sistema aqui proposto foi comparado com uma implementação do sistema de Itti et al [Itti et al., 1998]. O sistema *iLab Neuromorphic Vision C++ Toolkit* (iNVT, pronunciado "invent") é desenvolvido pelo laboratório iLab da *University of Southern California*. Esta implementação está disponível para download em *http://ilab.usc.edu/toolkit/*.

O iNVT é um conjunto completo de classes C++ para o desenvolvimento de modelos neuromórficos de visão. Modelos neuromórficos são algoritmos cuja arquitetura e função são inspiradas em cérebros biológicos. O *iLab Neuromorphic Vision C++ Toolkit* compreende não apenas classes base para imagens, neurônios e áreas cerebrais, mas também modelos tais como o modelo de atenção visual *bottom-up* e de surpresa Bayesiana[1].

As características fundamentais deste *toolkit* são:

- a principal plataforma de desenvolvimento é Linux;
- possui classes para processamento de baixo nível, tais como: Point2D, Rectangle, Range e Timer;
- possui funções de entrada/saída de imagens como leitura/escrita de arquivos de imagens (PNM ou PNG) ou *streams* de vídeo;
- modelos neuromórficos de atenção visual, integração de contorno e reconhecimento de objetos;
- ferramentas para processamento paralelo de modelos complexos em *clusters* de computadores.

5.3.1 Experimentos com o iNVT

Os experimentos foram realizados utilizando o módulo de atenção visual *bottom-up* do iNVT. Para estes experimentos o programa utilizado foi o *ezvision* e foram utilizadas 400 imagens (100 para cada classe de região saliente estudada). O *ezvision* foi executado por meio de um shell script que determinava a imagem e os parâmetros de entrada e salvava o

[1] Surpresa Bayesiana (*Bayesian Surprise*) quantifica como os dados afetam observadores naturais e artificiais através da medida de diferenças entre crenças posteriores e anteriores dos observadores.

log de saída em um determinado diretório. Este script determinava os seguintes parâmetros de entrada para o *ezvision*:

- *–output-frames=0-2500@EVENT* -+: indica a quantidade de pontos salientes, neste caso, 2500;
- *–salmap-iordecay=0*: coeficiente que indica o decaimento da inibição de retorno;
- *–out=pnm*: especifica um destino para os frames de saída;
- *–textlog=test.txt*: salva mensagens de log em arquivo. Estes arquivos contêm as coordenadas dos pontos de saliência para cada imagem;
- *–foa-radius=1*: raio do foco de atenção;
- *–in=nome_da_imagem*: imagem de entrada.

O código abaixo mostra o script utilizado para executar o *ezvision*. Como pode ser observado pelo parâmetro *–output-frames=0-2500@EVENT* -+, foram extraídos 2500 pontos salientes. Isto foi necessário porque, mesmo com a inibição de retorno ativada, ocorria redundância de pontos salientes. Para obter apenas 845 pontos (correspondentes a 1% do total de pontos de cada imagem) foi utilizado um programa para extrair os primeiros 845 pontos sem repetição. Estes pontos foram utilizados no processo de verificação das regiões salientes que será discutido na próxima seção.

```
#!/bin/bash
ls fotos_jpg/ > nomes_das_imagens.txt
function get_image_name {
        cat nomes_das_imagens.txt | head -n $1 | tail -n 1
                                  | cut -f1 -d'.';
}
for ((i=1; i<"101"; i++)); do
            IMG_NAME=$(get_image_name $i);
            IMG_NAME2=$IMG_NAME.jpg;
            IMG_NAME_TXT=$IMG_NAME.txt;
            echo "IMG_NAME=$IMG_NAME";
```

```
        ezvision --output-frames=0-2500@EVENT -+
        --salmap-iordecay=0 --out=pnm
        --textlog=testp.txt --foa-radius=1
        --in=Fotos_nao_treinadas/$IMG_NAME2
        mv testp.txt top2500_people/$IMG_NAME_TXT
done
```

5.4 Resultados da Verificação das Regiões Salientes

A verificação da presença dos pontos salientes nas regiões selecionadas manualmente foi executada para pontos obtidos por três sistemas: iNVT, atenção visual com otimização de pesos e atenção visual sem otimização de pesos. Os três sistemas foram executados para quatro conjuntos de imagens: imagens contendo faces de pessoas, imagens de objetos genéricos, imagens de carros e imagens de armas. Nas subseções seguintes, são apresentados os resultados obtidos.

5.4.1 Imagens Contendo Faces de Pessoas

O conjunto de imagens com pessoas usado no processo de verificação contém 194 faces de pessoas. A partir do gráfico da Figura C.7 pode-se obervar que, utilizando-se somente 1% do número total de pontos de cada imagem, o sistema que utiliza pesos otimizados encontrou pontos de interesse em 152 faces de pessoas previamente selecionadas. Para o mesmo conjunto, o sistema iNVT encontrou pontos salientes em 98 faces utilizando 1% do total de pontos da imagem. O resultado com pesos otimizados representa um ganho de 26% em relação ao sistema iNVT. Na Figura C.7, bem como em todas as figuras que mostram a comparação dos resultdos do sistema proposto com os resultados do sistema de iNVT, verifica-se que a curva dos resultados do sistema iNVT fica constante a partir de um certo valor de pontos por imagem (para imagens contendo faces de pessoas, $0{,}1\%$). Isso ocorre porque a partir desse valor o sistema não consegue sair da região, gerando um aglomerado de pontos muito próximos uns dos outros. A Figura 5.5 ilustra este fato, nela foram marcados 1% dos pontos da imagem.

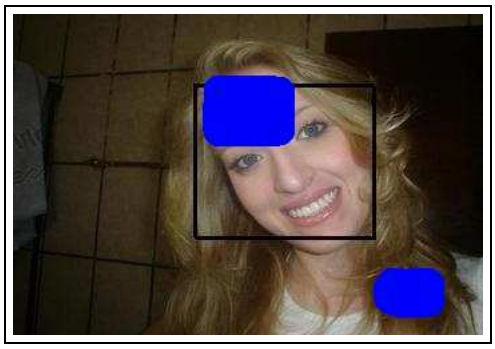


Figura 5.5: Marcação dos pontos salientes obtidos pelo sistema iNVT.

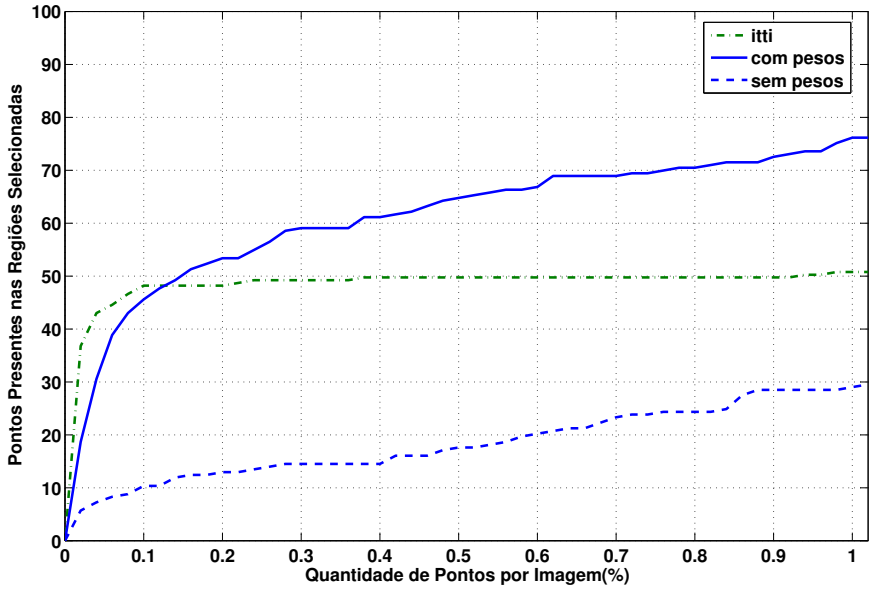


Figura 5.6: Comparação dos resultados para imagens contendo pessoas.

5.4.2 Imagens Contendo Objetos Genéricos

No conjunto de imagens contendo objetos genéricos, 258 objetos, ou regiões que despertam a atenção, foram manualmente selecionadas. O gráfico da Figura C.8 mostra que utilizando-se 1% do número total de pontos de cada imagem o sistema que utiliza pesos otimizados encontrou pontos de atenção em 222 objetos ou regiões, enquanto que o sistema iNVT encontrou pontos salientes em 189 objetos. Desta forma, a otimização de pesos para objetos genéricos incrementou em cerca de 9% a quantidade de regiões selecionadas atingidas por pontos salientes em relação ao sistema iNVT.

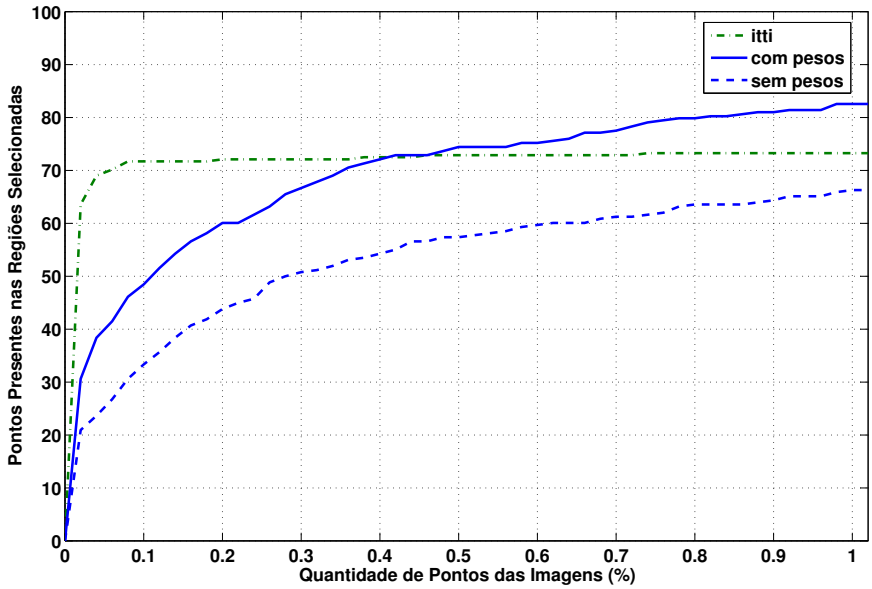


Figura 5.7: Comparação dos resultados para imagens contendo objetos genéricos.

5.4.3 Imagens Contendo Armas

As 100 imagens utilizadas para este experimento continham 104 armas. O gráfico da Figura 5.8 mostra que os três sistemas obtiveram altas taxas de acerto na localização das regiões selecionadas manualmente. Isto decorre do fato de que a maioria das imagens utilizadas apresentava as armas em *close-up*, ou seja, em algumas imagens as armas ocupavam uma grande área. No Apêndice D, há exemplos de imagens utilizadas. Apesar de os três sistemas apresentarem valores altos de acerto, o que deve ser levado em consideração é que mesmo assim o sistema de atenção visual proposto apresentou valores mais altos do que o iNVT. Utilizando 845 pontos o iNVT encontrou pelo menos um ponto saliente em 97 regiões e o sistema de atenção visual com otimização de pesos encontrou pontos salientes em 100 regiões.

5.4.4 Imagens Contendo Carros

Ocorreu um problema semelhante ao dos experimentos com imagens de armas com o experimento de imagens de carros. Devido à dificuldade de encontrar imagens apropriadas para os experimentos, algumas imagens utilizadas apresentavam carros ocupando uma grande área

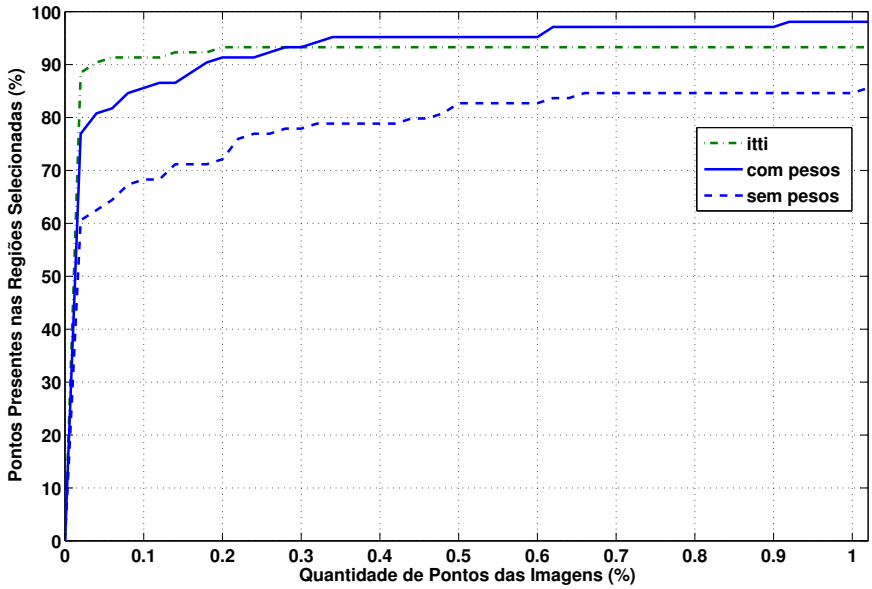


Figura 5.8: Comparação dos resultados para imagens contendo pistolas.

da imagem, como pode ser observado no Apêndice D. No conjunto de imagens de teste, foram selecionados 101 carros, destes, o iNVT acertou 100 e o sistema de atenção visual otimizado acertou todas as regiões, utilizando 845 pontos. Como no experimento com imagens de armas, o mais importante não é a alta taxa de acerto, mas a diferença entre as taxas do iNVT e do sistema proposto. Os resultados para este experimento são mostrados no gráfico da Figura 5.9

As Figuras C.7, C.8, 5.8 e 5.9 deixam evidente que o uso de pesos otimizados melhora a tarefa de encontrar o assunto das imagens. Além disso, os pesos otimizados guiam a detecção de assunto de modo que o usuário possa estabelecer previamente que tipo de objetos ele deseja que seja ressaltado no mapa de saliências.

5.5 Problemas Enfrentados com o Uso do OurGrid

No ínicio do processamento das tarefas na grade computacional, observou-se que quando a quantidade de tarefas chegava a um número expressivo, por exemplo 1000, o gerenciador de tarefas do OurGrid, MyGrid, que estava executando no computador local começava a consumir uma quantidade muito grande de memória o que causava sua finalização pelo sistema

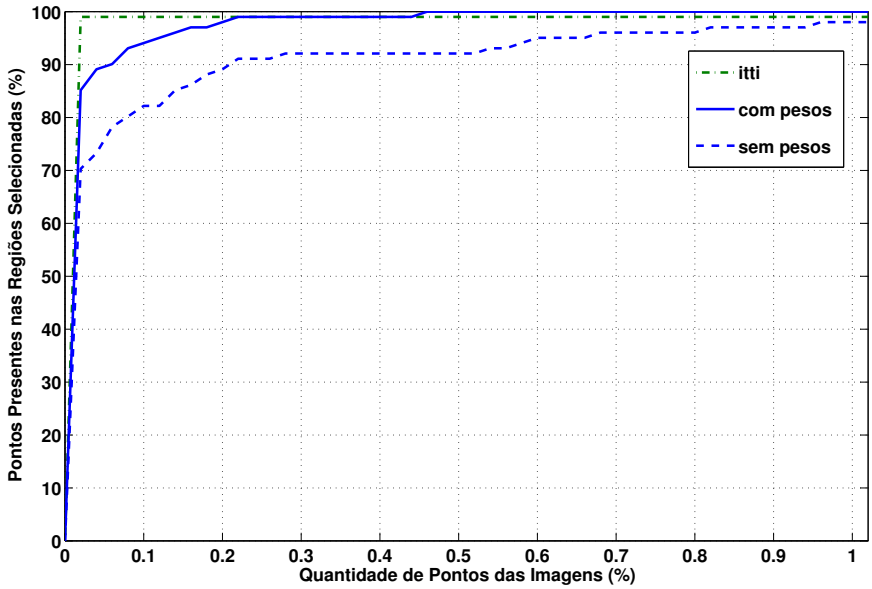


Figura 5.9: Comparação dos resultados para imagens contendo carros.

operacional. Após a postagem de dúvidas no forum do OurGrid, ficou entendido que o alto consumo de memória era devido a um problema na versão do OurGrid que estava sendo utilizada e que tal problema seria resolvido em versões posteriores do OurGrid.

Para resolver o problema de forma indireta, foi proposta a escrita de um *shell script* que gerenciasse a memória do sistema de forma que quando o consumo de memória pelo gerenciador de tarefas atingisse determinado valor, 70% do total de memória do sistema por exemplo, o *script* suspendesse temporariamente a execução das tarefas, salvasse os resultados obtidos até aquele momento e reiniciasse o processamento a partir do *job* onde havia parado.

Esta solução foi implementada criando-se um *shell script* que faz o gerenciamento da memória utilizada e acrescentado-se à função que envia os *jobs* para a grade um gerenciador de tarefas que ao enviar o *job* para a grade lança uma *thread* que executa o *shell script* em paralelo com o processo que é executado na grade e faz a verificação da memória consumida pelo processo local. O *shell script* implementado é mostrado no Apêndice A.

Apesar desta solução ter permitido a finalização dos experimentos, ela criou alguns efeitos colaterais ao processamento. Primeiro, quando a memória utilizada pelo gerenciador de tarefas atingia o valor máximo permitido pelo *script*, as tarefas que estavam sendo executa-

das eram abandonadas pelo gerenciador o que causava acúmulo de *lixo* (código e imagens) nas máquinas remotas. Após algumas paradas, as máquinas remotas que estavam com lixo acumulado não tinham mais espaço em disco disponível para guardar novo material enviado pelas tarefas o que causava erros durante o processamento. Segundo, o processo de parar a tarefa que está sendo executada e reiniciar o processamento do ponto onde havia parado é custoso em relação ao tempo necessário para ser executado o que tornou o processamento mais lento.

As bibliotecas do OurGrid sofreram algumas modificações e atualização de versões. Isto provocou algumas incompatibilidades do sistema aqui exposto com as novas versões, pois o sistema tinha sido implementado com a versão 3.2 e as máquinas da grade foram atualizadas para a versão 3.3. Como as versões 3.2 e 3.3 do OurGrid são incompatíveis, foi necessária a reimplementação de algumas classes. O fato de se estar utilizando versões muito novas destas bibliotecas ocasionou alguns problemas de implementação devido à incompletude da documentação. No entanto, o forum de discussão do OurGrid foi de grande valia na resolução dos problemas de implementação.

A nova versão do OurGrid trouxe duas ótimas melhorias. Primeiro, não é mais necessário que o programa salve em disco o número do último *job* executado, pois o *MyGrid* faz isso automaticamente. A segunda melhoria foi excelente, o problema de estouro de memória ao executar uma quantidade muito grande de *jobs* não ocorre mais.

Outro problema ocasionado pela atualização de versões do OurGrid foi a inserção de *bugs* inexistentes nas versões anteriores. Particularmente, um *bug* relacionado às funções de armazenamento das tarefas nos computadores remotos. Como anteriormente o sistema aqui exposto utilizava a função *storage* e esta função apresentou problemas na nova versão do OurGrid, ela teve que ser substituída pela função *put*. O problema com a função *storage* é que algumas tarefas nunca finalizavam. Apesar dos *bugs* inseridos e da incompatibilidade de versões, a nova versão do OurGrid trouxe uma série de características que superam todos os novos problemas.

5.6 Considerações Finais

Este capítulo relatou os experimentos realizados para otimizar os mapas de saliências do mecanismo de atenção visual, bem como os experimentos para avaliar o desempenho e comparar o sistema proposto com um existente e amplamente utilizado. Com excessão dos programas que rodaram na grade computacional, todos os experimentos foram executados em um PC convencional (com 1GHz de FLOPS e 512M de RAM). O sistema de algoritmo genético foi implementado utilizando a biblioteca *GAlib (http://lancet.mit.edu/ga/)*. A otimização do algoritmo genético foi realizada em grade por meio da utilização do *OurGrid* (www.ourgrid.org).

Além disso, foram realizados vários experimentos de comparação com o sistema de atenção visual desenvolvido pelo laboratório iLab da *University of Southern California*. Este sistema de atenção visual é baseado no modelo proposto por Itti et al [Itti et al., 1998]. O próximo capítulo apresenta as conclusões desta dissertação, apontando as principais contribuições, os objetivos atingidos e os possíveis trabalhos futuros.

Capítulo 6

Conclusão

Este capítulo apresenta um sumário dos principais pontos discutidos nesta dissertação, bem como as contribuições da pesquisa desenvolvida e sugestões de trabalhos futuros.

6.1 Sumário da Dissertação

O Capítulo 1 apresentou qual problema serviu de motivação para que este trabalho fosse realizado. A motivação surgiu do questionamento sobre a viabilidade de guiar a atenção visual *bottom-up* (baseado no modelo que utiliza mapas de saliência) utilizando algoritmos genéticos. Então, o principal objetivo apresentado foi investigar se é possível criar um mecanismo de atenção visual que possa ser otimizado para destacar como regiões importantes qualquer classe de objetos desejada.

Em seguida, no Capítulo 2, foi feita uma revisão bibliográfica sobre sistemas que utilizam atenção visual e algoritmos genéticos. Após a análise destes artigos, ficou evidente a ausência de trabalhos que utilizam algoritmos genéticos para inserir informações de alto nível em sistemas de atenção visual *bottom-up* do modo como é realizado neste trabalho.

Os principais conceitos envolvidos nesta dissertação foram elucidados no Capítulo 3. Neste capítulo foi apresentado o modelo proposto por Itti et al. [Itti et al., 1998] que é um dos modelos de atenção visual *bottom-up* mais conhecidos. O módulo de atenção visual desenvolvido neste trabalho é uma versão adaptada do modelo de Itti. Além disso, o Capítulo 3 também descreveu o principal tipo de algoritmo genético (Algoritmo Genético Canônico - AGC) [DeJong, 1975]. O módulo de otimização de pesos do sistema aqui exposto foi

implementado utilizando a biblioteca *GAlib*.

A arquitetura do sistema foi apresentada no Capítulo 4. O sistema é composto por três módulos: verificação de regiões de interesse, atenção visual e otimização de pesos. O módulo de verificação de regiões é responsável pelos cálculos estatísticos dos resultados obtidos pelo módulo de atenção visual. O módulo de atenção visual utiliza os pesos obtidos pelo módulo de otimização. O módulo de otimização de pesos contém as funções do algoritmo genético e as funções de comunicação com o *OurGrid*.

Os experimentos realizados bem como a análise de seus resultados são apresentados no Capítulo 5. Foram realizados experimentos com imagens nas quais o assunto era uma das seguintes classes: objetos genéricos, pistolas ou revólveres, carros e faces de pessoas. Também foram realizados experimentos comparativos entre o sistema proposto e o sistema de Itti. Além disso, este capítulo faz uma análise dos melhores pesos encontrados pelo algoritmo genético para cada classe de imagens, identificando quais as características mais importantes para guiar a atenção para as classes de imagens.

Diante dos resultados obtidos, consideramos que o objetivo de investigar a viabilidade de guiar a atenção visual *bottom-up* utilizando algoritmos genéticos foi alcançado. Os resultados apresentados no Capítulo 5 mostram que a utilização de algoritmos genéticos aumenta a taxa de pontos localizados em regiões pré-definidas em até 20%. Além disso, esta pesquisa resultou em duas publicações [Pereira and Gomes, 2006; Pereira et al., 2006].

Na próxima seção, apresentamos as principais contribuições deste trabalho e explicitamos os objetivos alcançados.

6.2 Contribuições

Como foi apresentado no Capítulo 3, a atenção visual *bottom-up* indica as regiões mais importantes de uma imagem como sendo aquelas que despertam o interesse do observador de forma inconsciente. Esta atenção é guiada apenas por características de baixo nível da imagem. Há vários sistemas e modelos que propõem modos de associar conhecimentos de alto nível a processos *bottom-up* [Milanese et al., 1994; Sun and Fisher, 2003; Navalpakkam and Itti, 2003]. No entanto, nenhum desses modelos utiliza otimização de

mapas de características como o sistema aqui proposto.

O sistema proposto nesta dissertação possui as seguintes contribuições:

- Aplicação de pesos aos diversos mapas de características que são utilizados para formar um mapa de saliência. A novidade está no modo como estes pesos são obtidos. Para a obtenção desses pesos, selecionam-se manualmente regiões salientes num conjunto de imagens. Em seguida, gera-se um conjunto de pesos que são aplicados aos mapas de característocas obtidos pelo processamento dessas imagens. Calcula-se a média de pontos de atenção de forma que pelo menos um ponto esteja presente nas regiões selecionadas manualmente. O algoritmo genético evolui a fim de minimizar essa média. Assim, tem-se um mecanismo genérico de ajuste, que pode ser facilmente aplicado a diferentes classes de problemas.

- Processamento em grade. Como o processamento de uma imagem pelo módulo de atenção visual leva, em média, 20 segundos para ser executado em um computador com 512MB de memória RAM e 1GHZ de clock, havia um número muito grande de imagens para serem avaliadas em cada geração do algoritmo genético, os parâmetros estabelecidos para o algoritmo genético determinavam o uso de 80 gerações de 40 indivíduos, foi necessária a utilização de processamento paralelo. Pois, considerando-se que a cada geração processam-se 100 imagens , seriam necessários mais de 100 dias para processar todas as imagens utilizando-se apenas um computador. Utilizando-se a grade computacional o tempo de processamento foi reduzido para mais ou menos 24 horas.

- Estudo de modelos de atenção visual *bottom-up* e de algoritmos genéticos. Comparação e avaliação estatística do modelo de atenção visual implementado com um modelo existente na literatura [Itti et al., 1998]. As contribuições deste item comprovam que os objetivos específicos apresentados no Capítulo 1 foram alcançados.

Além disso, os experimentos demonstram que o sistema proposto pode ser otimizado para diferentes classes de objetos. Desta forma, ele pode servir como um módulo para um sistema genérico de detecção de objetos. Na próxima seção, são apresentadas sugestões de trabalhos futuros.

6.3 Trabalhos Futuros

Esta seção apresenta algumas sugestões de trabalhos futuros relacionados à obtenção de um melhor desempenho pelo sistema de atenção visual. Além disso, apresenta sugestões de sistemas que poderiam utilizar o sistema aqui proposto como um meio para agilizar tarefas de detecção e reconhecimento.

6.3.1 Outras Formas para Otimização de Algoritmos Genéticos

Vários outros parâmetros poderiam ser utilizados na otimização de algoritmos genéticos. Por exemplo, ao invés de utilizar populações isoladas de indivíduos, poderia ser feito um estudo sobre a utilização de populações que evoluem paralelamente e que em determinados períodos trocam informações. Poderia, também, ser realizada uma investigação sobre a viabilidade de otimizar outros parâmetros além da média de pontos, como por exemplo, o desvio padrão. Esta otimização seria tanto dos parâmetros isolados quanto dos parâmetros associados (múltiplos objetivos).

Deve-se salientar que o sistema de atenção visual implementado nesta dissertação é uma versão adaptada do modelo proposto por Itti et al. [Itti et al., 1998] que está disponível para download em *http://ilab.usc.edu/toolkit/*. Há dois estudos que poderiam ser realizados com o intuito de melhorar o desempenho do módulo de atenção visual. O primeiro seria incrementar a quantidade de características primitivas utilizadas (originalmente cor, intensidade e orientação) com profundidade, movimento, textura, etc. O segundo seria investigar um modo de aplicar os pesos ao sistema de Itti e realizar a otimização do algoritmo genético tendo como módulo de detecção tal sistema.

Algumas tentativas de otimizar o próprio sistema de Itti com pesos obtidos por um algoritmo genético foram realizadas. No entanto, nos deparamos com dois principais problemas: dificuldade de instalação do sistema e tamanho dos executáveis resultantes. A dificuldade de instalação decorre do fato do sistema necessitar que uma grande quantidade de bibliotecas de otimização seja instalada. Como o módulo de atenção visual deve ser enviado para os computadores remotos a cada geração do algoritmo genético, é necessário que seu tamanho seja pequeno e que seja encontrada uma maneira de superar o problema das dependências de bibliotecas.

6.3.2 Aplicações do Sistema Proposto

O sistema proposto pode ser utilizado como um módulo em sistemas de detecção ou reconhecimento. Ele serviria como meio para agilizar a localização dos objetos mais importantes da cena. Como a otimização por meio de algoritmos genéticos provê uma capacidade de generalização a sistemas de atenção visual *bottom-up*, o método aqui exposto pode ser utilizado como etapa prévia na detecção ou reconhecimento de qualquer classe de objetos.

Como exemplos de aplicações práticas do sistema temos: filtragem web e segurança de ambientes. No primeiro caso, o sistema funcionaria acoplado a um navegador web e filtraria páginas que contivessem imagens com determinados tipos de objetos. Por exemplo, poderia-se evitar que o navegador mostrasse páginas que contivessem imagens de armas. No segundo caso, o sistema poderia ser integrado à rede de câmeras de segurança de algum estabelecimento comercial e ao sinal (emitido por um segurança) de algum indivíduo suspeito carregando um objeto estranho o sistema poderia rastrear as imagens das câmeras em busca do objeto e conseqüentemente do indivíduo.

A utilização de uma maior variabilidade de características (movimento, profundidade estereoscópica, *aspect ratio* e textura, por exemplo) na geração de mapas de saliência pode viabilizar a criação de um detector genérico de objetos utilizando um método de otimização de mapas de características com algoritmos genéticos. Esta seria mais uma aplicação do sistema proposto.

Bibliografia

[Bebis et al., 1999] Bebis, G., Uthiram, S., and Georgiopoulos, M. (1999). Genetic search for face detection and verification. In *International Conference on Information Intelligence and Systems*, pages 360–367.

[Burt and Adelson, 1983] Burt, P. J. and Adelson, E. H. (1983). The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540.

[Darwin, 1909] Darwin, C. (1909). *The Foundations of the Origin of Species*. Cambridge University Press.

[DeJong, 1975] DeJong, K. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan.

[Doyle and Dean, 1996] Doyle, J. and Dean, T. (1996). Strategic directions in artificial intelligence. *ACM Computing Surveys*, 28(4):653–670.

[Fischer and Weber, 1993] Fischer, B. and Weber, H. (1993). Express saccades and visual attention. *Behavioral and Brain Sciences*, 16:553–610.

[Fisher and MacKirdy, 1998] Fisher, R. B. and MacKirdy, A. (1998). Integrating iconic and structured matching. *Lecture Notes in Computer Science*, 1407:687–699.

[Fong. and Hui, 2001] Fong., A. C. M. and Hui, S. C. (2001). Web-based intelligent surveillance system for detection of criminal activities. *Computer and Control Engineering Journal*, pages 263–270.

[Freeman and Adelson, 1991] Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906.

[Fukushima, 1980] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

[Holland, 1975] Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. The MIT Press.

[Huang and Wechsler, 1999] Huang, J. and Wechsler, H. (1999). Eye location using genetic algorithm. In *Second International Conference on Audio and Video-Based Biometric Person Authentication (AVBPA)*, pages 130–135.

[Huang and Wechsler, 2000] Huang, J. and Wechsler, H. (2000). Visual routines for eye location using learning and evolution. *IEEE Transactions on Evolutionary Computation*, 4(1):73–82.

[Itti and Koch, 1999] Itti, L. and Koch, C. (1999). A comparison of feature combination strategies for saliency-based visual attention systems. In *SPIE human vision and eletronic imaging (HVEI '99)*, pages 473–482.

[Itti and Koch, 2000] Itti, L. and Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40:1489–1506.

[Itti and Koch, 2001a] Itti, L. and Koch, C. (2001a). Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203.

[Itti and Koch, 2001b] Itti, L. and Koch, C. (2001b). Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging*, 10(1):161–169.

[Itti et al., 1998] Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

[Itti et al., 2005] Itti, L., Rees, G., and Tsotsos, J. K. (2005). *Models of Bottom-Up Attention and Saliency*. San Diego, CA:Elsevier.

[Jain and Dorai, 1997] Jain, A. and Dorai, C. (1997). Practicing vision: Integration, evaluation and applications.

[Jolliffe, 2002] Jolliffe, I. (2002). *Principal Component Analysis*. Springer Verlag.

[Lopez et al., 2006] Lopez, M. T., Fernandez-Caballero, A., Fernandez, M. A., and Delgado, J. M. A. E. (2006). Visual surveillance by dynamic visual attention method. *Pattern Recognition*, pages 2194–2211.

[Milanese et al., 1994] Milanese, R., Wechsler, H., and Gil, S. (1994). Integration of bottom-up and top-down cues for visual attention using non-linear relaxation. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 781–785.

[Navalpakkam and Itti, 2002] Navalpakkam, V. and Itti, L. (2002). A goal oriented attention guidance model. In *Second International Workshop on Biologically Motivated Computer Vision*, pages 453–461.

[Navalpakkam and Itti, 2003] Navalpakkam, V. and Itti, L. (2003). Sharing resources: Buy attention, get object recognition. In *International Workshop on Attention and Performance in Computer Vision WAPCV'2003*, pages 73–79.

[Navalpakkam and Itti, 2006] Navalpakkam, V. and Itti, L. (2006). An integrated model of top-down and bottom-up attention for optimal object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2049–2056.

[Pereira and Gomes, 2006] Pereira, E. T. and Gomes, H. M. (2006). Guiding a bottom-up visual attention mechanism to locate specific image regions using a distributed genetic optimization. In *CIARP*, pages 257–266.

[Pereira et al., 2006] Pereira, E. T., Gomes, H. M., and Florentino, V. F. C. (2006). Bottom-up visual attention guided by genetic algorithm optimization. In *Eigth IASTED International Conference on Signal and Image Processing*, pages 228–233.

[Pinkerton, 1994] Pinkerton, B. (1994). Finding what people want: Experiences with the webcrawler.

[Rodrigues, 2002] Rodrigues, F. A. (2002). Localização e reconhecimento de placas de sinalização utilizando um mecanismo de atenção visual e redes neurais artificiais. Master's thesis, Universidade Federal de Campina Grande.

[Santos, 2005] Santos, S. M. (2005). Um mecanismo de atenção visual integrando evidências espaciais e temporais. Master's thesis, Universidade Federal de Campina Grande.

[Siagian and Ititi, 2004] Siagian, C. and Ititi, L. (2004). Biologically-inspired face detection: Non-brute-force-search approach. In *First IEEE-CVPR International Workshop on Face Processing in Video*, pages 62–69.

[Sim et al., 2003] Sim, T., Baker, S., and Bsat, M. (2003). The cmu pose, illumination, and expression (pie) database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1624.

[Simoncelli and Freeman, 1995] Simoncelli, E. P. and Freeman, W. T. (1995). The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *IEEE International Conference on Image Processing*, pages 444–447.

[Sun and Fisher, 2003] Sun, Y. and Fisher, R. (2003). Object-based visual attention for computer vision. *Artificial Intelligence*, 146(1):77–123.

[Sun et al., 2003] Sun, Z., Bebis, G., and Miller, R. (2003). Boosting object detection using feature selection. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03)*, pages 290–296.

[Tsotsos, 1990] Tsotsos, J. (1990). Analyzing vision at the complexity level. *The Behavioral and Brain Sciences*, 13(3):423–445.

[Turk and Pentland, 1991] Turk, M. and Pentland, A. (1991). Face recognition using eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591.

[Vapnik, 1995] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer Verlag.

[Walther et al., 2002] Walther, D., Itti, L., Riesenhuber, M., Poggio, T., and Koch, C. (2002). Attentional selection for object recognition - a gentle way. In *Biologically Motivated Computer Vision - Lecture Notes in Computer Science, Springer*, pages 472–479.

[Whitley, 1994] Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85.

[Wildes, 1998] Wildes, R. P. (1998). A measure of motion salience for surveillance applications. In *Proceedings of the IEEE International Conference on Image Processing*, pages 183–187.

[Wolfe, 2000] Wolfe, J. M. (2000). *Seeing*. Academic Press.

[Wolfe and Horowitz, 2004] Wolfe, J. M. and Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews: Neuroscience*, 5:1–7.

Apêndice A

Shell Script para Gerenciamento de Memória

O código de script utilizado para gerenciar a memória, quando o algoritmo genético é executado na Grade é mostrado abaixo.

```
#!/bin/bash
function get_number_of_jobs {
        NUMBER_OF_JOBS=$(ls ~/mem_manager/task_spec/);
}

function save_previous_results {
        mkdir ~/ga/resultados/$1/
        cp -r ~/workspace/VisualAttention/bin/. ~/ga/resultados/$1/.
}

function clear_previous_results {
        rm -rf ~/ga/resultados/*
}

function get_task {
        cat ~/mem_manager/task_spec/$1 | head -$2 |
              tail -$3 > ~/mem_manager/task.txt
}
function init_jdf {
        REQUERIMENTS="requirements : "\("os = linux &&
```

```
                        site != copad-lmrs.lmrs-semarh.ufcg.edu.br &&
                        site != topgrid.dcc.ufba.br &&
                        site != lsd.ufcg.edu.br &&
                        name != 150.165.87.151 &&
                        name != 150.165.87.172"\)""
        echo
        echo "job : "
        echo
        echo "label: $NUMBER_OF_JOBS"
        echo
        echo $REQUERIMENTS;
        echo
}
function write_last_job {
        get_number_of_jobs;
        init_jdf > ~/mem_manager/last_job.jdf;
        NUMBER_OF_LINES=$(cat
        ~/mem_manager/task_spec/$NUMBER_OF_JOBS | wc -l);
        j=8;

        for((i=1;i<$NUMBER_OF_LINES; i=i+8)); do
                get_task $NUMBER_OF_JOBS $j 8;
                INIT1=$(cat ~/mem_manager/task.txt |
                grep STORE | cut -f9 -d" ");
                INIT2=$(cat ~/mem_manager/task.txt |
                grep STORE | cut -f11 -d" " | cut -f1 -d"]");
                REMOTE=$(cat ~/mem_manager/task.txt | grep "xzf");
                FINAL=$(cat ~/mem_manager/task.txt |
                grep GET | cut -f11 -d" " | cut -f1 -d"]");
                j=$((j+8));

                echo "task : "
                echo
                echo "init : store $INIT1 $INIT2"
                echo
                echo "remote : $REMOTE"
                echo
```

```
                echo "final : get objetos.txt $FINAL"
                echo
        done
}
function recreate_last_job {
        write_last_job >> ~/mem_manager/last_job.jdf;
}
function kill_mygrid_gui {
        PID_MYGRID_GUI=$(ps ax | grep
        'org.ourgrid.mygrid.ui.gui.MyGridGUI' | cut -f1 -d'p');
        PID_MYGRID_GUI=$(echo $PID_MYGRID_GUI | cut -f1 -d' ');
        if ! [ "$PID_MYGRID_GUI" = "" ]; then
                kill $PID_MYGRID_GUI;
        fi
}

function kill_gaongrid {
        PID_GAONGRID=$(ps ax | grep  'GAOnGrid' | grep 'java' |
        cut -f1 -d'p');
        PID_GAONGRID=$(echo $PID_GAONGRID | cut -f1 -d' ');
        kill $PID_GAONGRID;
}
function see_if_ga_on_grid_is_running {
        PID_GAONGRID=$(ps ax | grep  'GAOnGrid' | grep 'java' |
        cut -f1 -d'p');
        PID_GAONGRID=$(echo $PID_GAONGRID | cut -f1 -d' ');
        while ! [ "$PID_GAONGRID" = "" ]; do
                PID_GAONGRID=$(ps ax | grep  'GAOnGrid' |
                grep 'java' | cut -f1 -d'p');
                PID_GAONGRID=$(echo $PID_GAONGRID | cut -f1 -d' ');
        done

}
function mygrid_manager {
        see_if_ga_on_grid_is_running ;
        kill_mygrid_gui;
        mygrid stop;
```

```
        mygrid start;
        mygrid setgrid /usr/share/mygrid/meugrid.gdf;
        PID_MYGRID=$(ps ax | grep  'org.ourgrid.mygrid.main.Main'
        | cut -f1 -d'p');
        PID_MYGRID=$(echo $PID_MYGRID | cut -f1 -d' ');
        MEM=$(ps -p $PID_MYGRID -o pmem);
        MEM=$(echo $MEM | cut -f2 -d' ');
        PID_GAONGRID=$(ps ax | grep  'GAOnGrid' | grep 'java' |
        cut -f1 -d'p');
        PID_GAONGRID=$(echo $PID_GAONGRID | cut -f1 -d' ');

        while ! [ "$PID_GAONGRID" = "" ]&& [ "$MEM" \< "70" ]; do
                PID_MYGRID=$(ps ax | grep
                'org.ourgrid.mygrid.main.Main' | cut -f1 -d'p');
                PID_MYGRID=$(echo $PID_MYGRID | cut -f1 -d' ');
                PID_GAONGRID=$(ps ax | grep  'GAOnGrid' |
                grep 'java' | cut -f1 -d'p');
                PID_GAONGRID=$(echo $PID_GAONGRID | cut -f1 -d' ');
                MEM=$(ps -p $PID_MYGRID -o pmem);
                MEM=$(echo $MEM | cut -f2 -d' ');
        done
        if [ "$MEM" \> "70" ]; then
                mygrid_manager;
        fi
        exit 0;
}
```

Apêndice B

Gráficos das Evoluções dos Algoritmos Genéticos no Processo de Escolha de um Valor para Mutação

Neste apêndice, são apresentados os gráficos de evolução dos algoritmos genéticos utilizados nos experimentos executados com o intuito de escolher o melhor valor de probabilidade de mutação para as otimizações. Nestes experimentos, apenas duas imagens eram processadas em cada tarefa. Como cada *job* continha 10 tarefas, foram utilizadas 20 imagens em cada experimento. Para todos estes experimentos, o valor de probabilidade de recombinação utilizado foi 60%. Os experimentos foram realizados com valores de mutação que variavam entre 1% e 10%. As seções seguintes mostram os gráficos dos dez experimentos realizados para cada conjunto de imagens. A análise dos gráficos, exceto dos gráficos para imagens contendo objetos, deixa claro que o melhor valor de probabilidade de mutação a ser utilizado com um valor de probabilidade de recombinação igual a 60% é de 1% para o problema aqui examinado. Devido à grande variabilidade dos objetos selecionados manualmente nas imagens contendo objetos genéricos, o algoritmo genético não estabilizou para nenhum valor de mutação nesses pequenos experimentos. Desta forma, o valor de probabilidade de mutação utilizado para objetos genéricos foi de 1%, o mesmo utilizado para os outros tipos de imagens. No processo de escolha do melhor valor de probabilidade de recombinação, foram levados em consideração o ponto onde a curva de evolução iniciou a estabilização e o menor valor atingido pelo indivíduo no ponto de início da estabilização.

B.1 Imagens Contendo Armas

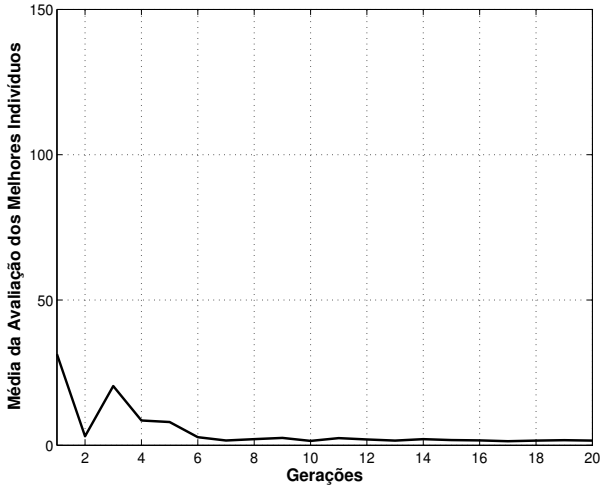


Figura B.1: Melhores médias de cada geração para imagens de armas e valor de mutação igual a 1%.

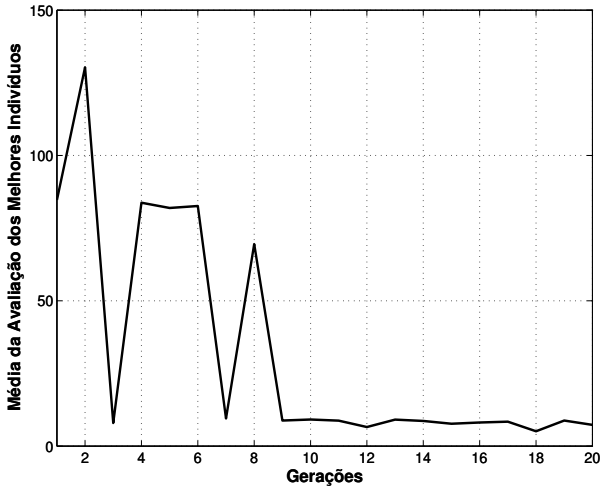


Figura B.2: Melhores médias de cada geração para imagens de armas e valor de mutação igual a 2%.

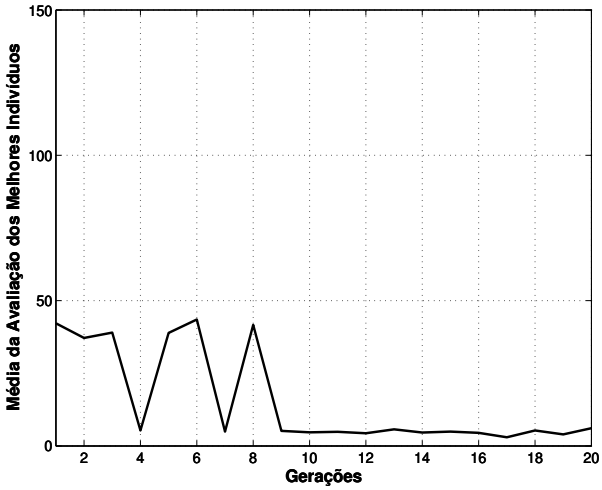


Figura B.3: Melhores médias de cada geração para imagens de armas e valor de mutação igual a 3%.

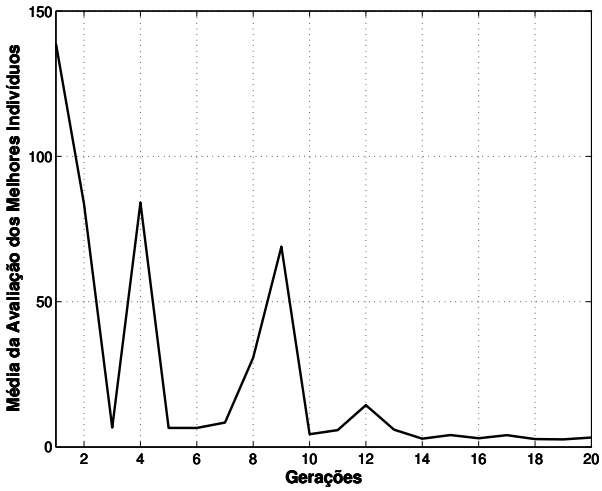


Figura B.4: Melhores médias de cada geração para imagens de armas e valor de mutação igual a 4%.

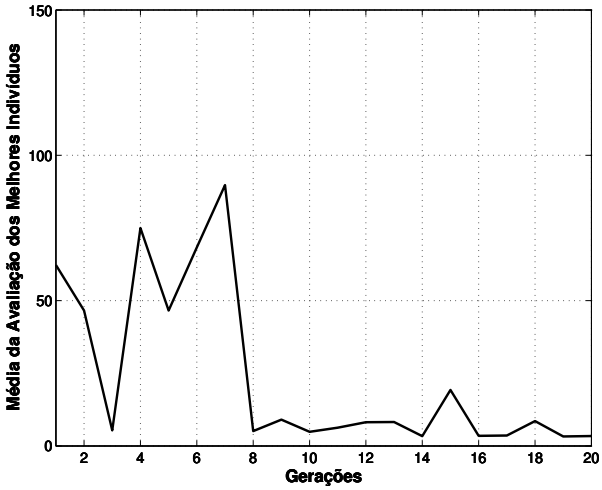


Figura B.5: Melhores médias de cada geração para imagens de armas e valor de mutação igual a 5%.

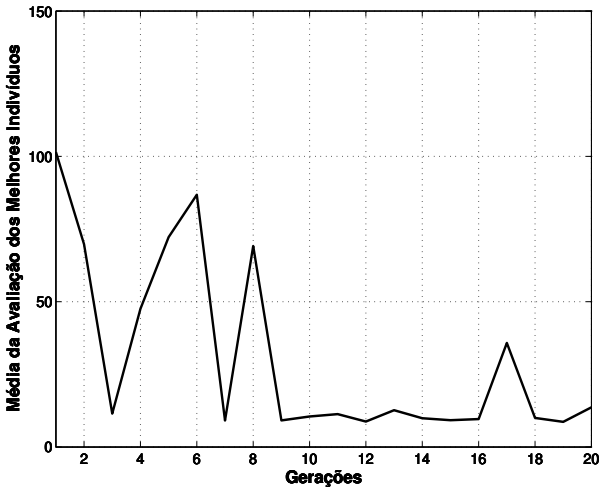


Figura B.6: Melhores médias de cada geração para imagens de armas e valor de mutação igual a 6%.

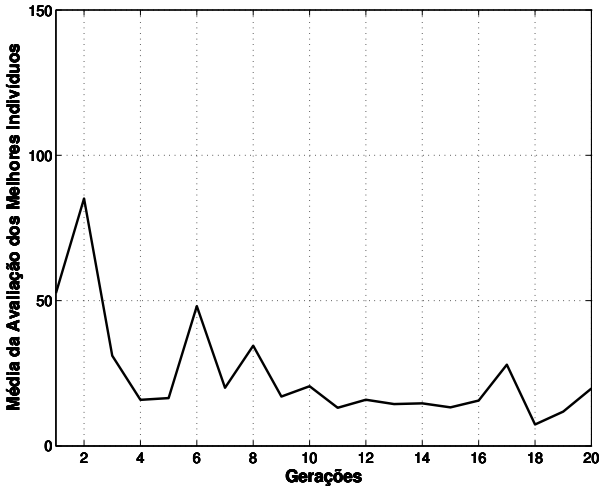


Figura B.7: Melhores médias de cada geração para imagens de armas e valor de mutação igual a 7%.

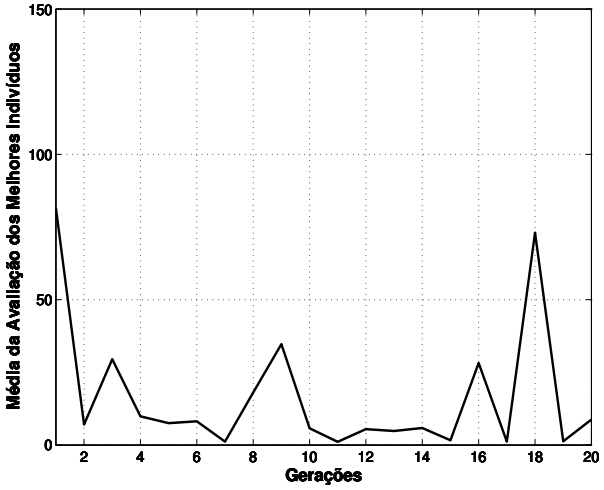


Figura B.8: Melhores médias de cada geração para imagens de armas e valor de mutação igual a 8%.

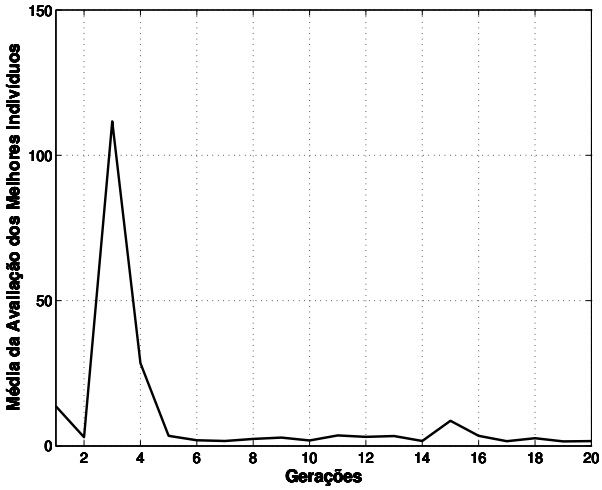


Figura B.9: Melhores médias de cada geração para imagens de armas e valor de mutação igual a 9%.

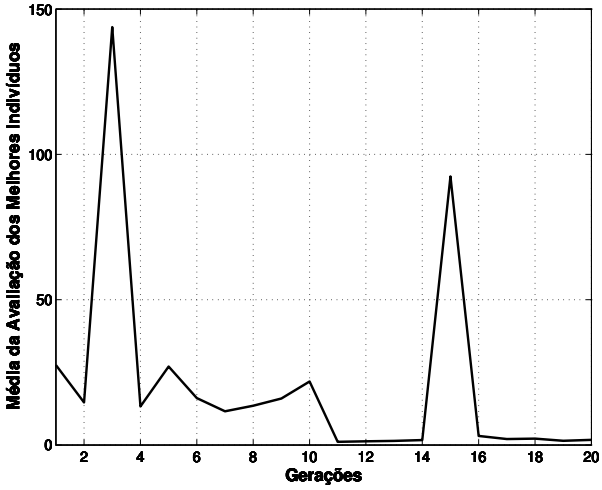


Figura B.10: Melhores médias de cada geração para imagens de armas e valor de mutação igual a 10%.

B.2 Imagens Contendo Objetos Genéricos

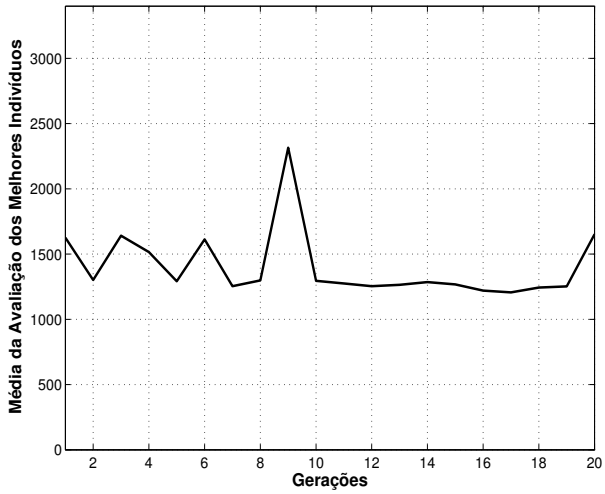


Figura B.11: Melhores médias de cada geração para imagens de objetos e valor de mutação igual a 1%.

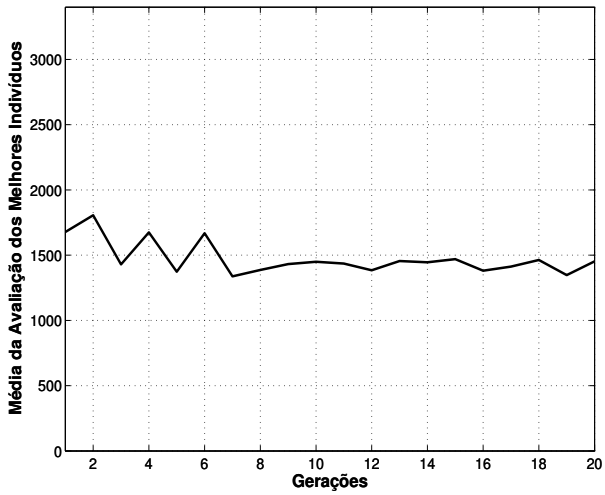


Figura B.12: Melhores médias de cada geração para imagens de objetos e valor de mutação igual a 2%.

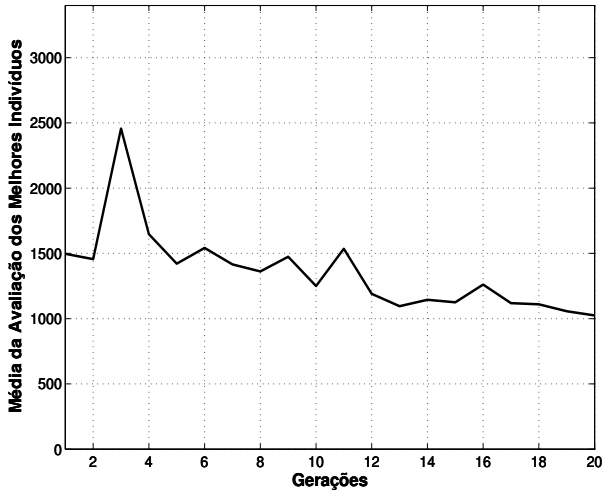


Figura B.13: Melhores médias de cada geração para imagens de objetos e valor de mutação igual a 3%.

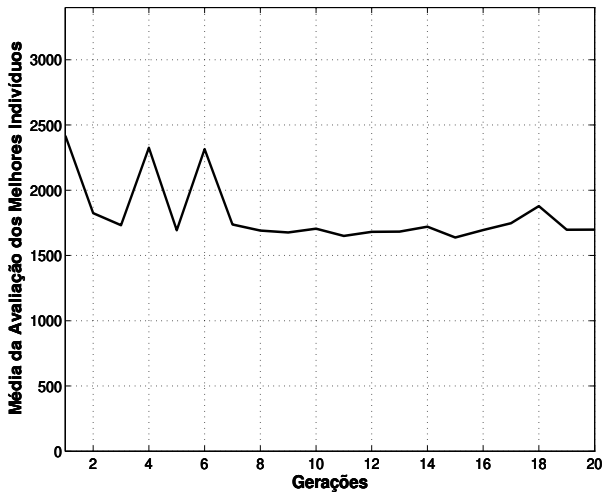


Figura B.14: Melhores médias de cada geração para imagens de objetos e valor de mutação igual a 4%.

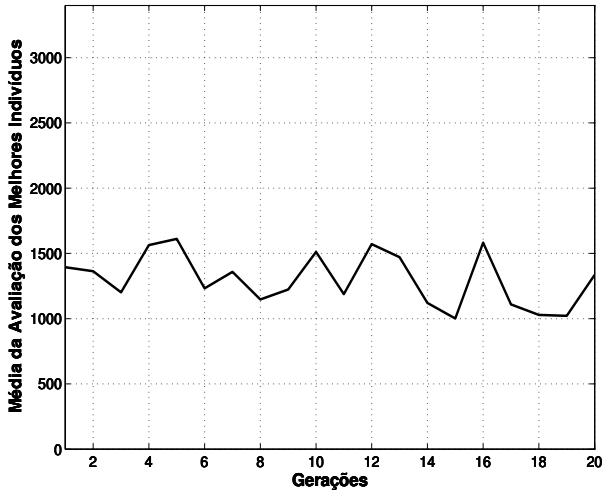


Figura B.15: Melhores médias de cada geração para imagens de objetos e valor de mutação igual a 5%.

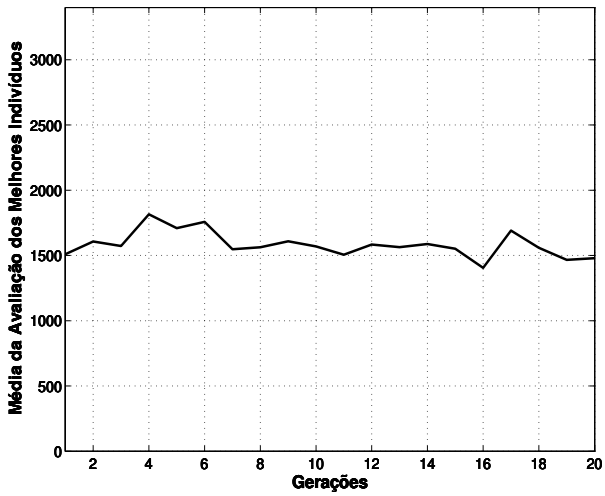


Figura B.16: Melhores médias de cada geração para imagens de objetos e valor de mutação igual a 6%.

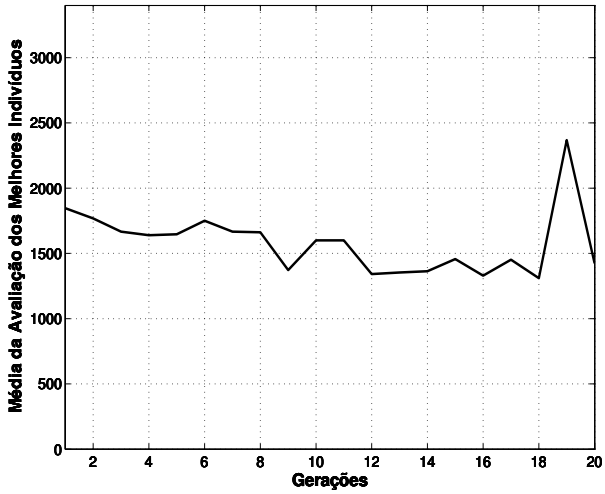


Figura B.17: Melhores médias de cada geração para imagens de objetos e valor de mutação igual a 7%.

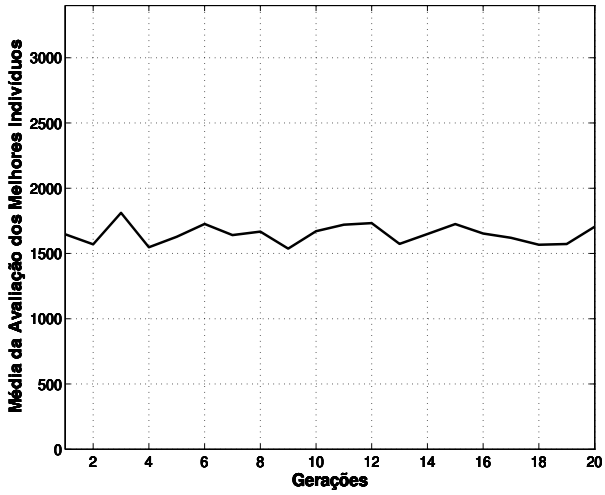


Figura B.18: Melhores médias de cada geração para imagens de objetos e valor de mutação igual a 8%.

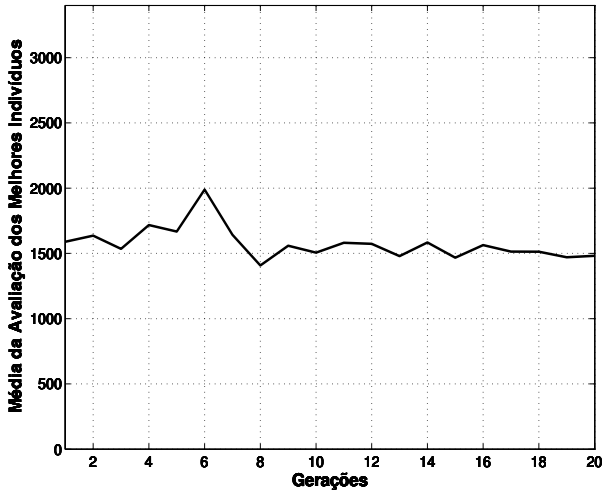


Figura B.19: Melhores médias de cada geração para imagens de objetos e valor de mutação igual a 9%.

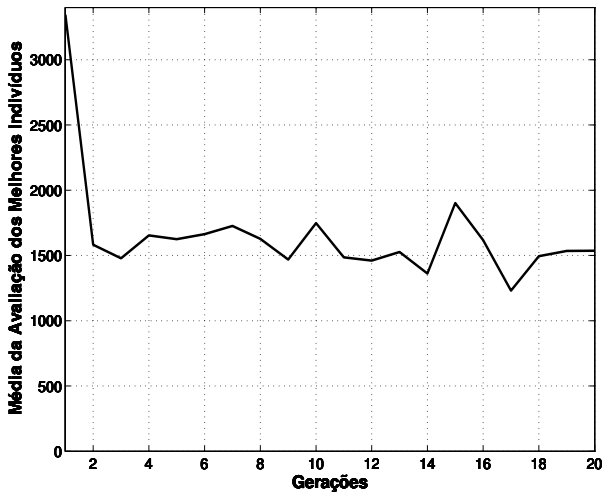


Figura B.20: Melhores médias de cada geração para imagens de objetos e valor de mutação igual a 10%.

B.3 Imagens Contendo Faces de Pessoas

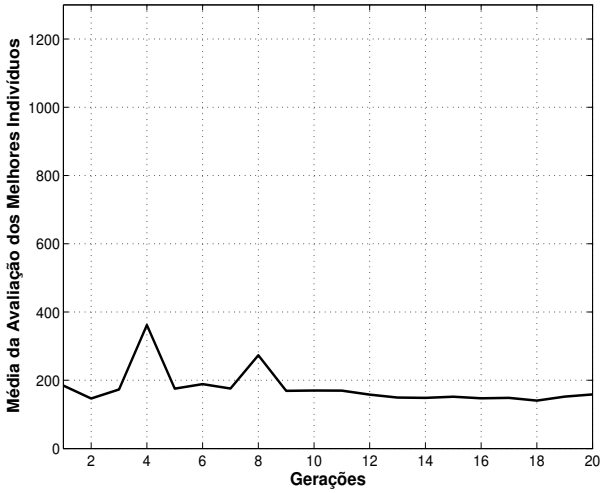


Figura B.21: Melhores médias de cada geração para imagens de pessoas e valor de mutação igual a 1%.

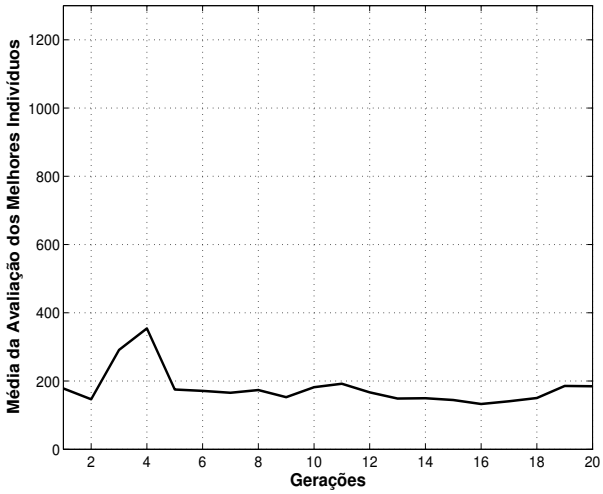


Figura B.22: Melhores médias de cada geração para imagens de pessoas e valor de mutação igual a 2%.

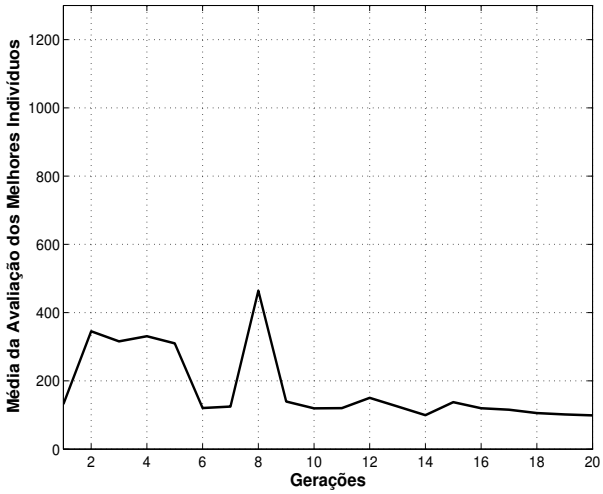


Figura B.23: Melhores médias de cada geração para imagens de pessoas e valor de mutação igual a 3%.

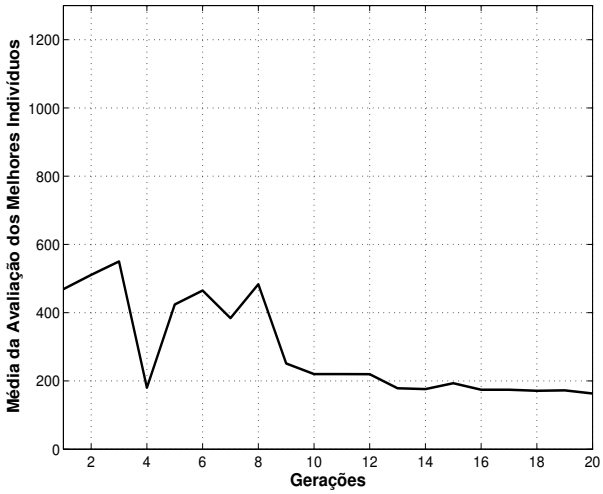


Figura B.24: Melhores médias de cada geração para imagens de pessoas e valor de mutação igual a 4%.

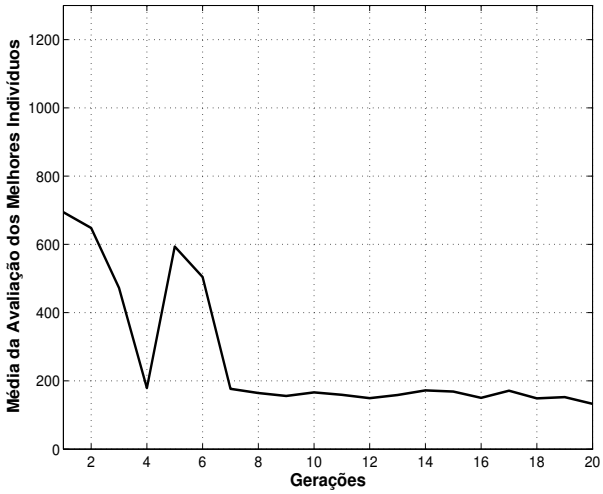


Figura B.25: Melhores médias de cada geração para imagens de pessoas e valor de mutação igual a 5%.

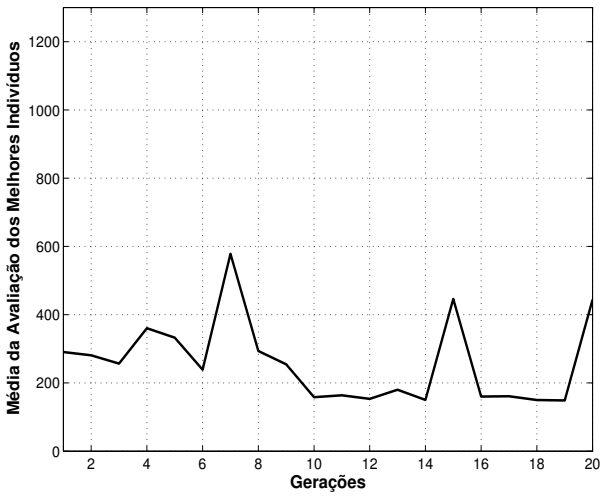


Figura B.26: Melhores médias de cada geração para imagens de pessoas e valor de mutação igual a 6%.

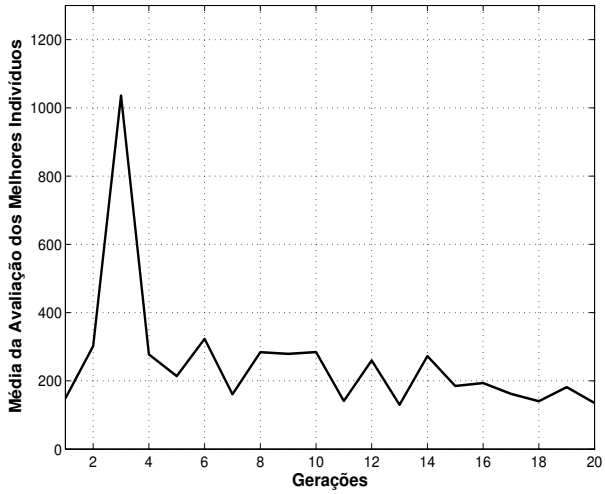


Figura B.27: Melhores médias de cada geração para imagens de pessoas e valor de mutação igual a 7%.

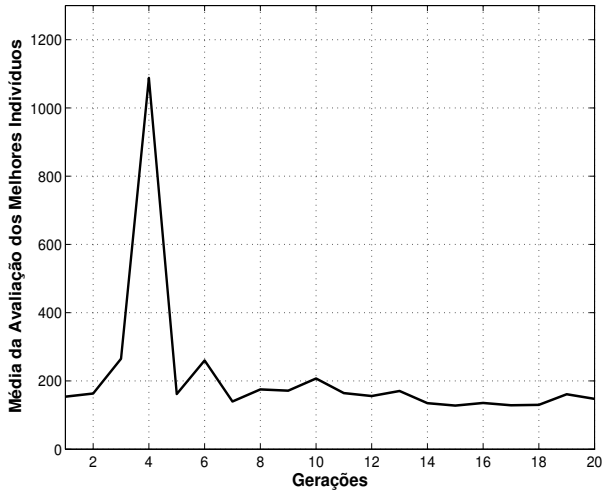


Figura B.28: Melhores médias de cada geração para imagens de pessoas e valor de mutação igual a 8%.

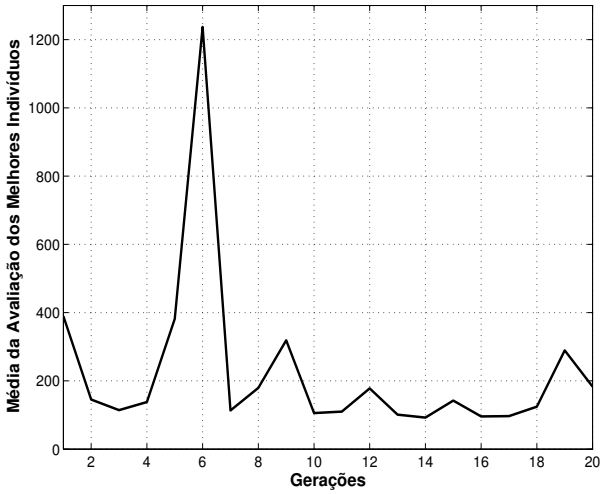


Figura B.29: Melhores médias de cada geração para imagens de pessoas e valor de mutação igual a 9%.

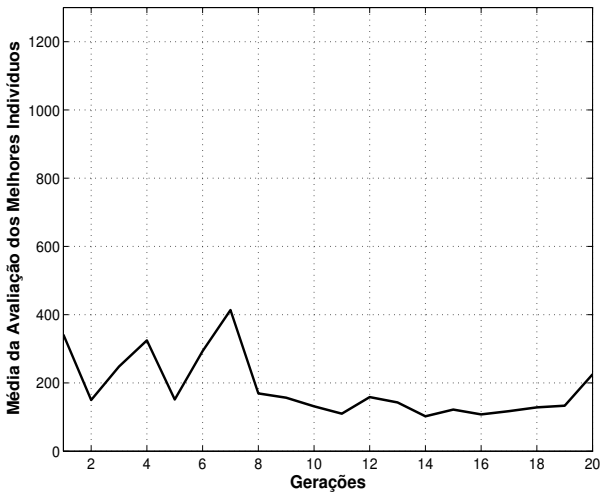


Figura B.30: Melhores médias de cada geração para imagens de pessoas e valor de mutação igual a 10%.

B.4 Imagens Contendo Carros

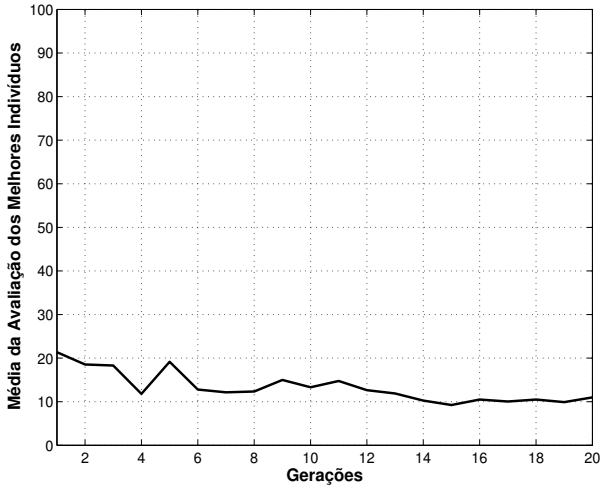


Figura B.31: Melhores médias de cada geração para imagens de carros e valor de mutação igual a 1%.

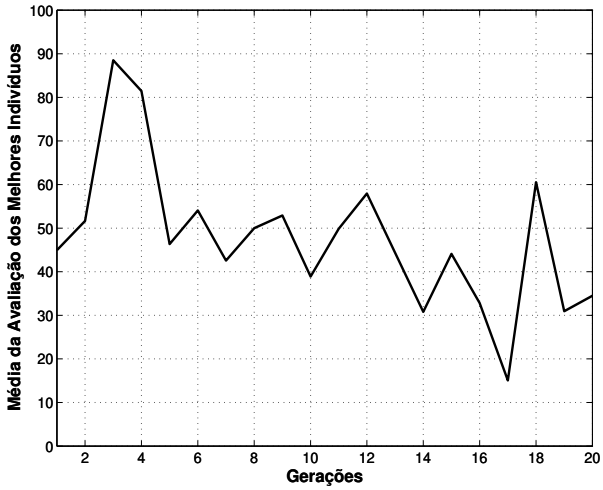


Figura B.32: Melhores médias de cada geração para imagens de carros e valor de mutação igual a 2%.

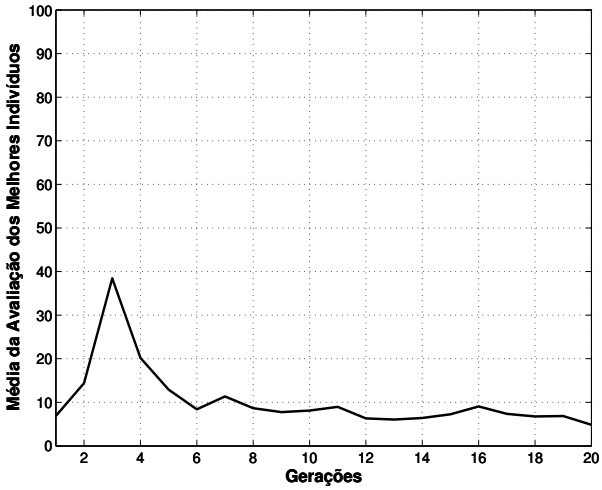


Figura B.33: Melhores médias de cada geração para imagens de carros e valor de mutação igual a 3%.

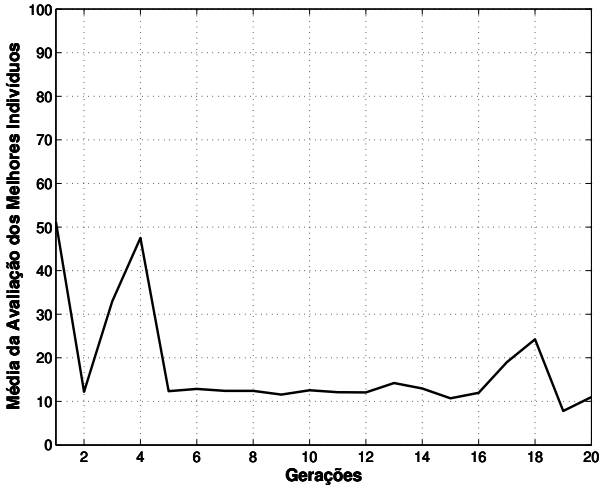


Figura B.34: Melhores médias de cada geração para imagens de carros e valor de mutação igual a 4%.

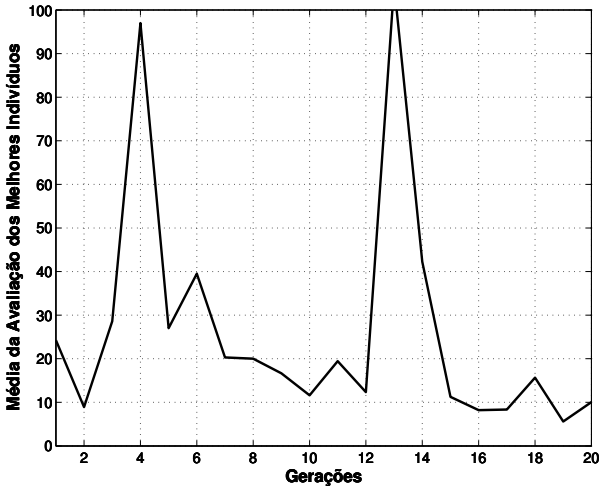


Figura B.35: Melhores médias de cada geração para imagens de carros e valor de mutação igual a 5%.

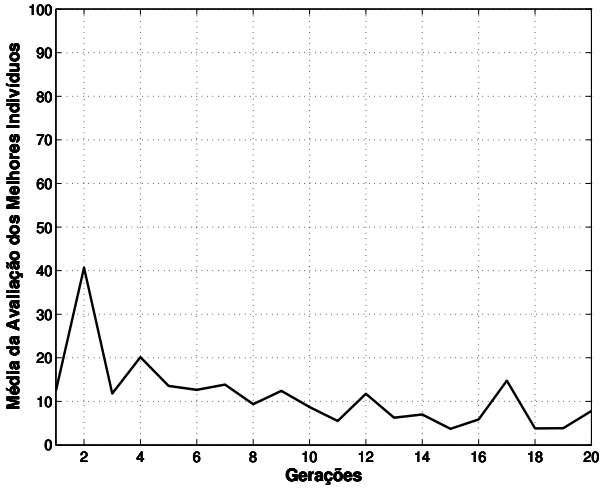


Figura B.36: Melhores médias de cada geração para imagens de carros e valor de mutação igual a 6%.

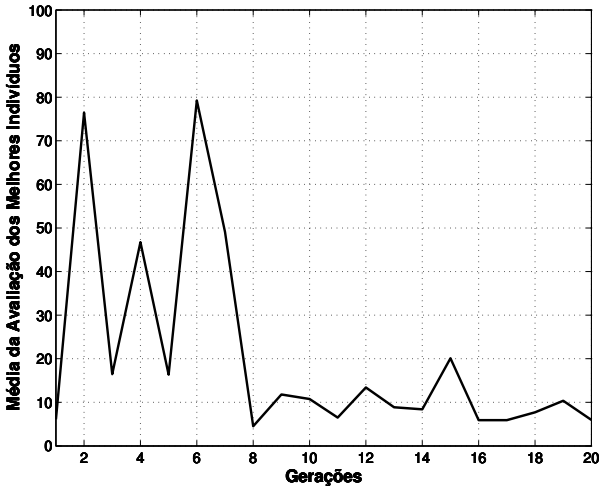


Figura B.37: Melhores médias de cada geração para imagens de carros e valor de mutação igual a 7%.

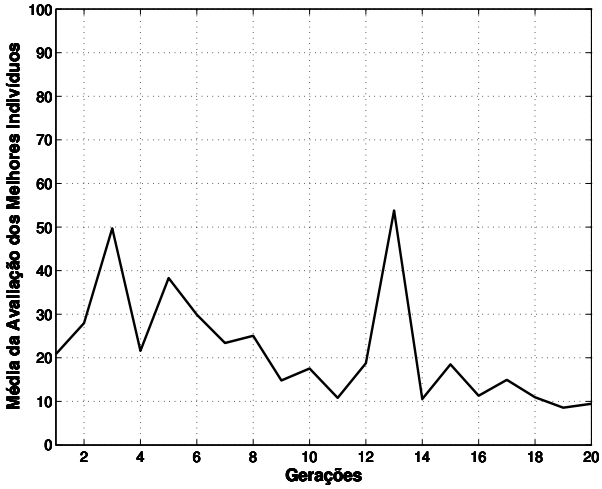


Figura B.38: Melhores médias de cada geração para imagens de carros e valor de mutação igual a 8%.

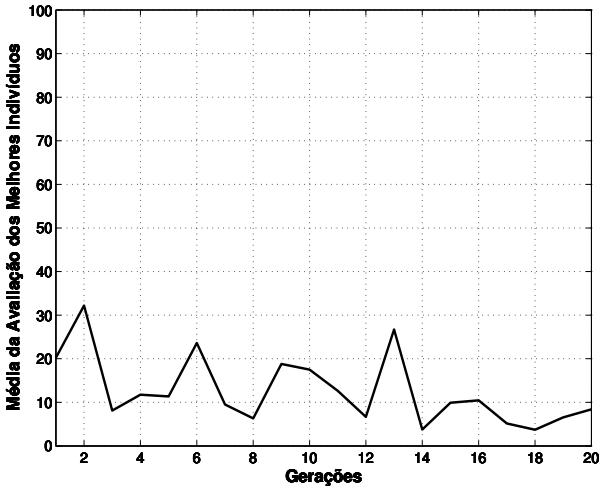


Figura B.39: Melhores médias de cada geração para imagens de carros e valor de mutação igual a 9%.

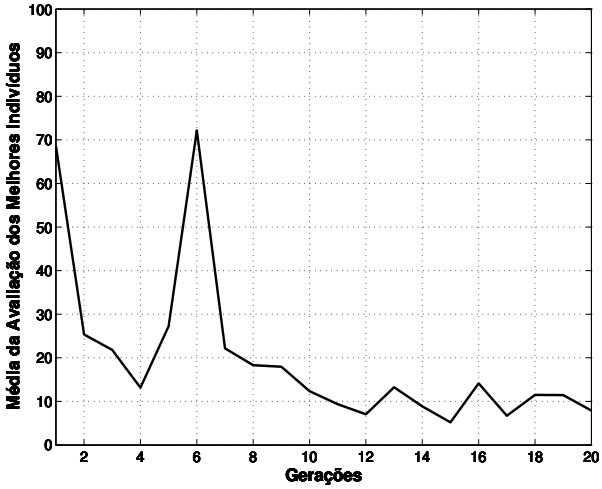


Figura B.40: Melhores médias de cada geração para imagens de carros e valor de mutação igual a 10%.

Apêndice C

Gráficos das Otimizações

Abaixo são apresentados os gráficos das médias e desvios-padrão das quantidades de pontos salientes para todos os indivíduos das otimizações para cada classe de região.

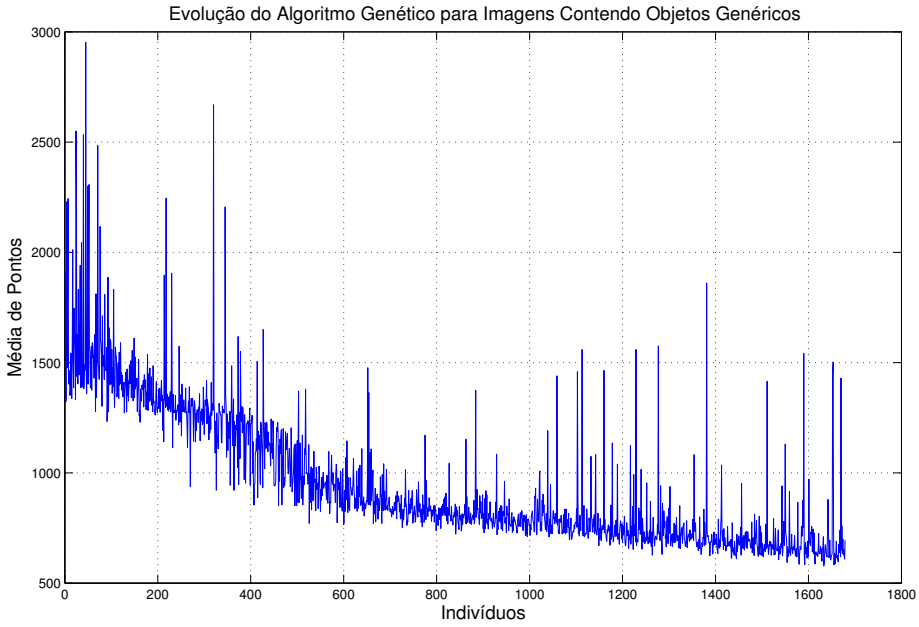


Figura C.1: Médias para imagens contendo objetos genéricos.

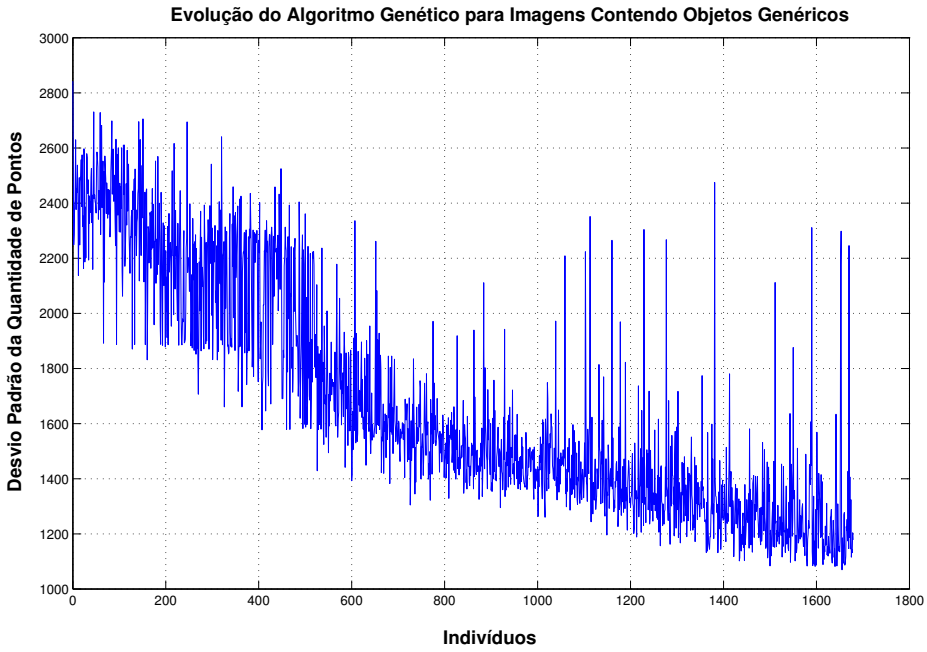


Figura C.2: Desvios-padrão para imagens contendo objetos genéricos.

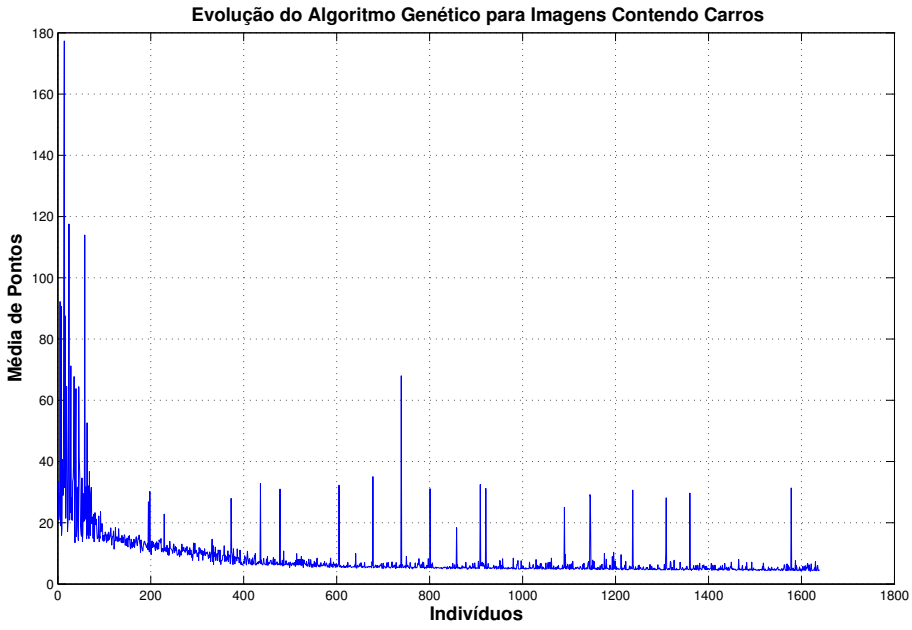


Figura C.3: Médias para imagens contendo carros.

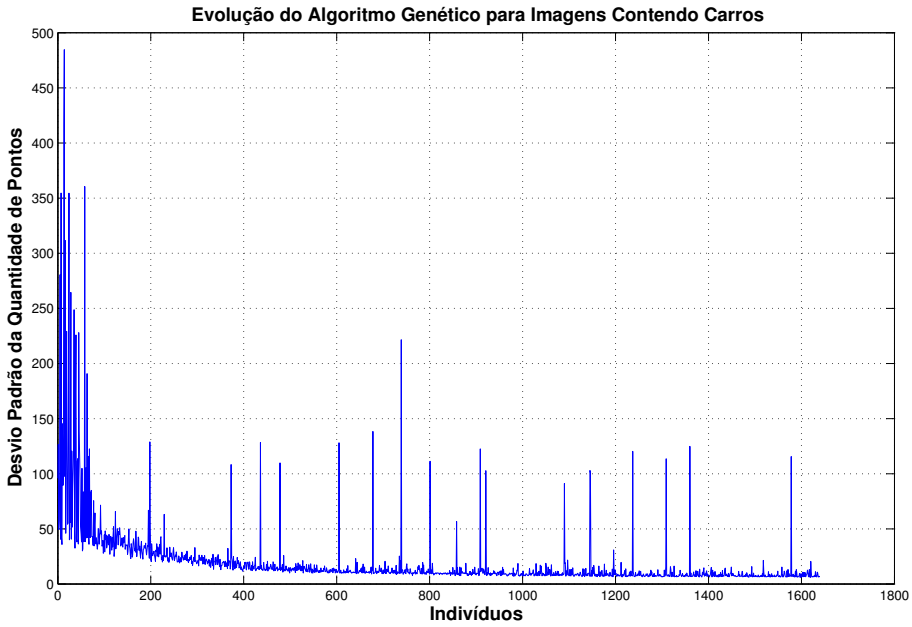


Figura C.4: Desvios-padrão para imagens contendo carros.

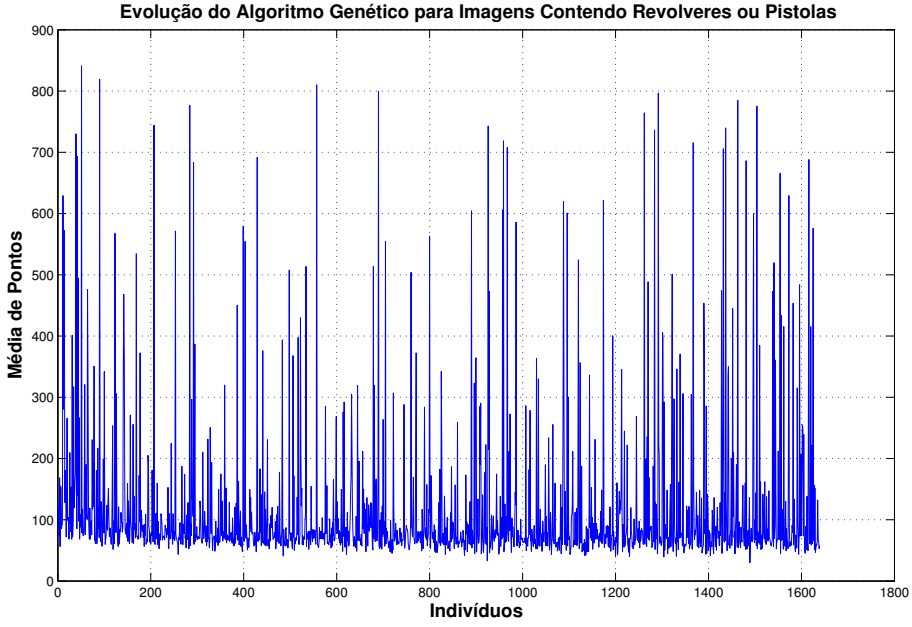


Figura C.5: Médias para imagens contendo pistolas.

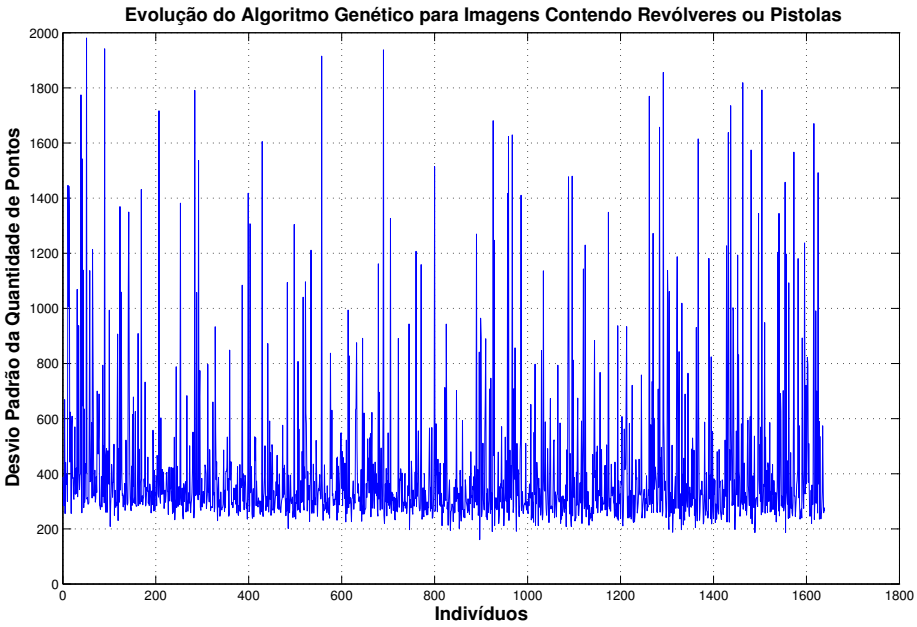


Figura C.6: Desvios-padrão para imagens contendo pistolas.

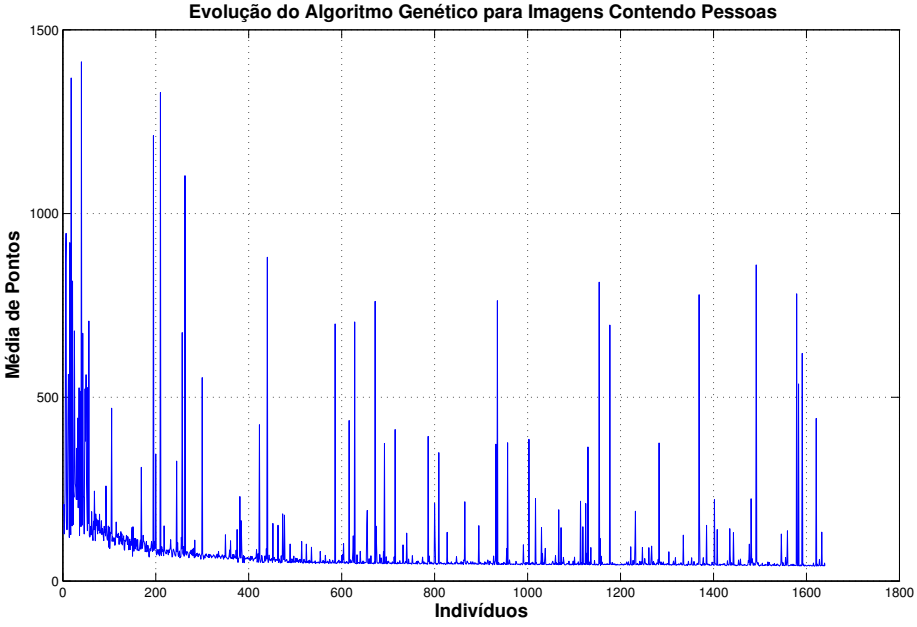


Figura C.7: Médias para imagens contendo faces de pessoas.

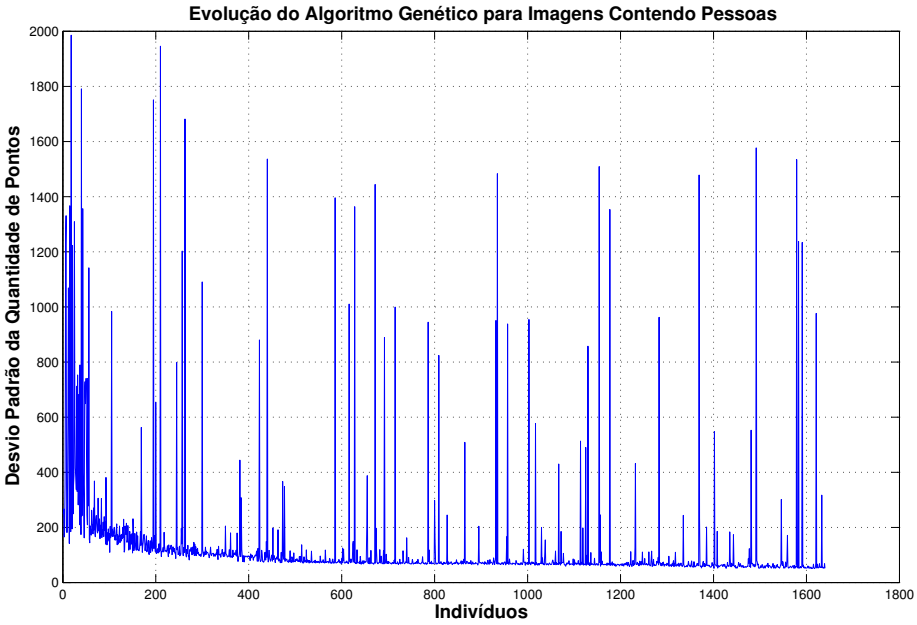


Figura C.8: Desvios-padrão para imagens contendo faces de pessoas.

Apêndice D

Amostra de Imagens Utilizadas

Abaixo são apresentadas amostras de imagens utilizadas nos processos de otimização e teste. Os testes foram realizados utilizando 100 imagens para cada classe. Nenhuma destas imagens do conjunto de teste foi utilizada no processo de otimização dos mapas de saliencias. As imagens estão organizadas como descrito a seguir. Para cada classe de imagens há duas figuras, a primeira exibe imagens utilizadas no processo de otimização e a segunda exibe as marcações dos cinco pontos mais salientes obtidos com o sistema de atenção visual otimizado por algoritmos genéticos. Todos os pontos salientes são mostrados com raio de inibição igual a 10. Os pontos salientes são representados por uma circunferência azul com raio 10 ao redor do ponto e as regiões selecionadas manualmente são representadas por retângulos pretos. As Figuras (D.1)-(D.8) mostram imagens contendo objetos genéricos, faces de pessoas, carros e pistolas, respectivamente.

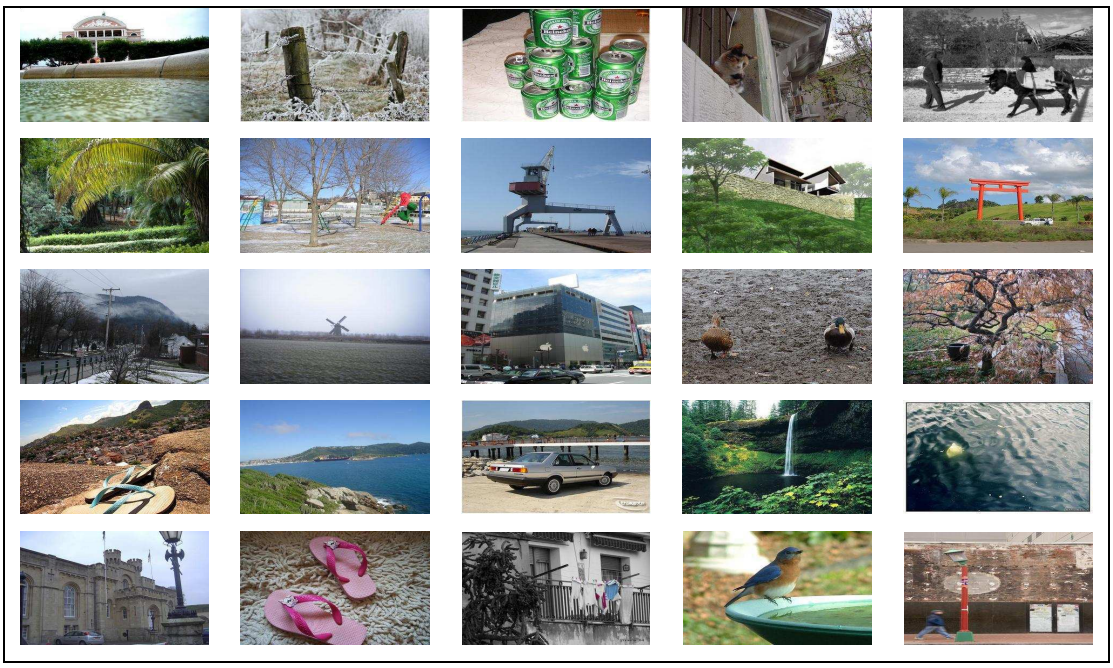


Figura D.1: Imagens contendo objetos genéricos utilizadas no processo de otimização.

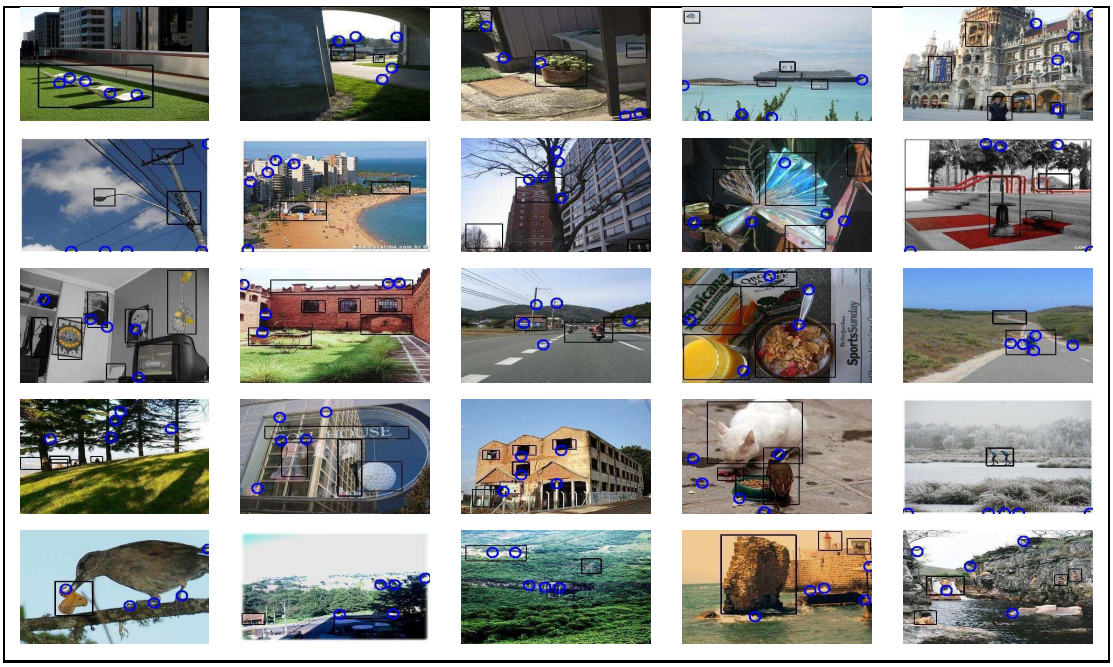


Figura D.2: Imagens contendo objetos genéricos com a marcação dos cinco pontos mais salientes obtidos com o sistema de atenção visual otimizado por algoritmos genéticos.

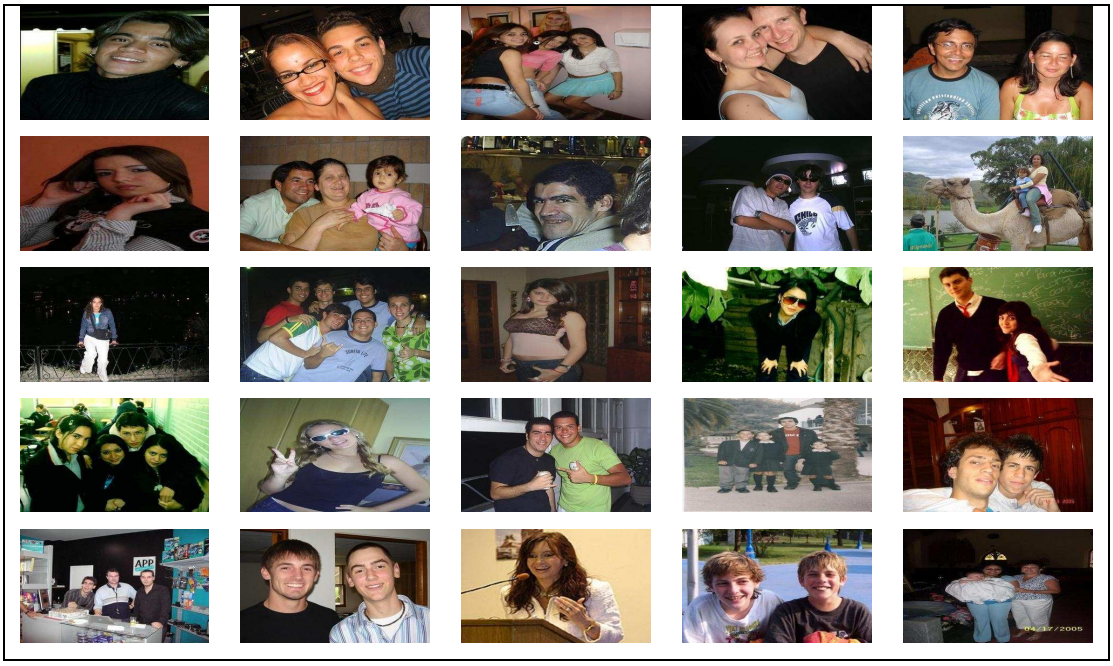


Figura D.3: Imagens contendo faces de pessoas utilizadas no processo de otimização.

Figura D.4: Imagens contendo faces de pessoas com a marcação dos cinco pontos mais salientes obtidos com o sistema de atenção visual otimizado.

Figura D.5: Imagens contendo carros utilizadas no processo de otimização.

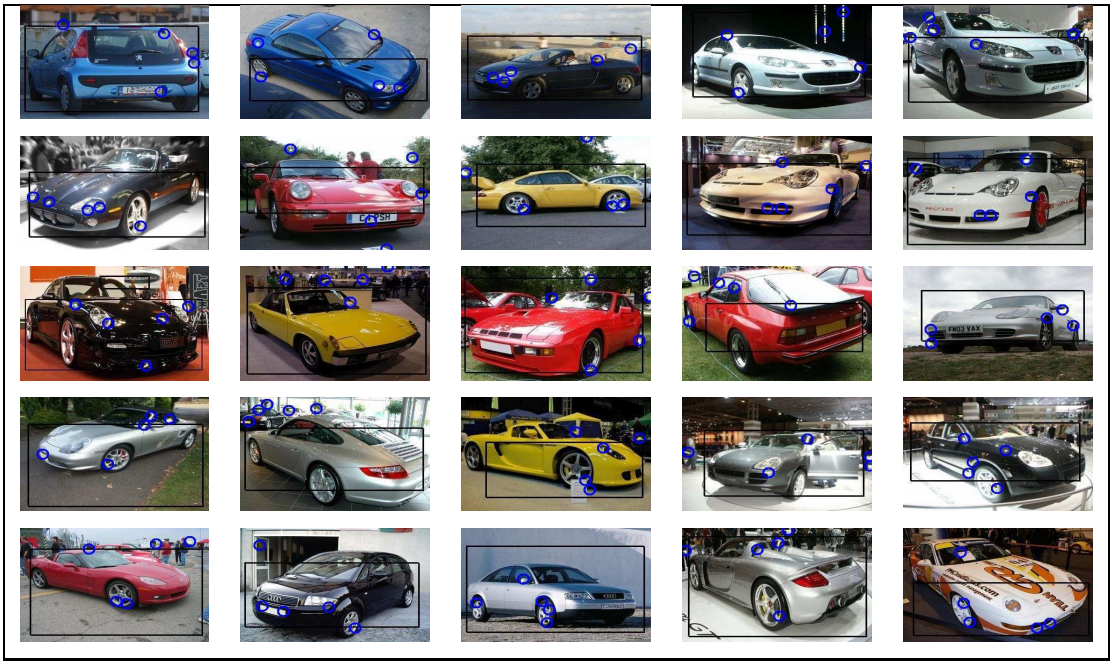


Figura D.6: Imagens contendo carros com a marcação dos cinco pontos mais salientes obtidos com o sistema de atenção visual otimizado.

Figura D.7: Imagens contendo armas utilizadas no processo de otimização.

Figura D.8: Imagens contendo pistolas ou revólveres com a marcação dos cinco pontos mais salientes obtidos com o sistema de atenção visual otimizado.