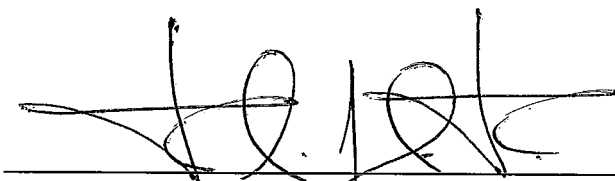


RECONHECIMENTO DE OBJETOS UTILIZANDO FILTROS ORIENTADOS

Ana Paula Tavares Leitão

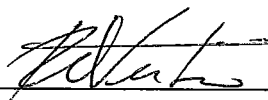
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:



Prof. Felipe Maia Galvão França, Ph.D.

Prof. Luís Alfredo Vidal de Carvalho, D.Sc.



Prof. Raul Queiroz Feitosa, Dr.Ing.

RIO DE JANEIRO, RJ - BRASIL
DEZEMBRO DE 1999

LEITÃO, ANA PAULA TAVARES

Reconhecimento de Objetos utilizando Fil-
tros Orientados [Rio de Janeiro] 1999

IX, 88 pp., 29.7 cm, (COPPE/UFRJ,
M.Sc., Engenharia de Sistemas e
Computação, 1999)

Tese – Universidade Federal do Rio de
Janeiro, COPPE

1 – Reconhecimento de Objetos

2 – Pirâmide de Imagens

3 – Rede neural Radial Basis Function

3 – Visão Computacional

I. COPPE/UFRJ II. Título (série)

*”Escrever é estar no extremo de si mesmo,
e quem está assim se exercendo nessa nudez,
a mais nua que há, tem pudor de que os outros
vejam o que deve haver de esgar, de tiques,
de gestos falhos, de pouco espetacular na torta visão
de uma alma no pleno extertor de criar.”*

João Cabral de Melo Neto

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

RECONHECIMENTO DE OBJETOS UTILIZANDO FILTROS ORIENTADOS

Ana Paula Tavares Leitão

Dezembro/1999

Orientador: Luiz Adauto Pessoa

Programa: Engenharia de Sistemas e Computação

Neste trabalho estamos propondo um novo sistema de reconhecimento de objetos baseado em um método de filtragem inspirado na biologia. Este sistema difere das propostas anteriores (1) pelo método de filtragem de vários estágios, (2) pela estrutura de pirâmide utilizada para representar a imagem, e, o aspecto mais importante, (3) pelo esquema de construção de protótipos para determinar os modelos armazenados na memória. O método é muito mais simples do que as propostas anteriores e tem um custo computacional relativamente inexpressivo, enquanto alcança taxas de erro em torno de 5% para reconhecimento das faces, um valor bem próximo aos melhores resultados publicados. O mesmo sistema é testado com objetos gerais, tão variados quanto carros e xícaras, com grande rotação em profundidade.

Ao final, implementamos um método bastante conhecido e amplamente utilizado para reconhecimento de objetos, denominado rede neural *Radial Basis Function*, utilizando o mesmo pré-processamento.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Masters of Science (M.Sc.)

OBJECT RECOGNITION USING ORIENTED FILTERS

Ana Paula Tavares Leitão

December/1999

Advisor: Luiz Adauto Pessoa

Department: Computing and Systems Engineering

In this work we propose a new object and face recognition system based on a biologically-inspired filtering method. Our work differs from previous proposals in (1) the multi-stage filtering method employed, in (2) the pyramid structure used, and most importantly, in (3) the prototype construction scheme to determine the models stored in memory. The method is much simpler than previous proposals and relatively inexpensive computationally, while attaining error rates as low as 5% for face recognition, very close to the best reported results. The same system was tested with a more general object data base, including objects, as different as cars and cups, in great depth rotations.

At the end, we implement a largely used neural network method, known as Radial Basis Function neural network, using the very same multi-stage filtering and pyramid structure.

AGRADECIMENTOS

A todo o amor e apoio de meus pais. O desejo deles de que o meu trabalho desse certo, muitas vezes foi maior do que o meu. Ter alguém acreditando em você é muito importante. Atender às suas expectativas nem sempre foi fácil.

Aos professores que, ao longo da minha carreira acadêmica, me ensinaram a desvendar e a me apaixonar por esta carreira.

Ao professor Felipe pela imensa oportunidade que está me oferecendo.

Ao professor Luís Alfredo, pelos conselhos sempre sábios.

A todas as amizades que conquistei e cultivei ao longo destes anos. A todos os que se deixaram cativar pela minha eterna alegria de viver. As longas conversas com a Nivea, os longos dias de trabalho com o Alexandre e as longas explicações do Sergio são inesquecíveis! Mas se eu for tentar enumerar todos os meus amigos, não vai sobrar espaço para uma tese...

Ao professor Luiz Adauto pela dedicação, pelo carinho e pelas broncas. O trabalho entre um aluno e um orientador é quase como um casamento, onde as duas partes tem que estar dispostas a se entender e a conversar sempre, buscando pontos em comum. Mas também tem muito da relação pai-filho, das preocupações, dos medos e da necessidade de se ensinar, de se abrir os caminhos para crescer e amadurecer. Olhando para o passado, sei que eu deveria ter feito várias coisas de outra forma, mas sei que nunca vou me arrepender das escolhas que fiz.

Índice

1	Introdução	1
2	Representação Interna dos Objetos	8
2.1	Formação da Imagem	9
2.2	Representação do Objeto	11
2.3	Detecção de Contorno	12
2.4	Representação Modelando Células do Sistema Visual	14
2.5	Pirâmides	21
3	Reconhecimento: Classificação e Identificação	28
3.1	Estrutura de Memória: Modelos	29
3.2	Categorização de Informação	31
3.3	Síntese de Modelos	34
4	Reconhecimento por Comparação	38
4.1	Faces	38
4.1.1	Testes e Resultados	40
4.2	Objetos Rotacionados	48
4.2.1	Testes e Resultados	48
5	Reconhecimento com Redes Neurais RBF	57
5.1	Outros Métodos	58
5.2	Radial Basis Function	59

5.2.1	Definições	60
5.2.2	Algumas Vantagens	63
5.3	Implementação	65
5.3.1	Unidades de Reconhecimento	65
5.3.2	Estabelecimento dos parâmetros	67
5.3.3	Treinamento	68
5.4	Simulações	69
6	Conclusões	72

Lista de Figuras

1.1	Analogia funcional entre (A) o Sistema Visual Humano e (B) o Sistema Visual Implementado. Podemos destacar as funções de (1) mapeamento detalhado da entrada e (2) comparação com os modelos pré-estabelecidos	4
1.2	Classificação da imagem de um objeto: a imagem é comparada a todos os modelos da memória para, depois, ser classificada pela categoria à qual pertence o modelo mais semelhante.	4
1.3	Variação de resolução. (A) Pirâmide da imagem de um objeto nas dimensões corretas; (B) A mesma pirâmide com os níveis menores ampliados para facilitar a visualização.	5
1.4	Imagens são mais parecidas em menor resolução. Pirâmides geradas para duas imagens do mesmo objeto. Os níveis menores foram ampliados para facilitar a comparação. Nestes, as faces se tornam cada vez mais parecidas.	6
2.1	Processo de percepção visual: A luz ilumina a cadeira e (1) é refletida para dentro dos olhos de uma pessoa, onde forma uma imagem na retina (2) gerando impulsos elétricos nos receptores (3). Os impulsos nervosos viajam ao longo das fibras nervosas (4), alcançando o córtex (5) onde são “processados” (6), e, então, a pessoa “vê” uma cadeira (7).	8

2.2	Câmara escura: Imagem projetada invertida e proporcional à distância do objeto.	10
2.3	Pixels: representação bidimensional discreta do objeto. Cada número indica a intensidade luminosa no “ponto”.	11
2.4	Filtro não isotrópico simples que detecta a variação de iluminação na direção horizontal. Note que o somatório dos valores positivos é igual ao negativo, garantindo que, ao ser aplicado sobre regiões uniformes, a resposta seja nula.	14
2.5	Filtro não isotrópico que detecta a variação de iluminação na direção diagonal. Note que o somatório dos valores positivos é igual ao negativo, garantindo que, ao ser aplicado sobre regiões uniformes, a resposta seja nula.	14
2.6	Filtro isotrópico que detecta a variação de iluminação, independente da direção. Note que o somatório dos valores positivos é igual ao negativo, garantindo que, ao ser aplicado sobre regiões uniformes, a resposta seja nula.	14
2.7	Seqüência de filtragens para gerar a representação da imagem, baseado no modelo hierárquico proposto por Hubel e Wiesel (ver referência no texto). A imagem é captada pelas células fotoreceptoras da retina e projetada para o Núcleo Geniculado Lateral (NGL) do tálamo. Esta informação será, depois, combinada em células simples e complexas do córtex.	15
2.8	Filtro Concêntrico: (A) Definição das regiões para o filtro <i>center-surround</i> .(B) Filtro Gaussiano de Diferença – isotrópico.	16
2.9	Filtro Gaussiano de Diferença: duas funções gaussianas de diferentes amplitudes e desvio padrão são combinadas numa nova função.	17

2.10	Filtros alongados em células simples de transição (A) claro-escuro e (B) escuro-claro, sensíveis a direção de contraste. Note que as áreas excitatórias e inibitórias são iguais, garantindo que, ao serem aplicados sobre regiões uniformes, a resposta seja nula. Não isotrópico.	18
2.11	Orientações dos filtros alongados (claro-escuro). Como os filtros são flexíveis, para cobrir razoavelmente todas as orientações, foi suficiente estabelecer quatro orientações para cada tipo de célula simples. Para calcular estas novas orientações, basta aplicar a matriz de rotação adequada. Para obter os outros filtros, basta inverter as regiões excitatória e inibitória.	19
2.12	Funções gaussianas, deslocadas de um <i>off-set</i>, são combinadas numa nova função utilizada para a definição dos filtros de células simples.	19
2.13	Filtro alongado em uma célula complexa. A combinação das respostas das células simples de mesma orientação é insensível a direção da transição do contraste.	20
2.14	Mapa de respostas das células complexas com o somatório de todas as orientações.	20
2.15	Exemplo do resultado final de uma filtragem mostra a representação da imagem através das respostas de células complexas.	21
2.16	Pirâmides: (A) Pirâmide de uma imagem utilizando apenas redução de dimensão. (B) Pirâmide da mesma imagem com os dois passos de construção: filtragem e redução de dimensão.	22
2.17	Variação da resolução a cada nível: (A) Pirâmide da imagem de um objeto nas dimensões corretas. (B) Pirâmide da mesma imagem com os níveis mais baixos ampliados para melhor visualização.	23

2.18	Construção da Pirâmide. F é a dimensão do filtro conforme a Figura 2.19, a seguir. $d(imagem)$ é uma função que retorna o <i>down-sampling</i> de uma imagem, uma redução nas dimensões desta.	24
2.19	Proporção dos filtros. Este esquema mostra que as dimensões dos filtros são dobradas para cada novo nível	24
2.20	Redução de dimensão ou <i>downsampling</i> . Calcula a média aritmética das intensidades de uma máscara 2x2 de pixels gerando um único pixel médio.	25
2.21	Pirâmides do gato e do bocal. Os níveis menores (à direita) foram ampliados para melhor comparação. Mesmo nos menores níveis a distinção ponto a ponto entre os objetos parece clara.	26
2.22	Pirâmides do gato com rotações diferentes. Os níveis menores (à direita) foram ampliados para melhor comparação. Nos menores níveis a semelhança ponto a ponto se torna cada vez mais clara.	27
3.1	Modelos para <i>todas</i> as imagens de cada objeto. Se todas as poses forem armazenadas na memória, o sistema reconhece sempre pois encontra um modelo 100% semelhante. Porém, isto é inviável.	30
3.2	Um modelo combinando várias imagens. Podemos armazenar uma combinação de várias imagens de um mesmo objeto.	30
3.3	Modelos com combinação de imagens de cada objeto. O sistema escolhe o modelo mais parecido com a imagem. Se os modelos forem suficientemente representativos, a resposta será o objeto contido na imagem.	31
3.4	Esquema da rede ART Fuzzy. A informação de entrada percorre a rede em direção da camada de saída para ser calculada a semelhança com o modelo de cada categoria, armazenado no vetor de pesos.	32

3.5	Primeiro sistema de classificação. A categoria escolhida é a mais ativa segundo alguma medida de similaridade. O vetor de pesos armazena o modelo que representa as categorias.	34
3.6	Segundo sistema de classificação: A categoria escolhida, a mais ativa segundo alguma medida de similaridade, define a resposta do sistema, ativando o objeto classificado. O vetor de pesos entre as camadas de entrada e de saída intermediária armazena o modelo que representa as categorias.	35
3.7	Combinação de 3 imagens filtradas. Na primeira linha, temos um intervalo de 3 imagens filtradas, e na segunda, os modelos mínimo e média, respectivamente.	36
3.8	Combinação de 5 imagens filtradas. Na primeira linha, temos um intervalo de 5 imagens filtradas, e na segunda, os modelos mínimo e média, respectivamente.	36
3.9	Combinação de 7 imagens filtradas. Na primeira linha, temos um intervalo de 7 imagens filtradas, e na segunda, os modelos mínimo e média, respectivamente.	37
4.1	Todas as faces do Laboratório Olivetti, disponível em http://www.camorl.co.uk/facedatabase.html	39
4.2	Imagens padrão – caso 1: variabilidade da base de dados representada por uma pose de cada pessoa.	40
4.3	Imagens padrão – caso 2: variabilidade da base de dados representada por uma outra pose de cada pessoa.	41
4.4	Construção dos modelos de mínimo e média, respectivamente, de <i>todas</i> as 10 imagens pré-processadas para esta pessoa apresentada na primeira linha.	42

4.5	Exemplo de construção dos modelos de mínimo e média, respectivamente, para o grupo de 7 imagens apresentadas na primeira linha, depois de pré-processadas.	42
4.6	Exemplo de construção dos modelos de mínimo e média, respectivamente, para o grupo de 5 imagens apresentadas na primeira linha, depois de pré-processadas.	43
4.7	Exemplo de construção dos modelos de mínimo e média, respectivamente, para o grupo de 3 imagens apresentadas na primeira linha, depois de pré-processadas.	43
4.8	Taxa de erro <i>versus</i> tamanho do grupo para mínimos e médias utilizando o <i>produto interno normalizado</i> como medida de similaridade. Estes gráficos representam uma curva para cada nível da pirâmide, conforme os resultados das Tabelas 4.3 e 4.4	44
4.9	Taxa de erro <i>versus</i> tamanho do grupo para mínimos e médias utilizando a <i>função de ativação da rede ART</i> como medida de similaridade. Estes gráficos representam uma curva para cada nível da pirâmide, conforme os resultados das Tabelas 4.5 e 4.6	44
4.10	Todos os objetos da Universidade de Columbia. A base é formada por 72 imagens de cada objeto, variando rotação em profundidade e, para o caso dos objetos mais alongados (como os carros) com pequenos ajustes de escala. Ver Figura 4.12 a seguir.	49
4.11	Rotações de 25 em 25 graus. Na primeira linha, as imagens originais e na segunda linha, as imagens utilizadas como entrada no sistema, filtradas conforme as definições do Capítulo 2.	49

4.12	Rotações de 5 em 5 graus. No carro, um objeto alongado, a imagem sofre alterações de escala. Por outro lado, na embalagem de talco, um objeto simétrico em relação ao eixo de rotação, a escala se matem fixa.	50
4.13	Escolha de imagens para construção de cada modelo $M_{i,o}$. As imagens que não são escolhidas para compor os modelos são usadas para teste. Os dois parâmetros tg (tamanho do grupo) e ts (tamanho do salto) definem, respectivamente, quantas imagens são usadas para a construção dos modelos e para teste, e, portanto, o número de modelos para cada objeto.	50
4.14	Modelo esperado. Construção de modelos a partir de um conjunto de imagens. Conjunto de teste é composto pelas imagens não utilizadas. A resposta esperada para cada teste é o modelo mais próximo da imagem. Considerando que nesta base de dados os objetos estão dispostos de 5 em 5 graus, o mais próximo deve ser sempre um dos modelos adjacentes na seqüência.	51
4.15	Taxa de erro <i>versus</i> tamanho do grupo utilizando modelos de mínimos e médias em um conjunto de teste com intervalos de tamanho 2 . Cada curva representa um nível da pirâmide, ou seja, uma diferente resolução dos dados.	52
4.16	Taxa de erro <i>versus</i> tamanho do grupo utilizando modelos de mínimos e médias em um conjunto de teste com intervalos de tamanho 4 . Cada curva representa um nível da pirâmide, ou seja, uma diferente resolução dos dados.	53

4.17	Taxa de erro <i>versus</i> tamanho do grupo utilizando modelos de mínimos e médias em um conjunto de teste com intervalos de tamanho 6 . Cada curva representa um nível da pirâmide, ou seja, uma diferente resolução dos dados.	54
5.1	Organização de uma rede neural RBF: o aprendizado dos pesos entre as unidades intermediária e de saída (linear) é supervisionado, enquanto que, entre as unidades de entrada e intermediárias (radial) é estabelecido pelos ajustes imediatos dos parâmetros das funções radiais a partir do conjunto de treinamento.	60
5.2	Organização de uma rede neural RBF: A saída da primeira camada é dada pela aplicação de uma função radial ao resultado da diferença entre a informação de entrada e um vetor de centros c_h definido durante o treinamento. A resposta da camada de saída é uma soma ponderada da saída da camada intermediária pelos vetores de peso, também estabelecidos durante o treinamento.	62
5.3	Funções radialmente simétricas e estritamente positiva com um único valor máximo no seu centro c_h e que decai rapidamente para zero ao se afastar do mesmo. São sugeridas por Orr (ver citação no texto) para serem empregadas nas camadas intermediárias de redes neurais RBF. A função mais utilizada é a Gaussiana.	64
5.4	Sistema combinado utilizando uma RBF para cada objeto. A unidade com a maior saída classifica a entrada.	66
5.5	Unidade de reconhecimento uma rede RBF para cada objeto, cuja saída será comparada com as demais.	67

5.6	Conjunto de treinamento de uma unidade de reconhecimento. Para cada imagem do conjunto de treinamento é escolhido um contra-exemplo (imagem mais semelhante ao exemplo). Como pode ser visto, todas as imagens são filtradas conforme a representação interna definida.	67
6.1	Transformações da base ORL: pequenas rotações planares e em profundidade, pequena variação de escala, variação de expressão e acessórios.	74

Lista de Tabelas

4.1	Taxas de erro para modelos com uma única pose , usando as duas medidas de similaridade.	41
4.2	Taxa de erro para mínimo e média de <i>todas</i> as imagens , usando apenas a função de ativação como medida de similaridade. . .	42
4.3	Taxa de erro para mínimos . Média dos resultados de todas as combinações, usando o <i>produto interno normalizado</i> como medida de similaridade.	45
4.4	Taxa de erro para médias . Média dos resultados de todas as combinações, usando o <i>produto interno normalizado</i> como medida de similaridade.	45
4.5	Taxa de erro para mínimos . Média dos resultados de todas as combinações, usando a <i>função de ativação</i> como medida de similaridade.	45
4.6	Taxa de erro para médias . Média dos resultados de todas as combinações, usando a <i>função de ativação</i> como medida de similaridade.	45
4.7	Menores taxas de erro para mínimos , a melhor combinação para cada nível da pirâmide com 7, 5, 3 e 1 imagens, usando o <i>produto interno normalizado</i> como medida de similaridade.	46
4.8	Menores taxas de erro para médias , a melhor combinação para cada nível da pirâmide com 7, 5, 3 e 1 imagens, usando o <i>produto interno normalizado</i> como medida de similaridade.	46

4.9	Menores taxas de erro para mínimos , a melhor combinação para cada nível da pirâmide com 7, 5, 3 e 1 imagens, usando a <i>função de ativação</i> como medida de similaridade.	46
4.10	Menores taxas de erro para médias , a melhor combinação para cada nível da pirâmide com 7, 5, 3 e 1 imagens, usando a <i>função de ativação</i> como medida de similaridade.	47
4.11	Tabela comparativa apresentada inicialmente no trabalho de Lawrence <i>et. al.</i> . São apresentados os resultados utilizando 1, 3 ou 5 imagens para o estabelecimento do modelo de memória. Estamos incluindo o nosso melhor resultado (dentre todas as combinações), que foi obtido utilizando média como forma de combinação, com a resolução do nível 2 da pirâmide de imagens e tendo como medida de similaridade a função de ativação.	47
4.12	Taxas de erro para os testes com modelos de <i>Mínimos</i> de objetos rotacionados (cada modelo é uma categoria) . Os resultados estão separados conforme o nível da pirâmide utilizado para o teste e organizados pelo tamanho do grupo (<i>tg</i>) e pelo tamanho do salto (<i>ts</i>).	52
4.13	Taxas de erro para os testes com modelos de <i>Médias</i> de objetos rotacionados (cada modelo é uma categoria) . Os resultados estão separados conforme o nível da pirâmide utilizado para o teste e organizados pelo tamanho do grupo (<i>tg</i>) e pelo tamanho do salto (<i>ts</i>).	53

4.14	Taxas de erro para os testes com modelos de <i>Mínimos</i> de objetos rotacionados. Independente da pose da imagem modelo. Os resultados estão separados conforme o nível da pirâmide utilizado para o teste e organizados pelo tamanho do grupo (<i>tg</i>) e pelo tamanho do salto (<i>ts</i>).	55
4.15	Taxas de erro para os testes com modelos de <i>Médias</i> de objetos rotacionados. Independente da pose da imagem modelo. Os resultados estão separados conforme o nível da pirâmide utilizado para o teste e organizados pelo tamanho do grupo (<i>tg</i>) e pelo tamanho do salto (<i>ts</i>).	56
5.1	Taxa de erro: média dos resultados de todas as combinações selecionando apenas 3 objetos da base de dados.	71
5.2	Taxa de erro: média dos resultados de todas as combinações selecionando apenas 5 objetos da base de dados.	71
5.3	Taxa de erro: média dos resultados de todas as combinações selecionando apenas 7 objetos da base de dados.	71
6.1	Taxas de erro para múltiplos modelos independentes. Resultado de uma única combinação.	76
6.2	Taxas de erro para um único modelo combinado pela média. Resultado da mesma combinação anterior.	76

Capítulo 1

Introdução

Entender, modelar e reproduzir as funções desempenhadas pelo cérebro é um dos nossos maiores desafios. Descobrir quais partes correspondem à consciência, ao aprendizado ou à memória, envolve pesquisas nas mais diversas áreas: da neurologia à psicologia e, até mesmo, à filosofia. Por enquanto, a neurobiologia começa a entender as funções elementares do cérebro, mas ainda descreve vagamente as funções mais simples como enxergar formas e cores, ou como interpretar sons.

A visão é o sentido mais desenvolvido no ser humano. A maior parte do nosso conhecimento do mundo exterior é absorvido por ele. No córtex cerebral, o córtex visual ocupa uma região relativamente grande, Zeki [66], mostrando a grande importância deste sistema.

O reconhecimento de objetos é um dos aspectos mais importantes da percepção visual. A capacidade de reconhecimento e classificação dos sistemas visuais biológicos é muito superior à dos sistemas artificiais. Ao nosso redor, observamos um mundo que se modifica a todo instante. Um objeto não é visto numa única posição, mas em diferentes ângulos, distâncias, ambientes e condições de iluminação. Entretanto, nosso cérebro é capaz de identificá-lo como um objeto único. Denomina-se constância ou invariância esta capacidade de descartar as mudanças para analisar apenas a parte fixa da informação. Juntamente com outras propriedades do sistema visual humano, a constância colabora para tornar a captação da essência de um objeto, uma tarefa imediata. No entanto, isto é muito difícil num sistema artificial.

Neste caso, essência é o conjunto das informações suficientes para distinguir um objeto dos demais, definindo-o como único.

Vários pesquisadores, seguindo o trabalho de Marr [36], acreditavam que o produto mais perfeito de qualquer sistema visual seria algum tipo de reconstrução tridimensional do ambiente. Porém, uma vez que foram feitos testes com representações tridimensionais, tornou-se claro que representações mais simples do que uma reconstrução completa deveriam ser mais apropriadas.

Pode parecer mais natural formar representações de objetos em 3D do que em 2D, dado a nossa aparente habilidade para visualizar manipulação tridimensional dos objetos. Mas isto pode ser uma confusão de níveis de cognição. Experimentos psico-físicos [7, 45] sugerem que a generalização no reconhecimento de poses não familiares se baseia na interpolação entre as imagens tridimensionais armazenadas. Embora sejamos capazes de pequena manipulação na visualização mental, é bem possível que nosso processamento diário de objetos seja feito usando representações mais simples.

Estudos sobre processos envolvidos na percepção visual destacam uma hierarquia entre eles, entretanto, não há uma distinção muito clara entre os possíveis níveis formados [60]. Decidimos adotar uma definição bastante grosseira dividindo o sistema visual em duas fases: “ver” e “entender” a imagem. A primeira é uma analogia aos primeiros níveis do sistema visual, onde a informação captada pela retina é mapeada detalhadamente nos estágios iniciais de processamento. Este mapeamento é estabelecido por uma série de *filtragens* aplicadas à imagem, emulando, através de modelos computacionais [23, 49], o comportamento de determinadas células do nosso córtex visual, criando uma *representação interna* pela conversão e combinação dos dados da imagem.

No sistema visual biológico, os mapas criados nas primeiras camadas do córtex visual acabam mapeados no córtex visual “associativo”, nos níveis mais altos do

sistema. Neste nível, supõe-se que as impressões visuais recebidas sejam associadas a impressões anteriores do mesmo tipo (os *modelos* previamente construídos), gerando reconhecimento [12, 42, 43]. Aproveitando esta idéia, na segunda fase do nosso sistema, tentamos decidir se uma imagem que estamos observando corresponde a um objeto visto anteriormente, um objeto conhecido. Inicialmente, parece que basta ter memória suficientemente grande e eficiente para resolver o problema. Porém, não é possível armazenar todas as imagens de cada objeto. Outra solução seria estabelecer um número suficiente de diferentes imagens a serem armazenadas, para cada objeto, de tal forma que, uma nova imagem seria reconhecida ao ser comparada a todas as anteriores. Este tipo de mecanismo é conhecido como *memória associativa* e foi proposto para implementar uma abordagem de reconhecimento “direto”. Nele é estabelecida uma medida de similaridade entre a imagem de entrada e cada uma das imagens armazenadas. Este mecanismo, que é usualmente aproximado por redes neurais, pode armazenar um grande número de padrões (P_1, P_2, \dots, P_n) , para então, dado um padrão de entrada Q , recuperar o padrão P_i mais semelhante a Q . Assim, quando o sistema armazena uma pose ou um conjunto de poses representativas para cada objeto, pode, automaticamente, encontrar a representação mais semelhante a qualquer nova entrada apresentada.

Neste trabalho, criamos uma “memória” composta por diversas imagens previamente armazenadas em *modelos* (Figura 1.1). O reconhecimento se dá quando uma dada imagem de um objeto é classificada por um modelo que codifique imagens deste mesmo objeto. Podemos supor que dispomos de uma memória que armazene modelos de apenas dois objetos: casas e carros, como na Figura 1.2. Ao apresentarmos uma imagem, a mesma será reconhecida como um carro se o sistema for capaz de classificá-la como um modelo da classe de carros. Ou seja, se a imagem mais semelhante à que foi apresentada for um dos modelos que definem esta classe.

Por maiores e mais sofisticadas que se tornem as memórias dos computadores,

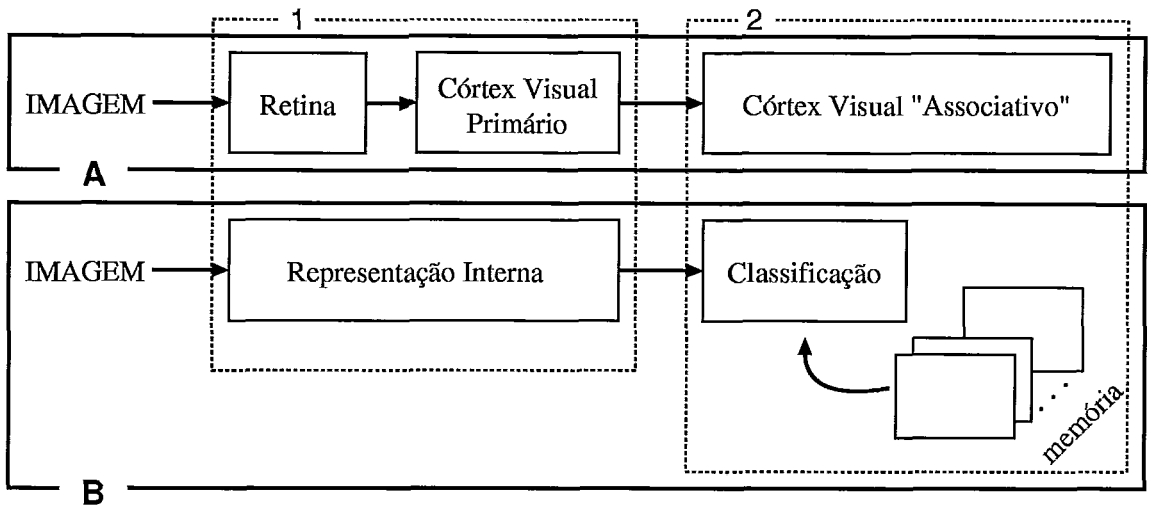


Figura 1.1: **Analogia funcional** entre (A) o Sistema Visual Humano e (B) o Sistema Visual Implementado. Podemos destacar as funções de (1) mapeamento detalhado da entrada e (2) comparação com os modelos pré-estabelecidos

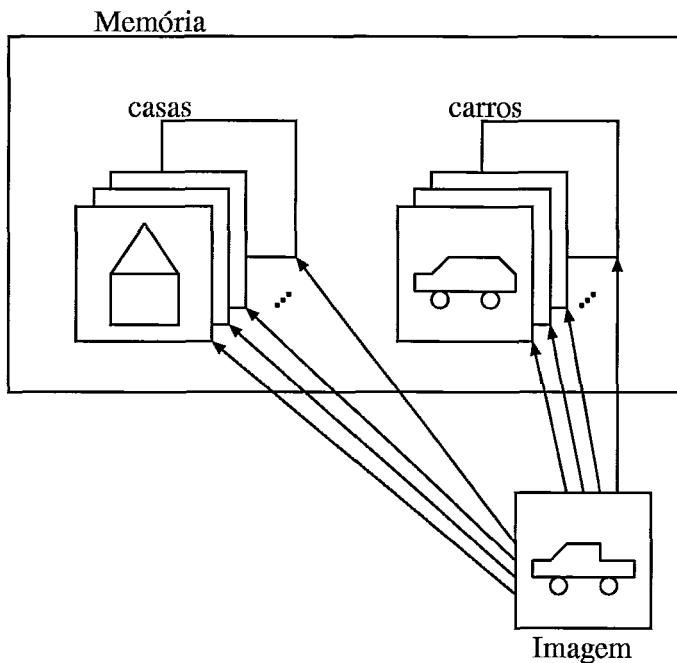


Figura 1.2: **Classificação da imagem de um objeto:** a imagem é comparada a todos os modelos da memória para, depois, ser classificada pela categoria à qual pertence o modelo mais semelhante.

sempre haverá um limite para a quantidade de informação armazenada. Duas idéias são discutidas neste trabalho para tentar amenizar este problema. A primeira seria criar modelos que sejam combinações de várias imagens de um mesmo objeto, diminuindo o espaço ocupado pela memória, sem alterar o número de poses repre-

sentadas. Para tal suposição, é necessário descobrir qual forma de combinação é representativa o suficiente para todas as poses que armazena, tal que, outras imagens do mesmo objeto possam ser reconhecidas. A outra proposta para reduzir a memória vem da estrutura de dados escolhida para armazenar as imagens filtradas: as *pirâmides*. Nesta estrutura, a mesma imagem é armazenada diversas vezes, em níveis de resolução variados. Tendo a imagem filtrada como primeiro nível, cada nível seguinte tem as informações cada vez menos detalhadas e com as dimensões cada vez menores (Figura 1.3). Assim, a imagem formada é cada vez mais grosseira, permitindo maior semelhança entre imagens com pequenas distorções (Figura 1.4), facilitando o reconhecimento e, potencialmente, diminuindo o número de poses necessárias. Devemos lembrar também, que imagens menores ocupam menos espaço e requerem menor custo computacional para serem processadas.

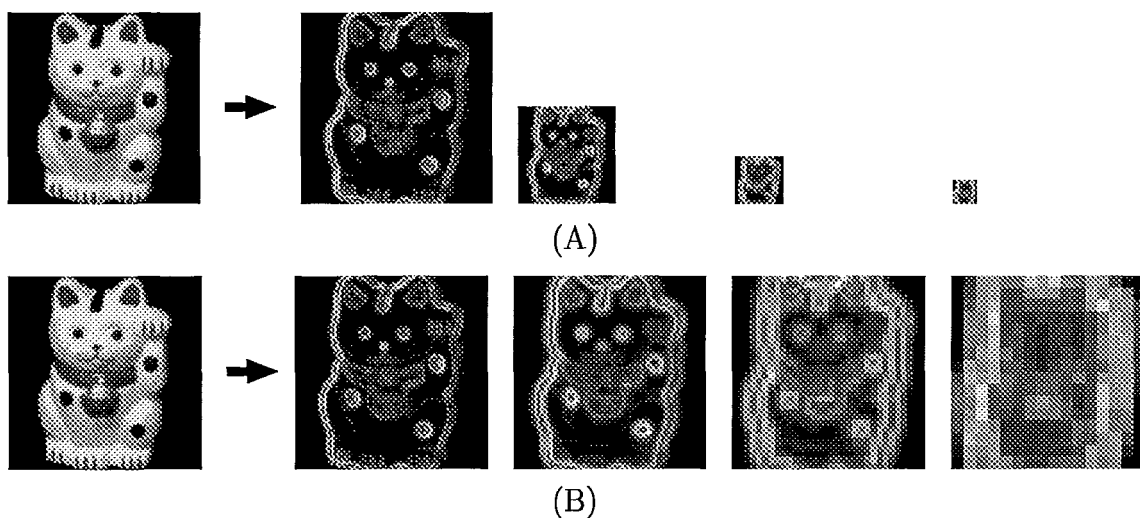


Figura 1.3: **Varição de resolução.** (A) Pirâmide da imagem de um objeto nas dimensões corretas; (B) A mesma pirâmide com os níveis menores ampliados para facilitar a visualização.

Além disso, um tipo de rede neural *feed forward* denominada *Radial Basis Function*, RBF, foi identificada como um modelo adaptativo eficiente por vários pesquisadores [5, 14, 18, 26, 51, 52]. Suas maiores vantagens são a simplicidade computacional, suportada por uma teoria matemática bem definida e uma capacidade de generalização robusta, poderosa o suficiente para atender a tarefas reais.

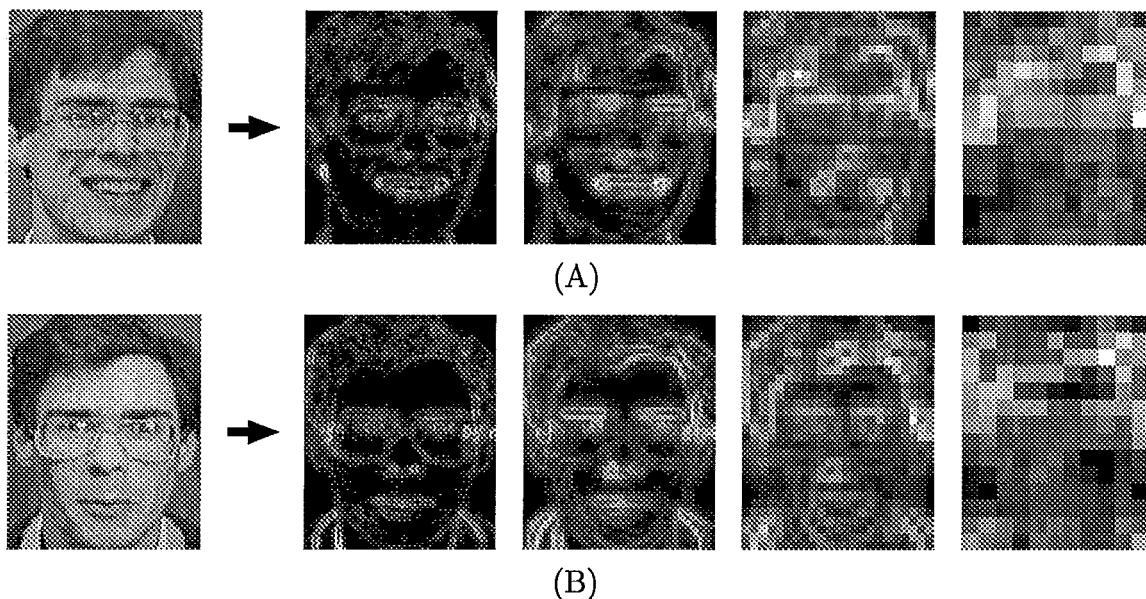


Figura 1.4: **Imagens são mais parecidas em menor resolução.** Pirâmides geradas para duas imagens do mesmo objeto. Os níveis menores foram ampliados para facilitar a comparação. Nestes, as faces se tornam cada vez mais parecidas.

Estas redes foram identificadas como ideais para aplicações práticas de problemas visuais por Girosi [19], pois são eficientes para tratar com dados esparsos de alta dimensionalidade (como imagens) e porque interpolam algumas imagens de treinamento, criando funções de aproximação para estes dados, que contornam os ruídos geralmente apresentados em dados reais.

No próximo Capítulo, após uma discussão sobre as diferentes formas de armazenamento de dados encontrados na literatura, abordamos a representação das imagens escolhidas, baseada nos trabalhos de Grossberg e Pessoa [22] e de Pessoa *et al.* [49]. A construção das pirâmides, propostas inicialmente por Burt e Adelson [1, 8, 9], é adaptada para os filtros de inspiração biológica definidos pela representação das imagens.

Para definir as estruturas de memória no Capítulo 3, voltamos à discussão da minimização da informação armazenada, sem perder, entretanto, a capacidade de reconhecimento. Mostramos que, pela categorização de informação, como nas redes neurais de aprendizado não supervisionado [10, 11, 22, 33, 46], diversas imagens de um mesmo objeto podem ser resumidas como o padrão de uma categoria. Desta

idéia, surge a proposta de sintetizarmos padrões, combinando várias imagens do objeto que queremos representar. Dois sistemas são propostos para armazenar estes padrões e testá-los com novas imagens.

No Capítulo 4, apresentamos os testes definidos para as simulações destes sistemas, bem como seus resultados. Para *conjuntos fechados*¹ de objetos, selecionamos algumas imagens de um mesmo objeto para a construção dos padrões, enquanto o restante é utilizado para testar a capacidade de classificação destes. Para fazer esta classificação, contruímos um sistema composto de uma camada de entrada e uma camada de saída (classes) interligadas por vetores de peso. Estes vetores são os protótipos sintetizados para cada classe, conforme as definições do Capítulo 3. Como medida de similaridade entre o protótipo e a entrada apresentada ao sistema, utilizamos tanto a correlação pelo produto interno quanto a função de ativação utilizada em redes ART Fuzzy [10, 11]. O segundo sistema atua de forma análoga, sendo que, a saída é analisada em uma camada posterior, que combina os vários modelos criamos para um determinado objeto.

No quinto Capítulo, utilizando a mesma base de dados de faces, selecionamos um conjunto de poses para treinamento de um sistema mais sofisticado de classificação: a rede neural *Radial Basis Function* [24, 25]. Neste sistema, para cada objeto criamos uma unidade de reconhecimento, onde a classificação é feita por uma RBF composta de uma camada de entrada, uma camada intermediária (onde se encontram as funções radiais) e uma camada de saída (com uma única unidade). A unidade de reconhecimento com a maior resposta para a imagem de entrada apresentada é a resposta do sistema.

O sexto e último Capítulo resume as propostas apresentadas, os resultados e discute algumas conclusões, além de sugerir novas questões a serem abordadas.

¹As imagens de todos os objetos são utilizadas nas duas fases do processo, ou seja, não existem objetos “desconhecidos” a serem classificados.

Capítulo 2

Representação Interna dos Objetos

Para melhor entender como criamos o sistema de reconhecimento de objetos a partir de suas imagens é interessante fazer, primeiramente, uma analogia com o processo que se desencadeia em nosso cérebro quando “vemos” um objeto. Goldstein [21] apresenta um esquema definindo alguns passos da percepção visual, como pode ser acompanhado pela Figura 2.1. Depois que a luz refletida no objeto penetra os olhos de uma pessoa, uma seqüência de processamentos é executada para destacar o objeto do fundo, definir suas formas, texturas, cores, etc. O conjunto das informações extraído pode ser considerado a *representação interna* do objeto.

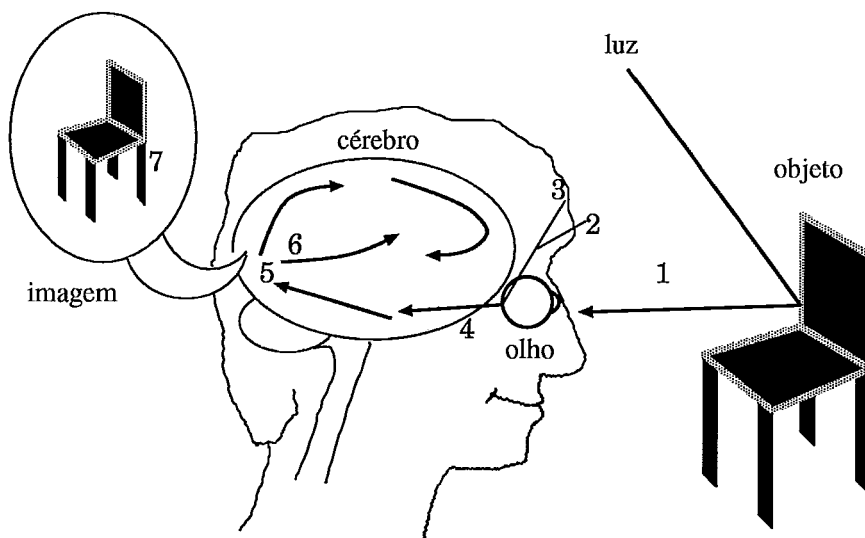


Figura 2.1: **Processo de percepção visual:** A luz ilumina a cadeira e (1) é refletida para dentro dos olhos de uma pessoa, onde forma uma imagem na retina (2) gerando impulsos elétricos nos receptores (3). Os impulsos nervosos viajam ao longo das fibras nervosas (4), alcançando o córtex (5) onde são “processados” (6), e, então, a pessoa “vê” uma cadeira (7).

Após analisarmos alguns aspectos geométricos da formação da imagem, vamos apresentar diversas representações encontradas na literatura. Podemos ressaltar, no entanto, que algumas representações foram definidas buscando otimizar o processamento computacional, enquanto outras buscam reproduzir o que ocorre em nosso córtex visual. Ao final, mostraremos a representação escolhida para este trabalho.

2.1 Formação da Imagem

Segundo Russel [56], o processo de formação da imagem pode ser bem descrito por seus aspectos geométricos e físicos. Dada a descrição de uma cena em 3D, podemos facilmente produzir uma foto (uma representação 2D) a partir de uma câmera devidamente posicionada. No entanto, inverter este processo para chegar à descrição de uma cena a partir de uma imagem se mostra bem mais difícil.

A geometria do processo de formação da imagem pode ser explicada por uma câmera escura onde a luz entra atravessando um *pinhole*, um furo muito pequeno (ver Figura 2.2) [41, 56]. Seja P um ponto na cena com coordenadas (X, Y, Z) e P' a sua projeção no plano da imagem com coordenadas (x, y, z) . Sendo f a distância entre o furo e o plano da imagem, por similaridade de triângulos, podemos derivar as seguintes equações:

$$\frac{-x}{f} = \frac{X}{Z}; \frac{-y}{f} = \frac{Y}{Z} \quad (2.1)$$

ou seja,

$$x = \frac{-fX}{Z}; y = \frac{-fY}{Z} \quad (2.2)$$

Note que a imagem formada é invertida, tanto vertical quanto horizontalmente. Estas equações definem um processo de formação de imagem denominado *projeção em perspectiva*.

A ferramenta mais comum para captação de imagem é a câmera digital. Esta, tal como os olhos dos vertebrados, utiliza *lentes* para entrada da luz. Entretanto, uma

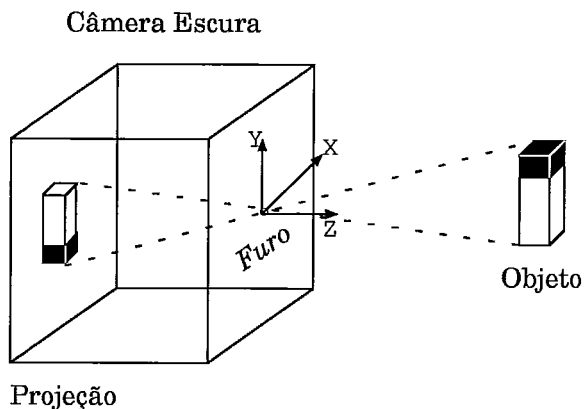


Figura 2.2: **Câmara escura:** Imagem projetada invertida e proporcional à distância do objeto.

lente é muito maior que um furo, deixando passar mais luz. Com isto, nem toda a imagem pode estar em foco ao mesmo tempo. Para focalizar objetos em diferentes distâncias, as lentes dos olhos mudam de forma enquanto as lentes de uma câmera se move na direção Z (conforme as coordenadas descritas na Figura 2.2).

Nos olhos e nas câmeras, o plano da imagem é subdividido em *pixels*. Numa câmera digital, estes pixels estão normalmente organizados em uma espécie de malha retangular (Figura 2.3). Nos olhos, temos 120×10^6 bastonetes e 6×10^6 cones organizados em um mosaico hexagonal. Nos dois casos, podemos modelar o sinal detectado no plano da imagem pela variação da intensidade luminosa no tempo: $I(x, y, t)$. Como não estamos interessados em abordar aspectos temporais, como movimento, vamos interpretar apenas fotos estáticas que serão representadas por $I(x, y)$, a intensidade luminosa em cada ponto no plano da imagem. O reconhecimento de objetos em 3D é dificultado, pois sua aparência em imagens bidimensionais varia muito quando observados em diferentes condições luminosas ou em diferentes ângulos, principalmente, quando a auto-occlusão esconde características importantes para a correta identificação do objeto. Por exemplo, quando vemos um rosto de perfil, uma das orelhas desaparece. Na próxima seção, vamos apresentar diversas formas de representação internas, extraídas a partir desta primeira imagem de entrada.

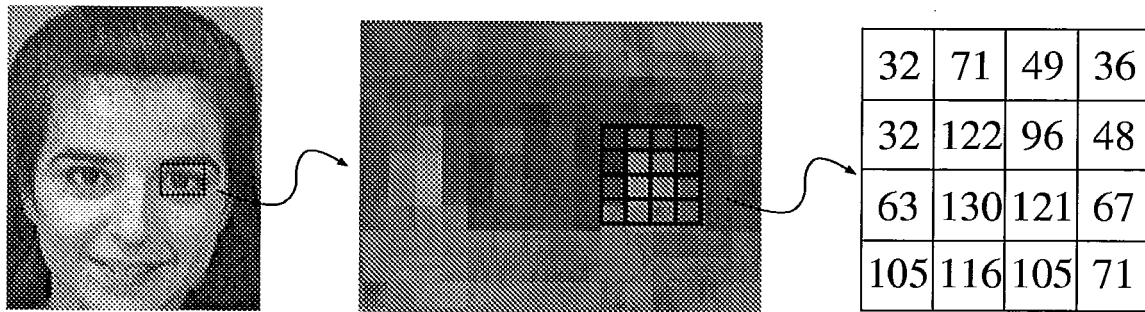


Figura 2.3: **Pixels**: representação bidimensional discreta do objeto. Cada número indica a intensidade luminosa no “ponto”.

2.2 Representação do Objeto

Na literatura, encontramos várias formas de representação interna dos objetos. A fase de formação desta representação é muitas vezes denominada pré-processamento pois pode ser executada *off-line*, ou seja, antes do processamento dos dados para o reconhecimento propriamente dito.

Uma classificação para diversos pré-processamentos é apresentada por Edelman [13], a partir de alguns aspectos que Marr [36] denominaria como “teoria computacional”, examinando quatro abordagens de representação associadas a processos de reconhecimento.

Descrição estrutural: um objeto é representado como uma coleção de partes (escolhidas dentre um pequeno alfabeto comum a todos os objetos) juntamente com suas relações espaciais [30, 63, 64].

Aspectos Geométricos: um objeto é representado por coordenadas relativas de um pequeno conjunto de suas características [6, 61].

Espaço multidimensional de características: um objeto é representado por um vetor de características que podem ser geométricas, fotométricas, etc., como, por exemplo, os *eigenvectors* [38, 39, 59].

Aproximação em espaços de características: um objeto é representado como

uma superfície de baixa dimensão definida por pontos pré-definidos num espaço de características. [51].

Edelman também afirma que em ciências cognitivas, tradicionalmente, debates envolvendo teorias de representação de objeto são centradas em problemas computacionais derivados do efeito da variação do ponto de vista – i.e. posição do observador – na aparência dos objetos. Da mesma forma, o surgimento de métodos formais poderosos para superar este efeito e seus ótimos resultados empíricos parecem dirigir a atenção das discussões para outros tópicos.

2.3 Detecção de Contorno

Brunelli e Poggio [6] comentam que existem várias técnicas aplicadas para reconhecimento tão variadas como as redes neurais, a comparação de padrões elásticos, a Análise de Componentes Principais, os momentos algébricos e as linhas de iso-densidade. No entanto, as performances destes métodos não foram formalmente comparadas com testes precisos e dados equivalentes. Neste trabalho, onde todos os objetos são faces, eles fazem uma comparação entre dois métodos que têm diferentes formas de armazenamento de informação: aspectos geométricos e comparações de padrão. Este último se resume em, a partir de uma imagem armazenada em uma matriz bidimensional de intensidade de luminância, comparar, usando alguma métrica como a distância euclidiana, a imagem de entrada com as imagens padrão armazenadas. Na verdade, não foram feitas apenas comparações entre imagens completas contendo todo o objeto, mas também de partes relevantes (no caso de faces: olhos, nariz, boca, etc.) foram comparadas separadamente entre si.

Independente do método aplicado, uma normalização dos dados é necessária antes do reconhecimento. A normalização tenta alinhar as regiões relevantes em pontos comuns a todos os padrões, ajustando, também, as imagens de entrada (Brunelli e Poggio criam invariância à escala e à rotação fixando uma distância interocular

e uma direção do eixo entre os olhos, localizados por comparações com padrões de olhos). Dentre os métodos discutidos, a comparação de padrões obteve melhores resultados. Entretanto, resultados ainda melhores foram obtidos utilizando a variação da intensidade luminosa ao longo da imagem, ou seja, a magnitude do gradiente. Segundo Nalwa [41], a variação abrupta de magnitudes pode ser interpretada como bordas ou contornos, regiões onde a intensidade luminosa da imagem muda brusca-mente. Ao trabalharmos apenas com os contornos do objeto, estamos buscando uma invariância à iluminação, reduzindo sua influência na representação bidimensional de objetos tridimensionais.

Temos diversos métodos disponíveis para detecção de contorno. Todos os que serão citados neste trabalho se utilizam de uma operação denominada *convolução* entre uma região da imagem, função $I(x, y)$, e um padrão de mesmas dimensões denominado *filtro*, $f(x, y)$. A convolução discreta entre duas funções bidimensionais pode ser formalizada como [56]:

$$[f \otimes I](x, y) = \sum \sum f(u, v)I(x - u, y - v), \quad (2.3)$$

gerando uma nova imagem, onde cada ponto armazena o resultado do somatório dos pontos da imagem ponderados pelo filtro aplicado. Deve ser lembrado, no entanto, que, além da detecção da variação da magnitude, também podem ser criados filtros para operar outras transformações na imagem original (como veremos na última seção deste capítulo).

As Figuras 2.5 e 2.4 mostram filtros bastante simples para extrair o gradiente de um ponto, apresentados por Prewitt e Roberts, respectivamente [41]. Um filtro como o da Figura 2.6, além de detectar a variação da intensidade, é também independente à direção desta variação, ou seja, ele é isotrópico. Outros filtros mais complexos, mostrados na próxima seção, exploram um modelo de comportamento de células do sistema visual dos primatas. O princípio da filtragem, entretanto, é o mesmo.

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

Figura 2.4: **Filtro não isotrópico simples** que detecta a variação de iluminação na direção horizontal. Note que o somatório dos valores positivos é igual ao negativo, garantindo que, ao ser aplicado sobre regiões uniformes, a resposta seja nula.

0	1
-1	0

Figura 2.5: **Filtro não isotrópico** que detecta a variação de iluminação na direção diagonal. Note que o somatório dos valores positivos é igual ao negativo, garantindo que, ao ser aplicado sobre regiões uniformes, a resposta seja nula.

1/4	1/2	1/4
1/2	-3	1/2
1/4	1/2	1/4

Figura 2.6: **Filtro isotrópico** que detecta a variação de iluminação, independente da direção. Note que o somatório dos valores positivos é igual ao negativo, garantindo que, ao ser aplicado sobre regiões uniformes, a resposta seja nula.

2.4 Representação Modelando Células do Sistema Visual

Conforme vários exemplos apresentados na literatura [6, 15, 17, 34, 35, 36, 38, 47, 48, 65], o processamento de imagens com intuito de reconhecimento de objetos, faces, etc. se torna mais eficiente quando, ao invés de valores absolutos da luminância, utilizamos informações sobre os contornos do objeto, definidos por contraste. Esta representação torna o sistema mais robusto a variações de direção e intensidade da

iluminação. Ullman [60] afirma que, embora também sejam utilizadas outras informações, como cor e textura (ou mesmo conhecimento anterior) para construir uma representação do objeto, o reconhecimento pela forma é provavelmente o aspecto mais comum e importante no reconhecimento visual.

Desta forma, criamos um sistema de extração de bordas inspirado em modelos biológicos (ver Apêndice 6 e [23, 49]), onde a representação adotada está baseada nas respostas das células dos primeiros níveis do sistema visual dos primatas: os filtros concêntricos da retina e as células simples e complexas do córtex primário (Figura 2.7). Embora existam outros processos eficientes para detecção de borda, escolhemos manter o embasamento biológico como forma de tratamento das imagens.

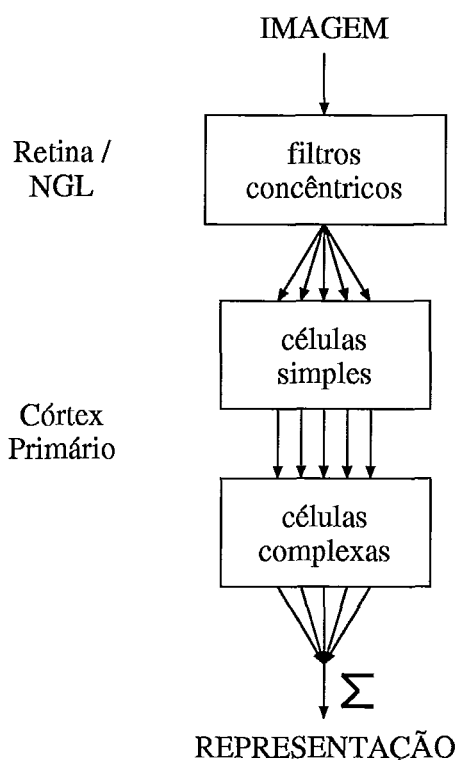


Figura 2.7: **Seqüência de filtragens** para gerar a representação da imagem, baseado no modelo hierárquico proposto por Hubel e Wiesel (ver referência no texto). A imagem é captada pelas células fotoreceptoras da retina e projetada para o Núcleo Geniculado Lateral (NGL) do tálamo. Esta informação será, depois, combinada em células simples e complexas do córtex.

Na retina, temos um conjunto de células fotoreceptoras, neurônios adaptados para transformar estímulo luminoso em energia [32, 66]. A saída de informações

da retina se dá por meio das células ganglionares (cujos axônios formam o nervo óptico) se projetando para o núcleo geniculado lateral do tálamo. As entradas para cada célula ganglionar sempre se originam a partir dos mesmos fotorreceptores numa área circunscrita na retina, o *campo receptor* daquela célula. No modelo hierárquico proposto por Hubel e Wiesel [28, 29, 37, 66], células simples combinam informações provenientes das células ganglionares da retina e representam o primeiro estágio de processamento cortical. Células complexas, por sua vez, recebem como entrada a soma de várias células simples com seletividade para a mesma orientação.

Em nosso modelo, inicialmente utilizamos filtros com campos receptores concêntricos (*center-surround*) não orientados cujo comportamento se assemelha às células ganglionares da retina e às células do núcleo geniculado lateral do tálamo (NGL). Na Figura 2.8–A, definimos os campos receptores como duas regiões concêntricas não orientadas ao redor de um ponto, tal que, regiões próximas ao ponto (*center*) irão excitar a célula, enquanto, regiões mais afastadas (*surround*), irão inibi-la. Graças a esta *inibição lateral*, estas células são sensíveis ao contraste produzido pela variação de luminância entre o centro e a periferia, detectando descontinuidades luminosas na imagem.

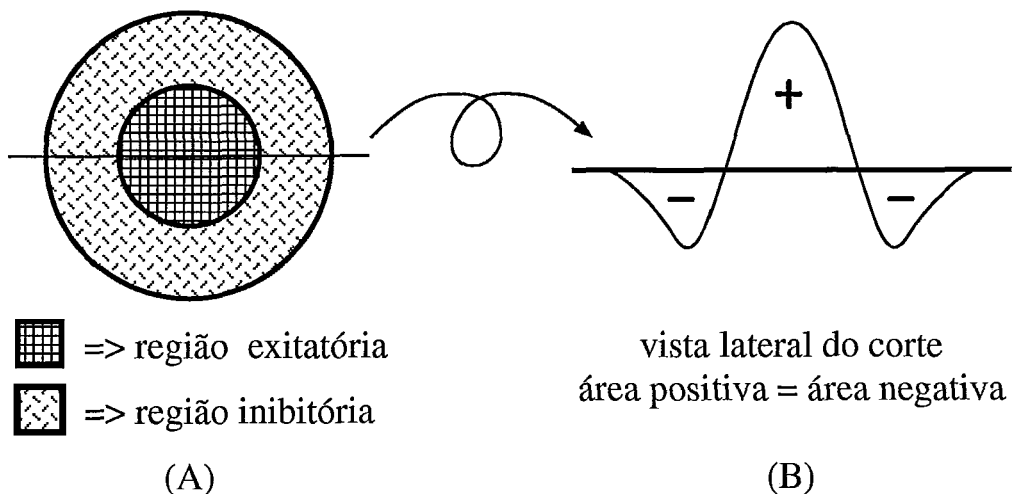


Figura 2.8: **Filtro Concêntrico:** (A) Definição das regiões para o filtro *center-surround*. (B) Filtro Gaussiano de Diferença – isotrópico.

Pela vista lateral do corte apresentado na Figura 2.8–B, podemos verificar a

distribuição da função, que define as regiões excitatória e inibitória do campo receptor na forma de “chapéu–mexicano”, produzida pela diferença de duas gaussianas (Figura 2.9). A área sob a região excitatória deve ser equivalente às áreas das regiões inibitórias. Desta forma, quando o campo receptor da célula cobrir uma região com estímulo totalmente uniforme, a resposta será zero. Além disso, para criar uma invariância em relação à quantidade e à direção da iluminação, as respostas devem ser normalizadas. Estas respostas podem ser resumidas por:

$$r = \mathcal{N}(I \otimes G), \quad (2.4)$$

onde r é a resposta na retina/NGL, I é a entrada da distribuição luminosa, G é um filtro Gaussiano de Diferença (chapéu–mexicano – Figura 2.9), e \otimes implementa a operação de convolução. O operador \mathcal{N} normaliza as respostas para que sejam independentes dos níveis de iluminação.

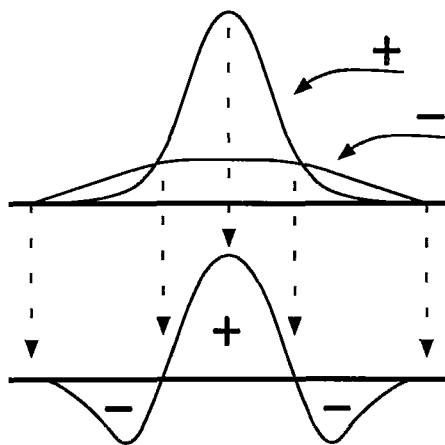


Figura 2.9: **Filtro Gaussiano de Diferença:** duas funções gaussianas de diferentes amplitudes e desvio padrão são combinadas numa nova função.

Em seguida, as respostas não orientadas são processadas por filtros alongados sensíveis à *orientação* do contraste, cujos campos receptores estão representados na Figura 2.10. Estes filtros são sensíveis também à *direção* do contraste, já que existem células simples que respondem bem a transição claro–escuro (Figura 2.10–A), enquanto outras respondem melhor a transição escuro–claro (Figura 2.10–B).

Estas células simples não devem ser vistas apenas como detectores de borda. Mesmo sendo fortemente disparadas por transições luminosas abruptas em uma determinada orientação, também são capazes de responder a variações suaves da imagem. Assim, as respostas não são simplesmente o contorno dos objetos, mas uma modulação do contraste da imagens que permite destacar bordas, texturas e sombras.

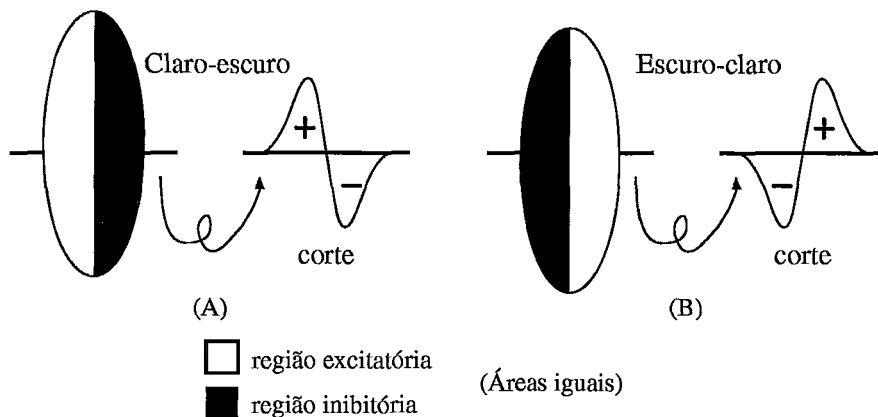


Figura 2.10: **Filtros alongados em células simples** de transição (A) claro–escuro e (B) escuro–claro, sensíveis a direção de contraste. Note que as áreas excitatórias e inibitórias são iguais, garantindo que, ao serem aplicados sobre regiões uniformes, a resposta seja nula. Não isotrópico.

Para detectar todas as orientações de contraste é necessário criar diversos filtros que acompanhem estas variações. Estes filtros são flexíveis e respondem, mesmo que fracamente, a orientações que não estejam no ângulo exato definido por ele. Desta forma, para cobrir razoavelmente todas as orientações, testes executados por Exel [16] mostraram que se torna suficiente estabelecer quatro orientações para cada tipo de célula simples. A Figura 2.11 ilustra orientações de filtros alongados com transição claro–escuro¹.

Novamente, estimulação constante deve gerar resposta zero e, para tanto, a região excitatória deve gerar a mesma resposta que a região inibitória. Os campos receptores (área de atuação do filtro) podem ser formalizados diretamente por filtros que representem a diferença entre duas gaussianas deslocadas (Figura 2.12). As respostas das células simples s , para cada orientação k , são computadas pela con-

¹Para obter os outros filtros, basta inverter as regiões excitatória e inibitória.

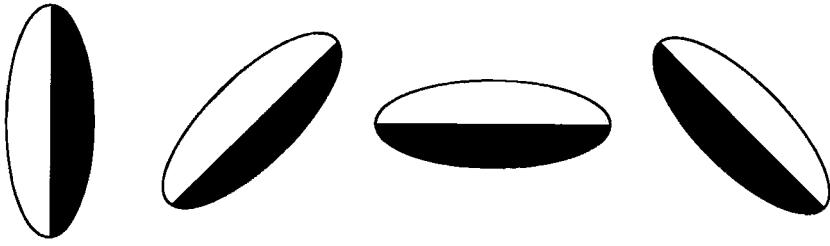


Figura 2.11: **Orientações dos filtros alongados (claro–escuro).** Como os filtros são flexíveis, para cobrir razoavelmente todas as orientações, foi suficiente estabelecer quatro orientações para cada tipo de célula simples. Para calcular estas novas orientações, basta aplicar a matriz de rotação adequada. Para obter os outros filtros, basta inverter as regiões excitatória e inibitória.

volução entre as respostas dos filtros concêntricos r e os respectivos filtros S para transição claro–escuro (ce) e escuro–claro (ec):

$$s_k^{ce} = r \otimes S_k^{ce} \quad \text{e} \quad s_k^{ec} = r \otimes S_k^{ec}. \quad (2.5)$$

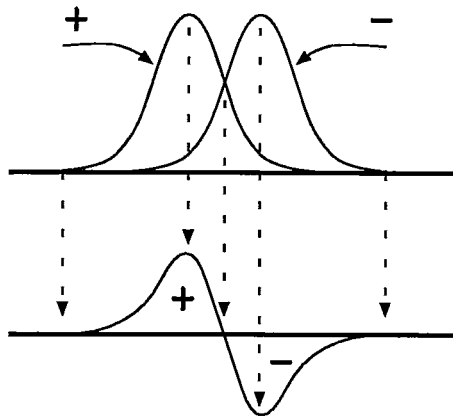


Figura 2.12: **Funções gaussianas, deslocadas de um *off-set*,** são combinadas numa nova função utilizada para a definição dos filtros de células simples.

Acompanhando o modelo hierárquico, combinamos as respostas de células simples de orientação igual e direção complementar de transição (claro–escuro e escuro–claro), modelando as células complexas (Figura 2.13). Ao contrário das células simples, as complexas não apresentam regiões discretas de excitação e inibição pois nestas elas se fundem. Esses filtros podem ser definidos como na equação:

$$c_k = s_k^{ce} + s_k^{ec} \quad (2.6)$$

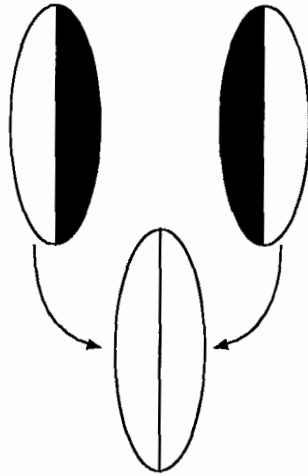


Figura 2.13: **Filtro alongado em uma célula complexa.** A combinação das respostas das células simples de mesma orientação é insensível a direção da transição do contraste.

Um mapa sensível aos contornos (bordas, texturas e sombras) da imagem em todas as orientações é gerado pela combinação (Figura 2.14) das respostas de todas as células complexas aplicadas sobre um mesmo ponto [17, 47, 48].

$$C = \sum_k c_k \quad (2.7)$$

O resultado final destas filtragens pode ser visto no exemplo da Figura 2.15.

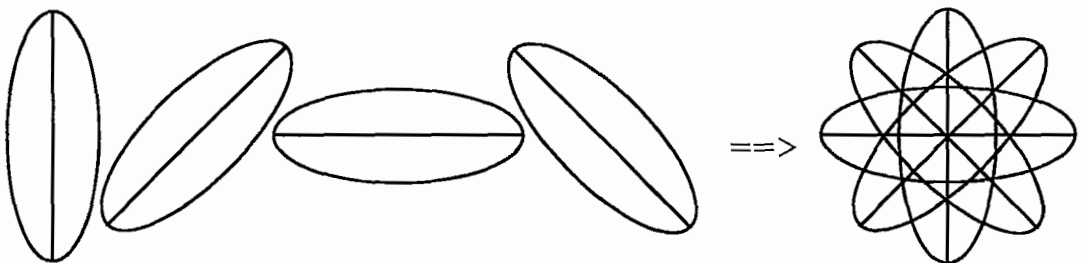


Figura 2.14: Mapa de respostas das células complexas com o somatório de todas as orientações.

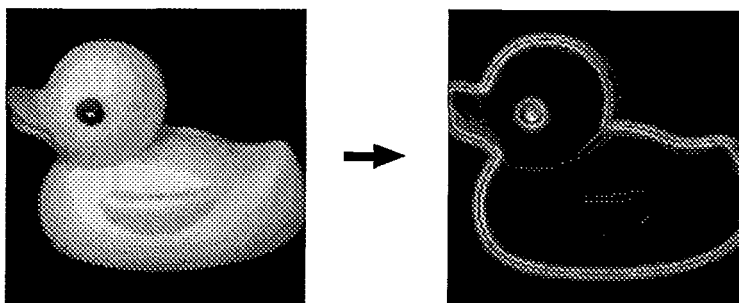


Figura 2.15: Exemplo do resultado final de uma filtragem mostra a representação da imagem através das respostas de células complexas.

2.5 Pirâmides

Voltando ao trabalho de Brunelli e Poggio [6], uma questão apresentada é a dependência no reconhecimento da resolução do objeto na imagem. Para analisar este assunto, é sugerido o uso de pirâmides. A definição de Burt e Adelson [1, 8, 9] para pirâmides é a seguinte: “Pirâmide de imagens é uma estrutura de dados desenvolvida para proporcionar convoluções de escala (espacial)² eficientes através de representações reduzidas da imagem. Consiste de uma seqüência de cópias da imagem original nas quais, tanto a densidade, quanto a resolução, são diminuídas em passos regulares”. A Figura 2.16 apresenta dois exemplos de pirâmide, cada uma com um diferente método de construção.

A estrutura de dados de pirâmide é interessante pois descreve a mesma imagem em diversos níveis de detalhe, ou resolução. A resolução mais alta (a resolução original da imagem) permite a representação de detalhes. As resoluções mais baixas, por sua vez, são menos detalhadas mas, ao mesmo tempo, requerem menos espaço para serem armazenadas. É importante ressaltar, no entanto, que resoluções mais baixas representam informações de baixa frequência (ou escala) espacial. Nestas escalas podem ser representadas informações sobre regiões maiores da imagem. Por exemplo, enquanto uma frequência alta – associada a um nível de alta resolução da pirâmide – pode especificar os contornos dos olhos numa imagem de rosto, uma

²Comentário acrescentado.

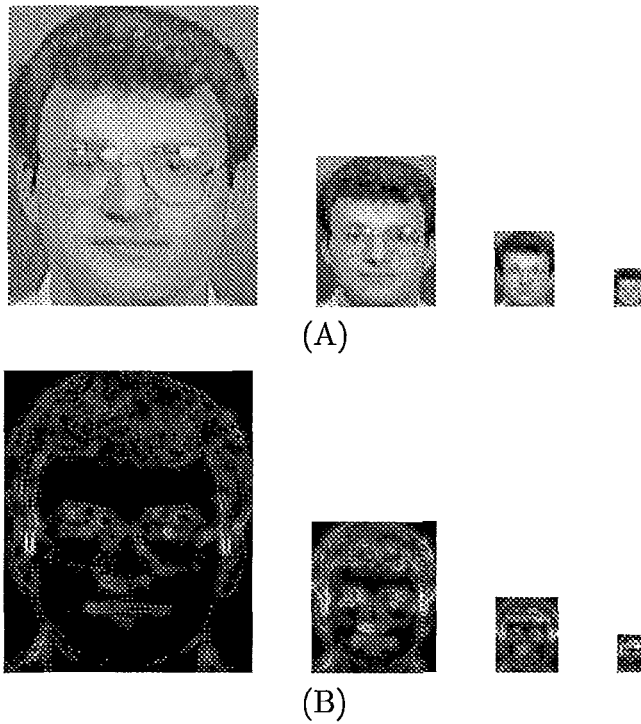


Figura 2.16: **Pirâmides:** (A) Pirâmide de uma imagem utilizando apenas redução de dimensão. (B) Pirâmide da mesma imagem com os dois passos de construção: filtragem e redução de dimensão.

frequência mais baixa pode representar os olhos como uma estrutura “única” (ver as duas imagens mais à esquerda na Figura 2.17, A e B). Deve ser mencionado que este comportamento possui um paralelo direto com a organização do sistema visual, onde células sensíveis a diferentes frequências espaciais são encontradas [2].

A utilização de estrutura de dados de pirâmides permite a investigação sistemática da escala espacial mais adequada para o reconhecimento de objetos particulares. De fato, estudos psicofísicos indicam que frequências espaciais baixas são importantes no reconhecimento de faces. Deve ser notado que, em níveis de menor resolução (associados à frequência espacial baixa), pequenas variações da imagem original levarão a pequenas mudanças na representação da mesma, devido à baixa resolução disponível e ao *blurring* (“embaçamento”) associado (Figura 2.17; ver também Figura 2.21 e discussão associada).

As pirâmides são construídas em 2 passos: uma filtragem e uma redução de dimensão da imagem, denominada *downsampling* (Figura 2.18). Conforme as

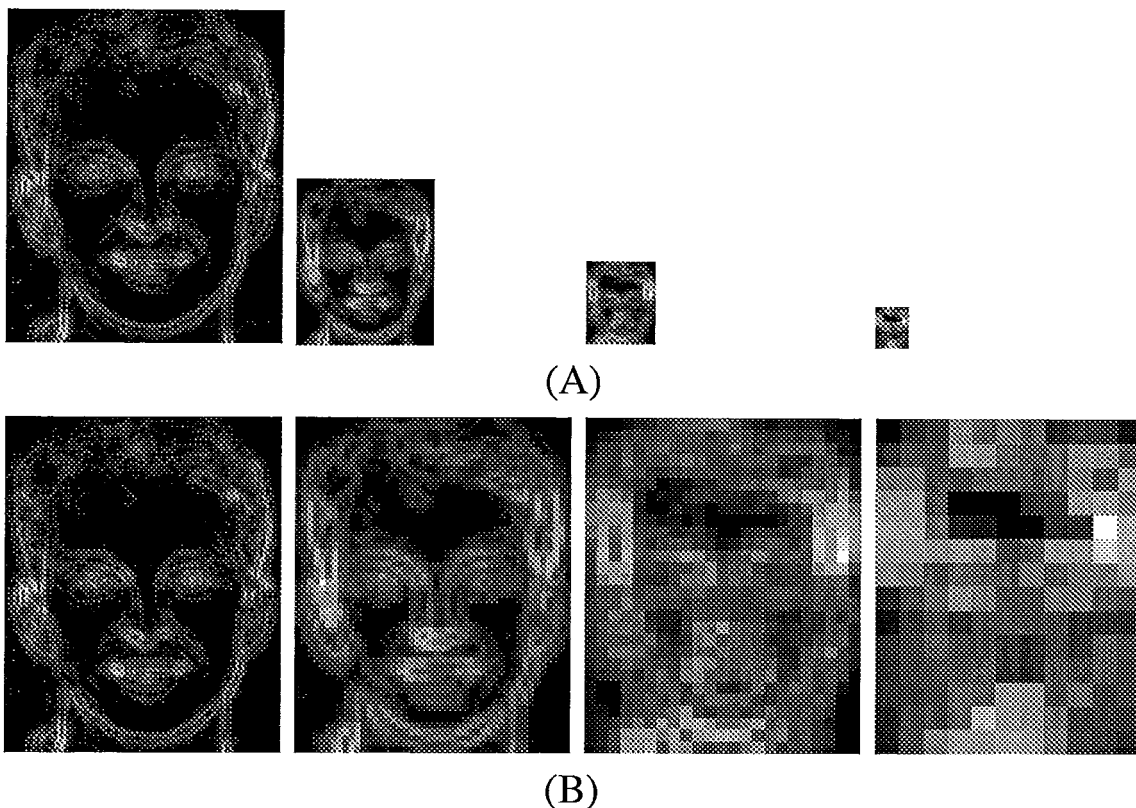


Figura 2.17: **Variação da resolução a cada nível:** (A) Pirâmide da imagem de um objeto nas dimensões corretas. (B) Pirâmide da mesma imagem com os níveis mais baixos ampliados para melhor visualização.

funções escolhidas para a filtragem, criam-se pirâmides Gaussianas, Laplacianas, etc. [1, 8, 9]. Usando filtros de mesmo tamanho para todos os níveis da pirâmide, os *downsamplings* fazem com que este filtro cubra uma região cada vez maior do objeto, de tal forma que as imagens resultantes sejam cada vez menos detalhadas, como pode ser visto na Figura 2.17.

Para as pirâmides criadas neste trabalho, usamos as filtragens baseadas no sistema visual, conforme descrito anteriormente, definindo 4 níveis cujas dimensões são sempre a metade do anterior. Cada nível segue a mesma regra de formação sendo feita a filtragem seguida pelo ajuste de dimensão. Para o primeiro nível (nível 0) estabelecemos um tamanho para o filtro (Figura 2.19) que será aplicado à imagem original. Neste nível, a imagem final mantém as mesmas dimensões da imagem original (em outras pirâmides, como as Gaussianas, a própria imagem original é

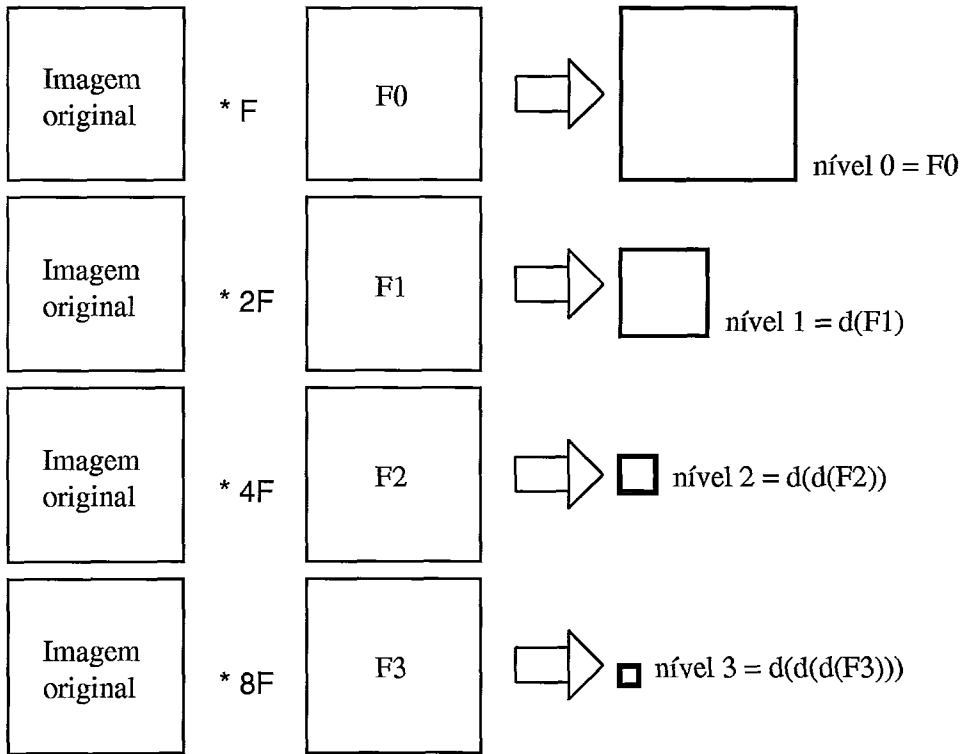


Figura 2.18: **Construção da Pirâmide.** F é a dimensão do filtro conforme a Figura 2.19, a seguir. $d(imagem)$ é uma função que retorna o *downsampling* de uma imagem, uma redução nas dimensões desta.

armazenada como nível 0). Para o segundo nível, dobramos as dimensões do filtro (Figura 2.19) utilizado anteriormente. Depois de filtrada, a imagem tem suas dimensões reduzidas à metade aplicando o *downsampling* (ver abaixo). Seguindo o mesmo esquema, dobramos novamente o filtro anterior (Figura 2.19) para a filtragem do nível 2. Depois de filtrada, a imagem tem suas dimensões reduzidas duas vezes consecutivas. Finalmente, dobramos o filtro (Figura 2.19) uma terceira e última vez e aplicamos o *downsampling* três vezes.

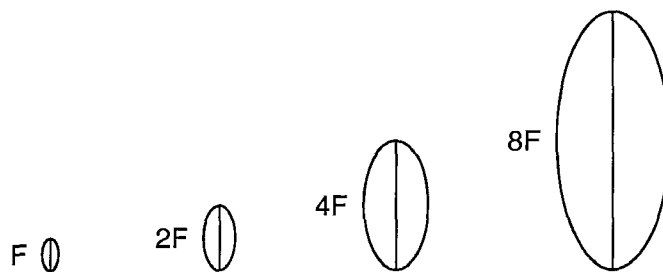


Figura 2.19: **Proporção dos filtros.** Este esquema mostra que as dimensões dos filtros são dobradas para cada novo nível

Para a redução de dimensão ou *downsampling*, utilizamos um método bastante simples onde a imagem é dividida em pequenas matrizes 2x2. Em seguida, cada matriz é mapeada em um pixel (um ponto) da imagem reduzida através da média aritmética da intensidade de seus 4 pixels, como mostrado na Figura 2.20. Como pode ser notado, o número de *downsamplings* aplicados foi igual ao número de vezes que o filtro foi dobrado para cada nível. Conforme foi construída, cada pirâmide armazena representações da mesma imagem em quatro níveis de detalhamento (Figura 2.17), como no trabalho de Brunelli e Poggio [6].

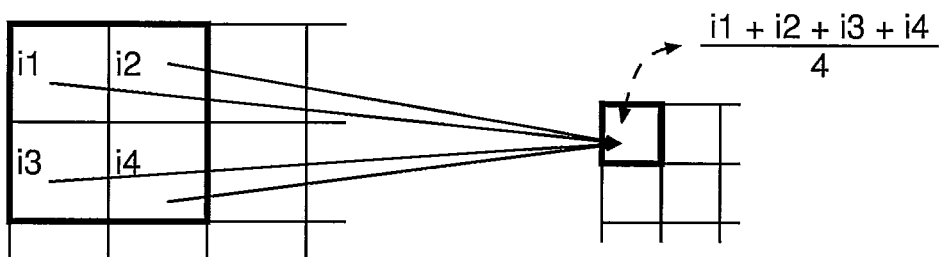


Figura 2.20: **Redução de dimensão** ou *downsampling*. Calcula a média aritmética das intensidades de uma máscara 2x2 de pixels gerando um único pixel médio.

Para guardar mais detalhes, mais informações, precisamos de mais pixels, o que gera imagens maiores e aumenta o custo computacional de processamento. Analogamente, para armazenar menos detalhes geramos imagens menores, diminuindo o custo de processamento. Buscamos otimizar a relação detalhamento *versus* custo, investigando, dentre os níveis da pirâmide, aquele no qual a imagem é o mais simples possível (menos detalhes), mantendo-se suficientemente representativa para o reconhecimento.

Tomemos como exemplo as Figuras 2.21 e 2.22. Nelas apresentamos uma pirâmide para cada imagem. Para facilitar a comparação, os níveis mais baixos foram ampliados num processo de *upsampling*. Seguindo a ordem da esquerda para a direita, temos: a imagem original, e os níveis 0, 1, 2 e 3 da pirâmide. Nos primeiros níveis (de 0 a 2) é possível perceber vários detalhes dos objetos, tais como a rosca e o parafuso (bocal) e os olhos, focinho e manchas (gato). Nos níveis 0 e 1 da

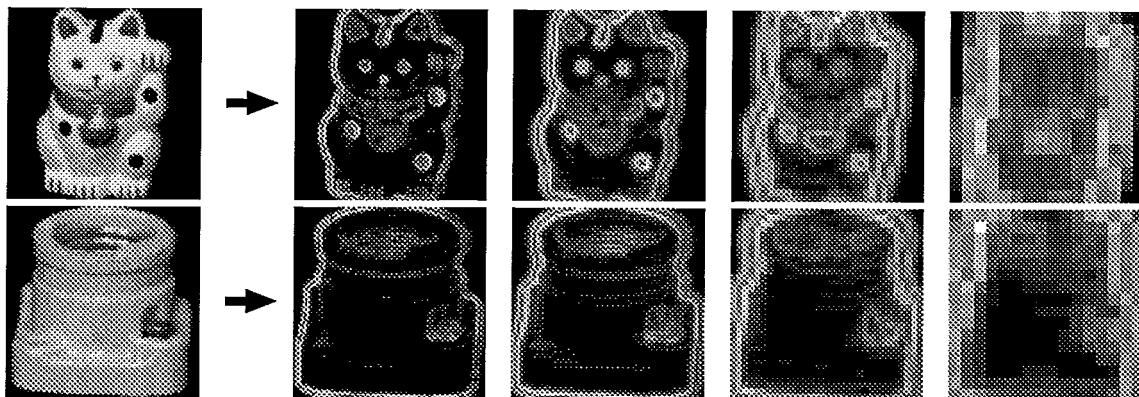


Figura 2.21: **Pirâmides do gato e do bocal.** Os níveis menores (à direita) foram ampliados para melhor comparação. Mesmo nos menores níveis a distinção ponto a ponto entre os objetos parece clara.

Figura 2.21, é bastante claro verificar que uma comparação pixel a pixel pode levar à distinção entre o gato e o bocal. Em nosso trabalho, tentamos verificar se ainda é possível distinguir objetos nos níveis mais baixos da pirâmide, implementando o mesmo processo de comparação.

Considere, agora, a situação complementar da Figura 2.22. Duas imagens do *mesmo* objeto, produzem representações distintas nos níveis 0 e 1 da pirâmide. Ou seja, uma rotação de um objeto pode gerar imagens em 2-D bem diferentes. Entretanto, nas imagens dos níveis 2 e 3, os filtros cobrem uma região cada vez maior na imagem original, assim cada pixel representa uma porção maior do objeto. Desta forma, os deslocamentos causados pela rotação podem ser amenizados, e um reconhecimento mais robusto pode ser obtido com estes níveis da pirâmide. Deve ser notado, também, que resoluções muito baixas em uma pirâmide não serão informativas o suficiente para permitir um reconhecimento eficaz. Imagens de diferentes objetos podem tornar-se muito parecidas, confundindo o processo de reconhecimento.

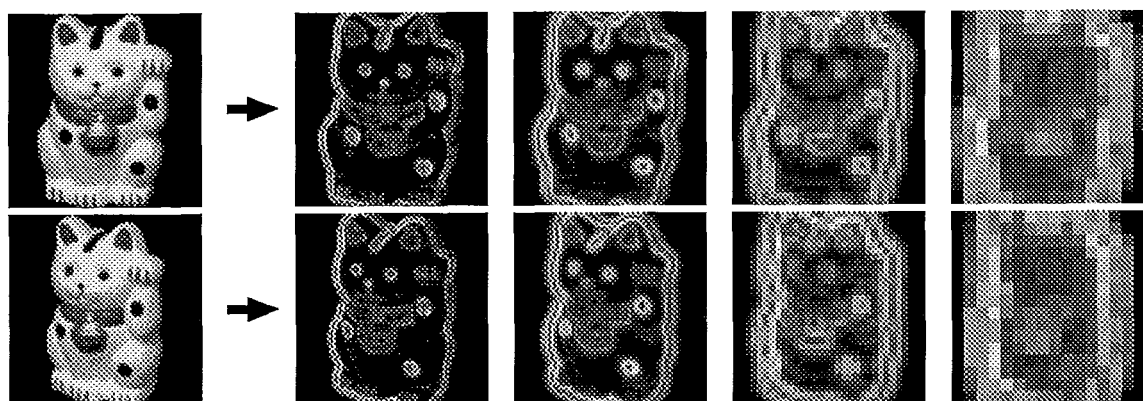


Figura 2.22: Pirâmides do gato com rotações diferentes. Os níveis menores (à direita) foram ampliados para melhor comparação. Nos menores níveis a semelhança ponto a ponto se torna cada vez mais clara.

Capítulo 3

Reconhecimento: Classificação e Identificação

Reconhecimento de objetos é um dos mais importantes e menos estudados aspectos da percepção visual. Para sistemas visuais biológicos, reconhecimento e classificação de objetos é uma atividade espontânea e natural. Em contraste, o reconhecimento de objetos ainda está além das capacidades dos sistemas artificiais, ou de qualquer modelo de reconhecimento proposto até agora. A comparação entre o cérebro e o computador quanto à capacidade de realizar reconhecimento e classificação visual torna este problema fascinante. O cérebro generaliza espontaneamente a partir de exemplos visuais, sem a necessidade de regras explícitas ou instruções trabalhosas [60]. Mais do que nomear, o reconhecimento é o produto final da habilidade de recuperar informações associadas a um objeto ou classe de objetos, que não seja a imagem propriamente dita. O nome é, tão somente, mais uma informação.

Objetos podem ser reconhecidos em diferentes graus de especificação. Algumas vezes estão associados a uma classe mais geral, como “casa”, “cachorro”, “carro”, “faces” – classes que contêm uma grande variedade de objetos, de diversos formatos. Objetos também podem ser identificados como únicos, como a minha casa, ou a face de um amigo. Classes de objetos podem ainda ter diferentes níveis de generalização como “poodle”, “cachorro” ou “quadrúpede”. Vários esquemas de reconhecimento se baseiam na identificação de objetos, distinguindo bem objetos individuais, tendo dificuldade, porém, para classificar novas imagens dos mesmos objetos.

A tarefa de identificar objetos individuais pode, a princípio, parecer mais custosa do que outros níveis mais gerais, pois necessita de informações mais detalhadas para melhores resultados. Ullman [60] afirma que, para sistemas artificiais, é mais fácil reconhecer formas conhecidas, mesmo complexas, sob diferentes condições de iluminação do que capturar as características em comum em uma classe de objetos, abrindo uma discussão sobre a quantidade de informação necessária para classificar um objeto. Para que não seja preciso usar toda a informação sobre o objeto quando queremos identificá-lo, podemos usar a classificação como um estágio intermediário, tornando-se necessário muito menos informação para distinguir um carro de uma casa do que para distinguir o meu carro dentre várias imagens de carros e de casas.

3.1 Estrutura de Memória: Modelos

Neste trabalho, o reconhecimento de determinada imagem de um objeto consistiu, num primeiro passo, do estabelecimento de uma *memória*, ou seja, uma representação interna de diversas imagens dos objetos¹. Em seguida, a imagem foi comparada a cada instância desta memória para então ser reconhecida pela imagem mais semelhante. Definimos como *modelo* cada instância desta memória e cada combinação de condições de apresentação de um objeto como *imagem*. Numa situação extrema, cada modelo seria uma cópia exata de cada imagem do objeto, tal que a memória representaria todas as poses deste. Assim, quando uma imagem qualquer fosse apresentada, seria sempre reconhecida por ser idêntica a um modelo da memória (Figura 3.1). Mas esta situação é notadamente irreal, pois não podemos ter *todas* as poses possíveis de cada objeto armazenadas.

Objetos podem ser observados em diversas condições: em diferentes ângulos, distâncias, ambientes e condições de iluminação. Além disto, objetos deformáveis podem apresentar pequenas distorções, como as mudanças de expressões em faces.

¹Neste trabalho, todas as imagens manipuladas pelo sistema foram previamente filtradas e estão armazenadas em pirâmides, conforme a descrição no Capítulo 2.

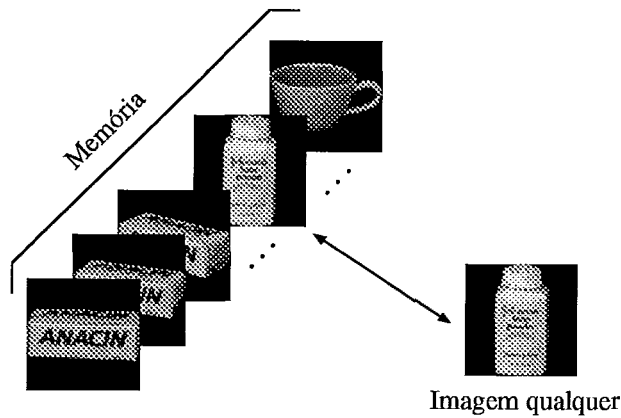


Figura 3.1: Modelos para *todas* as imagens de cada objeto. Se todas as poses forem armazenadas na memória, o sistema reconhece sempre pois encontra um modelo 100% semelhante. Porém, isto é inviável.

Seria mais interessante criar modelos mais ricos que conseguissem resgatar as informações mais importantes para a distinção de um objeto. Neste trabalho, buscamos isto através de modelos de memória estabelecidos pela combinação de algumas imagens (Figura 3.2) que representem as variações dessas condições. Ullman [60] estabelece que uma combinação de um pequeno número destas poses pode ser suficiente para o reconhecimento de objetos tridimensionais. Deste modo, ao comparar a imagem de entrada com toda a memória, o reconhecimento ocorre quando a melhor resposta do sistema, o modelo mais semelhante a esta imagem, pertencer à classe que representa o objeto desta entrada (Figura 3.3). Em linhas gerais, o reconhecimento é abordado como um processo de classificação.

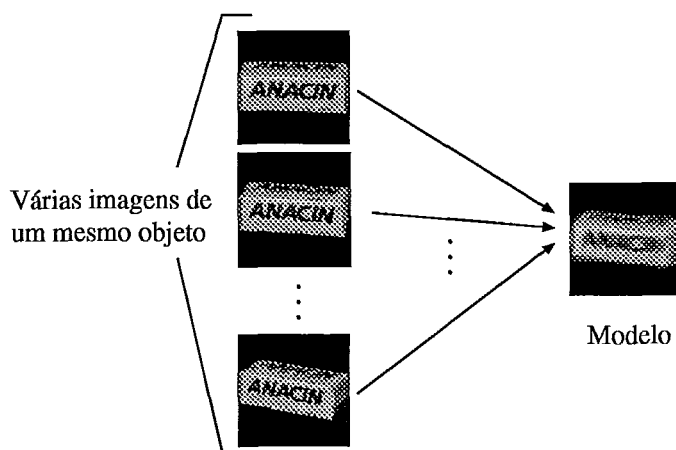


Figura 3.2: Um modelo combinando várias imagens. Podemos armazenar uma combinação de várias imagens de um mesmo objeto.

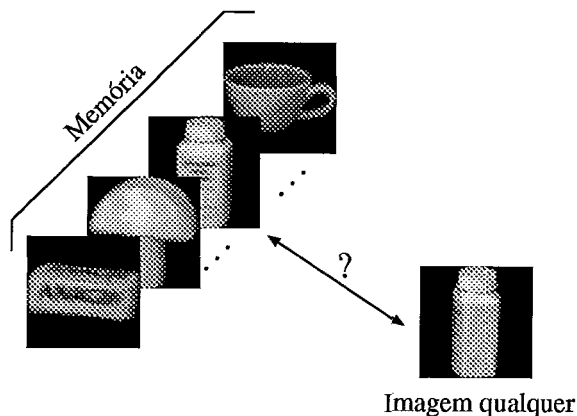


Figura 3.3: **Modelos com combinação de imagens de cada objeto.** O sistema escolhe o modelo mais parecido com a imagem. Se os modelos forem suficientemente representativos, a resposta será o objeto contido na imagem.

3.2 Categorização de Informação

Temos dois conceitos a serem discutidos: o estabelecimento dos modelos que compoem a classe e a classificação de uma entrada em determinado modelo. Dizemos que uma dada informação pertence a uma determinada categoria quando ela é similar a outras informações da mesma categoria e, ao mesmo tempo, distante das informações das outras categorias. Entretanto, uma categoria não precisa armazenar todas as informações que representa, pode armazenar um *padrão* que represente uma combinação de todas elas. Para que estes padrões, ou modelos de memória, representem várias imagens, devemos extrair o que elas têm em comum e, uma forma de conseguir isto é extraindo os padrões criados por uma categorização das imagens. Por outro lado, a mesma medida usada para calcular a distância entre a imagem e as categorias pode definir o mecanismo de classificação, de tal forma que, o modelo mais mais parecido irá estabelecer a classe que melhor representa a entrada.

Existem várias formas de categorização de informação. Algumas delas são encontradas nas redes neurais não-supervisionadas, como a família de redes neurais ART (Adaptative Resonance Theory) [10, 11, 22] e a rede auto-organizável de Kohonen [33], onde conjuntos de exemplos são, automaticamente, mapeados em classes.

Cada classe, por sua vez, pode ser descrita por um conjunto de características das informações que representa, os protótipos.

Na Figura 3.4, apresentamos um esquema simplificado da rede neural ART, que contém uma camada de entrada e uma camada de saída ou de categorias. A imagem de entrada determina os valores da camada de entrada, que são enviados para a camada de categorias por um conjunto de pesos. Neste pesos estão armazenados os padrões de cada modelo. Na camada de saída, cada nó apresenta uma ativação pela medida de similaridade do seu padrão com a entrada. Ao final, o nó mais ativo representa a categoria que melhor classifica a entrada (os demais são desativados). Em princípio, diz-se que a rede neural ART Fuzzy, como as demais redes ART, codifica a entrada por uma medida de similaridade implementada através do seu conjunto de pesos e a regra de ativação da camada de saída. Fica claro afirmar que o protótipo de cada classe está armazenado no vetor de pesos associado. Estes vetores são a memória desta rede.

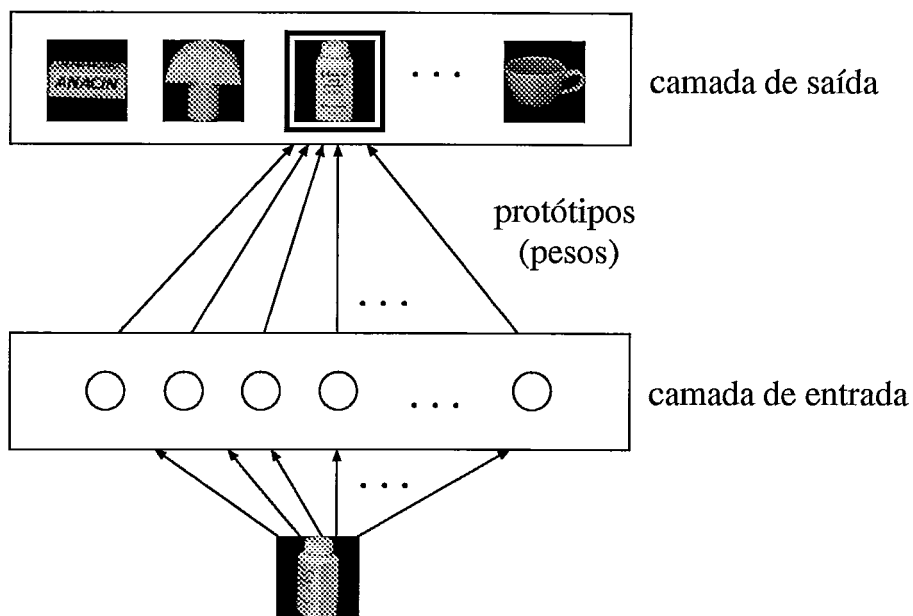


Figura 3.4: **Esquema da rede ART Fuzzy.** A informação de entrada percorre a rede em direção da camada de saída para ser calculada a semelhança com o modelo de cada categoria, armazenado no vetor de pesos.

Um vetor de pesos atua ponderando o cálculo da ativação das unidades de saída a

partir das características da entrada, dando mais ênfase às informações mais importante para distinguir cada classe das demais. No sistema proposto neste Capítulo, vamos estabelecer protótipos pela combinação simples das imagens de cada objeto, seguindo um esquema parecido com o das redes ART, conforme veremos na próxima seção.

Neste trabalho, propomos um sistema que se comporta como uma rede neural *feed forward*, ou seja, as informações trafegam em paralelo em um único sentido, i.e. da camada de entrada para a camada de saída. Num primeiro esquema, onde cada categoria de saída representa um único objeto, temos apenas duas camadas (entrada e saída). Os modelos que constituem a memória (um para cada categoria) estão armazenados como padrões vetores de peso (W_j) conectando a camada de entrada à categoria que ele representa, como mostra o esquema na Figura 3.5.

Cada nova entrada apresentada é comparada aos modelos armazenados. A ativação dos neurônios de saída é o valor de semelhança entre estes dados. Esta semelhança pode ser medida de várias formas, tais como, a distância euclidiana e o produto interno. Uma outra medida de semelhança implementada é encontrada nas redes neurais Fuzzy ART [10, 11], denominada função de ativação mostrada na Equação 3.1, onde I é a imagem de entrada, W_j é o vetor de pesos da categoria j , α é uma constante e \wedge , o operador AND Fuzzy.

$$\text{Ativação} = \frac{|I \wedge W_j|}{\alpha + |W_j|} \quad (3.1)$$

Para qualquer uma das medidas escolhidas, a resposta do sistema é dada pela categoria mais ativa, ou seja, qual modelo de qual objeto tem maior similaridade com determinada entrada e as demais saídas são desconsideradas.

Em um caso onde existem poses muito diferentes, i.e. o mesmo objeto de frente, de perfil e de costas, estabelecermos uma combinação única para representar cada objeto pode acabar degenerando a informação armazenada pelo modelo de memória.

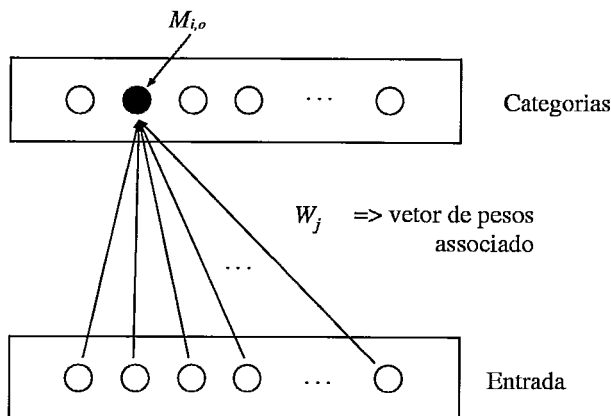


Figura 3.5: **Primeiro sistema de classificação.** A categoria escolhida é a mais ativa segundo alguma medida de similaridade. O vetor de pesos armazena o modelo que representa as categorias.

Num segundo esquema proposto, definimos vários modelos para cada objeto, ou seja, um conjunto de categorias serve como saída intermediária do sistema. Em uma outra camada, os neurônios associados ao mesmo objeto são mapeados para a mesma saída, como pode ser acompanhado na Figura 3.6. Neste caso, a resposta do sistema é dada pelo objeto associado à categoria mais próxima da imagem testada. Novamente, a resposta é considerada correta se o objeto escolhido for o mesmo que estiver representado na imagem testada.

3.3 Síntese de Modelos

Depois de estabelecidos os conjuntos de poses que darão origem aos modelos em cada simulação, precisamos definir como iremos extrair o que estas imagens têm em comum. Dado o conjunto de imagens I_1, I_2, \dots, I_n , criamos duas funções para geração de modelos neste trabalho, denominadas *Mínimo* e *Média*. A primeira, aproveita idéias apresentadas pela própria rede ART Fuzzy. De uma forma geral, podemos afirmar que uma categoria de uma rede neural, durante o processo de aprendizagem, armazena uma combinação de todas as entradas por ela classificadas em seu vetor de pesos. Quando os vetores de entrada representam imagens bidimensionais, o padrão armazenado no vetor de pesos também será a representação de uma imagem em 2D.

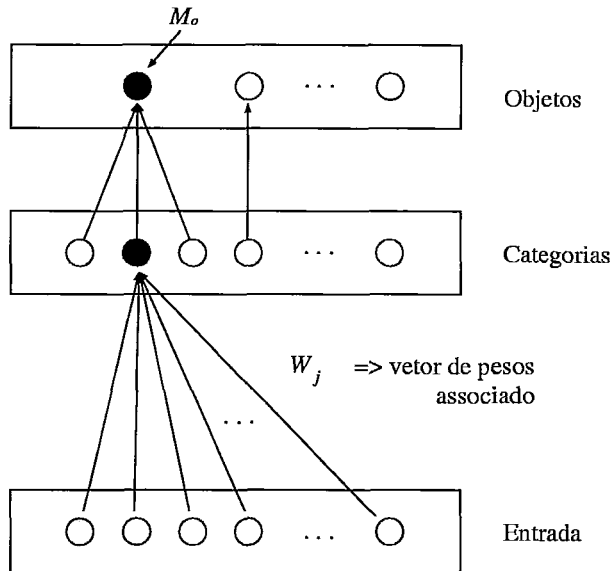


Figura 3.6: **Segundo sistema de classificação:** A categoria escolhida, a mais ativa segundo alguma medida de similaridade, define a resposta do sistema, ativando o objeto classificado. O vetor de pesos entre as camadas de entrada e de saída intermediária armazena o modelo que representa as categorias.

Na rede ART Fuzzy, o vetor de peso é dado pela combinação destas entradas através do operador AND fuzzy, de tal forma que a informação codificada seja o *mínimo valor* dentre todas as entradas. Criamos, então, uma regra para o estabelecimento de modelos segundo a Equação 3.2.

$$Mínimo = I_1 \wedge I_2 \wedge \dots \wedge I_n = \min(I_1, I_2, \dots, I_n). \quad (3.2)$$

É bastante razoável supor que exista uma representação média de um conjunto capaz de representar todos os elementos deste conjunto. Seguindo esta idéia, outra forma encontrada para combinar as informações de um grupo de imagens foi a *média da intensidade* dos pixels, fazendo com que o padrão armazenado por um modelo codifique a posição média entre todas as imagens do conjunto. Utilizando a média aritmética, a Equação 3.3 constrói o modelo para cada conjunto de imagens.

$$Média = media(I_1, I_2, \dots, I_n) = \frac{I_1 + I_2 + \dots + I_n}{n} \quad (3.3)$$

As Equações 3.2 e 3.3 estão simplificadas. Estas funções mapeiam entradas \mathcal{R}^2

em saídas \mathcal{R}^2 . Os resultados – *Mínimo* e *Média* – e as entradas – I_1, I_2, \dots, I_n – são imagens bidimensionais e as operações devem ser aplicadas ponto a ponto em cada coordenada (x, y) . As Figuras 3.7, 3.8 e 3.9 abaixo ilustram as duas estratégias estudadas. Deve ser lembrado, no entanto, que as entradas para estas combinações foram sempre imagens pré-processadas, ou seja, cada nível da pirâmide devidamente filtrado.

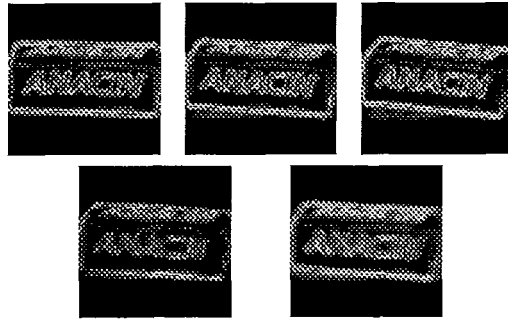


Figura 3.7: **Combinação de 3 imagens filtradas.** Na primeira linha, temos um intervalo de 3 imagens filtradas, e na segunda, os modelos mínimo e média, respectivamente.

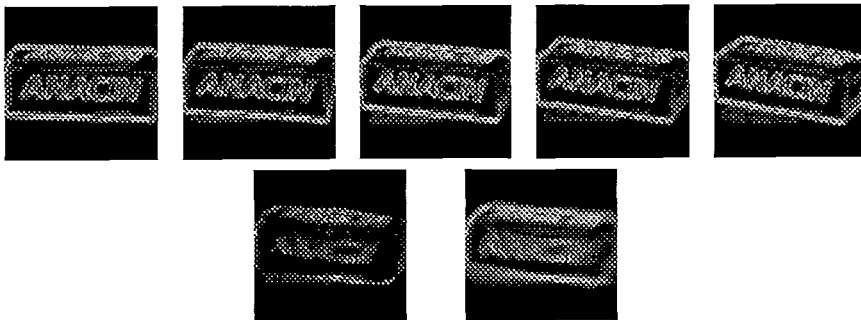


Figura 3.8: **Combinação de 5 imagens filtradas.** Na primeira linha, temos um intervalo de 5 imagens filtradas, e na segunda, os modelos mínimo e média, respectivamente.

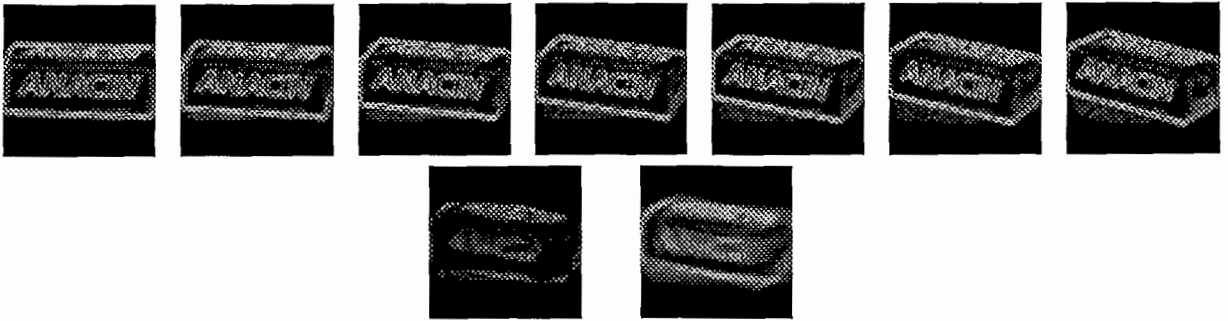


Figura 3.9: **Combinação de 7 imagens filtradas.** Na primeira linha, temos um intervalo de 7 imagens filtradas, e na segunda, os modelos mínimo e média, respectivamente.

Capítulo 4

Reconhecimento por Comparação

Foram estabelecidos dois grandes grupos de simulações, cada um voltado para a investigação das variações de condições apresentadas nas imagens dos objetos presentes nas bases de dados escolhidas. Uma destas é composta por faces humanas (objetos deformáveis) em várias poses, e a outra, por objetos rígidos rotacionados em seqüência.

Como já foi dito, Uma vez que a base de dados foi coletada e a representação interna da imagens definida, deve ser determinado o método de comparação entre as entradas e os modelos conhecidos, bem como uma medida de similaridade. Para medir esta semelhança, utilizamos tanto o produto interno normalizado quanto a função de ativação da camada de saída de uma rede ART Fuzzy [10, 11], como medidas de similaridade, procurando o modelo mais parecido com a imagem de entrada em teste.

4.1 Faces

Nas imagens do Olivetti Research Laboratory¹, temos 10 poses de 40 pessoas, de tamanho 92x112 pixels, em 256 tons de cinza (Figura 4.1). Estas imagens foram feitas variando iluminação, expressões faciais (olhos abertos/fechados, sorrindo/não sorrindo) e detalhes faciais (com/sem óculos). Com uma certa tolerância para movimentos de rotação, todas as pessoas estavam de frente para a câmera fotográfica.

¹Disponível em <http://www.camorl.co.uk/facedatabase.html>.

Pequenas variações de translação e escala também podem ser notadas.

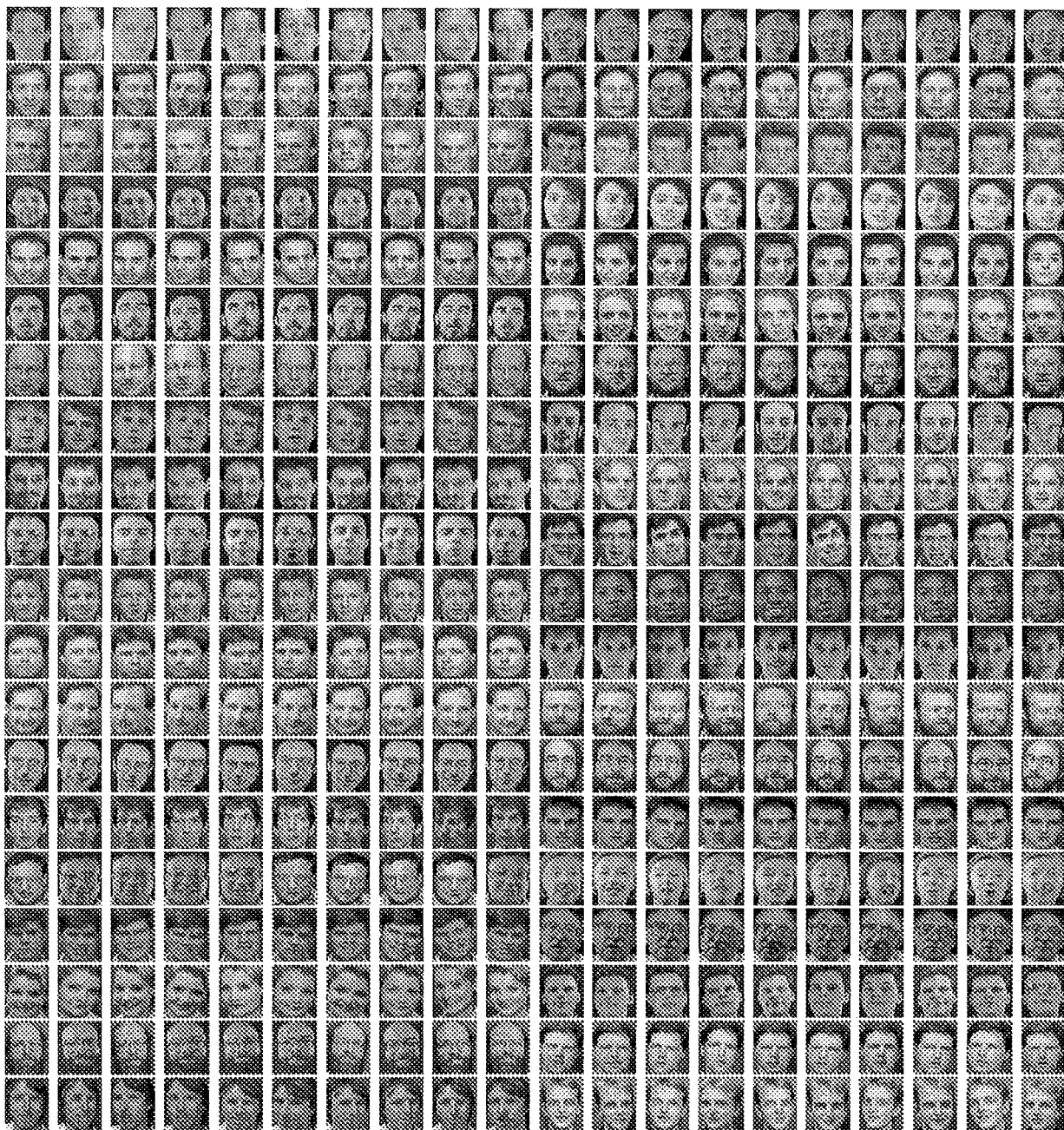


Figura 4.1: Todas as faces do Laboratório Olivetti, disponível em <http://www.camorl.co.uk/facedatabase.html>.

Usando o primeiro esquema proposto no Capítulo 3 como sistema de reconhecimento, com uma camada de entrada e outra de saída, cujas unidades representam diferentes objetos, aplicamos as duas medidas de similaridade estabelecidas, obtendo os seguintes resultados.

4.1.1 Testes e Resultados

O sistema implementado reconhece imagens a partir de uma coleção de modelos armazenada na memória. Num primeiro teste, procuramos estabelecer um único modelo para cada pessoa. Todas as imagens, tanto as usadas para a construção do modelo, quanto as usadas para teste, foram pré-processadas utilizando os filtros para extração de contorno, como os apresentados no Capítulo 2. No caso mais simples, em cada nível da pirâmide, uma única imagem foi armazenada como único modelo de cada pessoa, a ser comparado com as demais imagens. Foram executados testes com todas as 10 poses. Para mostrar as possíveis variações de resultados, as Figuras 4.2 e 4.3, apresentam dois exemplos de conjuntos de modelos (imagens não filtradas). Para este sistema, usando as duas medidas de similaridade, as outras imagens não usadas como modelo foram apresentadas como entrada a serem classificadas. O sistema obteve uma taxa de erro de, no máximo, 35% para as 360 imagens testadas, como pode ser visto na Tabela 4.1, a seguir, formada pela média dos 10 resultados.



Figura 4.2: **Imagens padrão – caso 1:** variabilidade da base de dados representada por uma pose de cada pessoa.

Outros modelos armazenando combinações de informação de várias imagens foram estabelecidos. Foram criados modelos na forma de *Mínimos* e *Médias*, cada



Figura 4.3: **Imagens padrão – caso 2:** variabilidade da base de dados representada por uma outra pose de cada pessoa.

	produto interno	função de ativação
nível 0	35.00%	31.39%
nível 1	29.72%	27.50%
nível 2	19.44%	16.11%
nível 3	17.78%	16.94%

Tabela 4.1: **Taxas de erro para modelos com uma única pose**, usando as duas medidas de similaridade.

um com sua função, definida no Capítulo 3, para combinar informação. No caso dos modelos de *Mínimos*, foi utilizado o valor 0.01 para o parâmetro α . Para comparar a qualidade da informação armazenada pelos dois tipos de combinação de informação (mínimos ou média), *todas* as imagens de cada pessoa foram armazenadas em um único modelo (Figura 4.4). Apresentando todas as imagens para teste, obtivemos as taxas de até 100% de acerto, mostradas nas Tabela 4.2.

Os bons resultados foram bastante surpreendentes, porém, *todas* as entradas foram utilizadas no estabelecimento da memória, não restando outra opção senão utilizá-las também para teste. Procuramos avaliar, então, o quanto o sistema perderia ao usarmos cada vez menos imagens para a geração do modelo. Definimos então, modelos com 7, 5 e 3 imagens. As Figuras 4.5, 4.6 e 4.7, a seguir, apresentam um

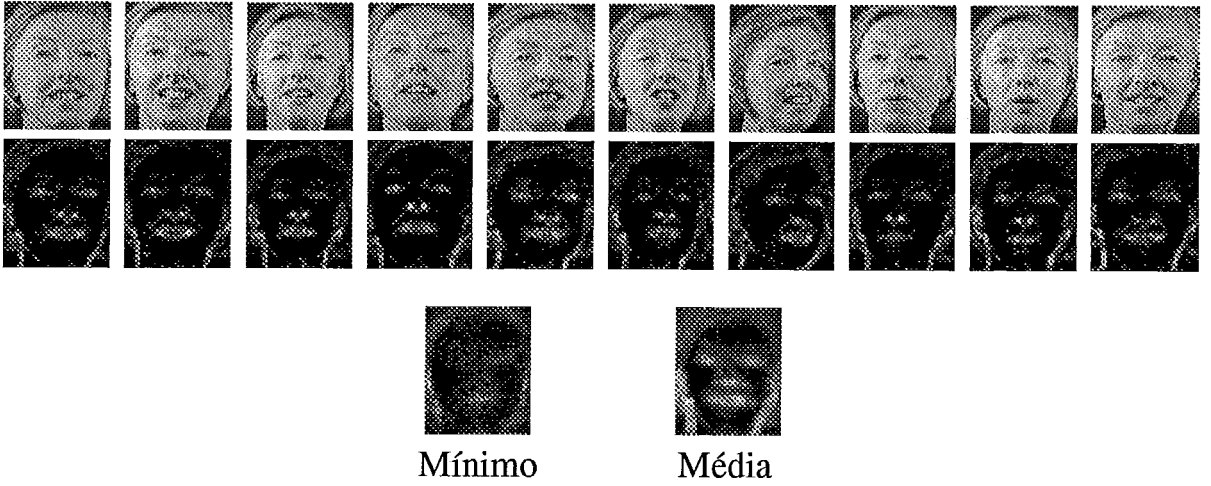


Figura 4.4: **Construção dos modelos** de mínimo e média, respectivamente, de *todas* as 10 imagens pré-processadas para esta pessoa apresentada na primeira linha.

	mínimo	média
nível 0	5.75 %	1.00 %
nível 1	8.00 %	0.25 %
nível 2	2.75 %	0.00 %
nível 3	6.25 %	1.25 %

Tabela 4.2: **Taxa de erro para mínimo e média de *todas* as imagens**, usando apenas a função de ativação como medida de similaridade.

exemplo de geração de modelos pela função de mínimo e de média. Dentre as 10 imagens, todas as combinações possíveis de 3, 5 e 7 modelos foram testadas.

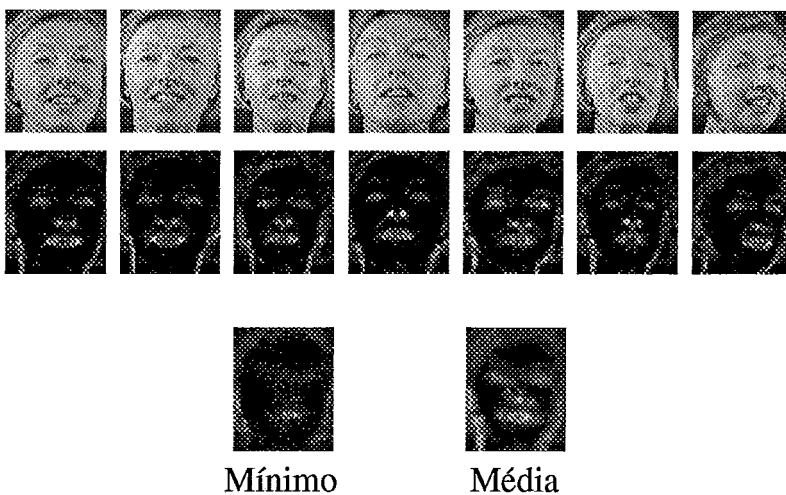


Figura 4.5: **Exemplo de construção dos modelos** de mínimo e média, respectivamente, para o grupo de 7 imagens apresentadas na primeira linha, depois de pré-processadas.

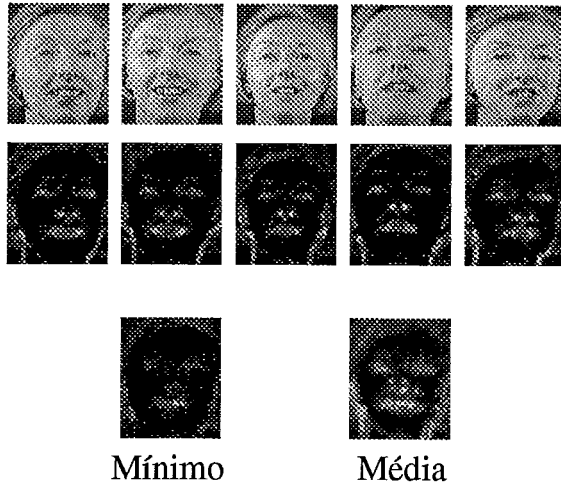


Figura 4.6: **Exemplo de construção dos modelos** de mínimo e média, respectivamente, para o grupo de 5 imagens apresentadas na primeira linha, depois de pré-processadas.



Figura 4.7: **Exemplo de construção dos modelos** de mínimo e média, respectivamente, para o grupo de 3 imagens apresentadas na primeira linha, depois de pré-processadas.

Para cada pessoa, as imagens não selecionadas para a criação dos modelos serviram para a definição do conjunto de teste, resultando nos seguintes gráficos apresentados nas Figuras 4.8 e 4.9. Podemos observar, pelas respectivas tabelas com as médias das taxas de erro de todas as combinações (Tabelas 4.3 e 4.4 usando produto interno normalizado e 4.5 e 4.6 usando função de ativação), que os melhores resultados são encontrados nos níveis 2 e 3 da pirâmide. Devido à compressão de dados, para estes níveis, cada pixel representa uma região de 32 ou 64 pixels, respectiva-

mente, da imagem original. Com isto, pequenas variações como olhos fechados, são eliminadas.

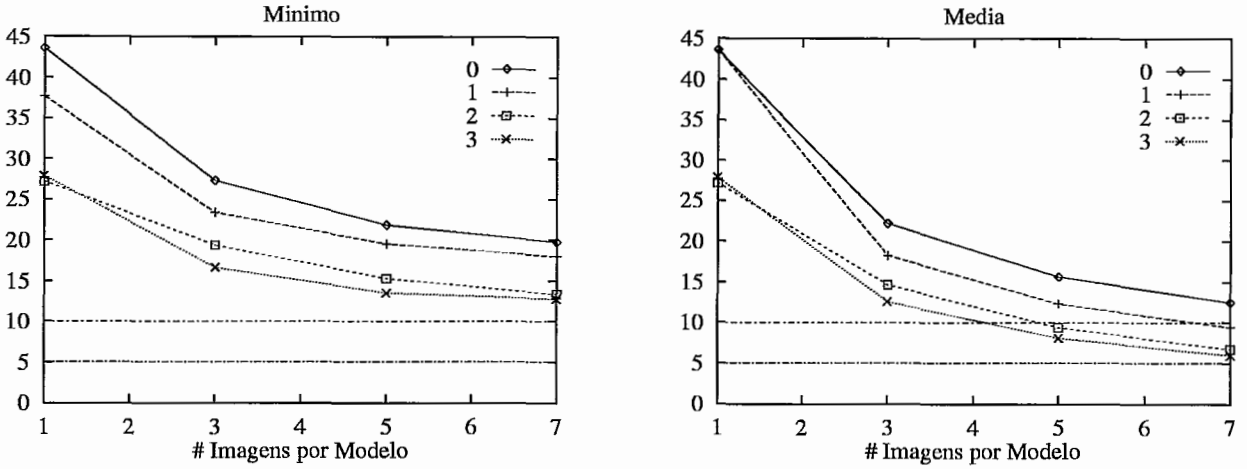


Figura 4.8: Taxa de erro *versus* tamanho do grupo para mínimos e médias utilizando o *produto interno normalizado* como medida de similaridade. Estes gráficos representam uma curva para cada nível da pirâmide, conforme os resultados das Tabelas 4.3 e 4.4

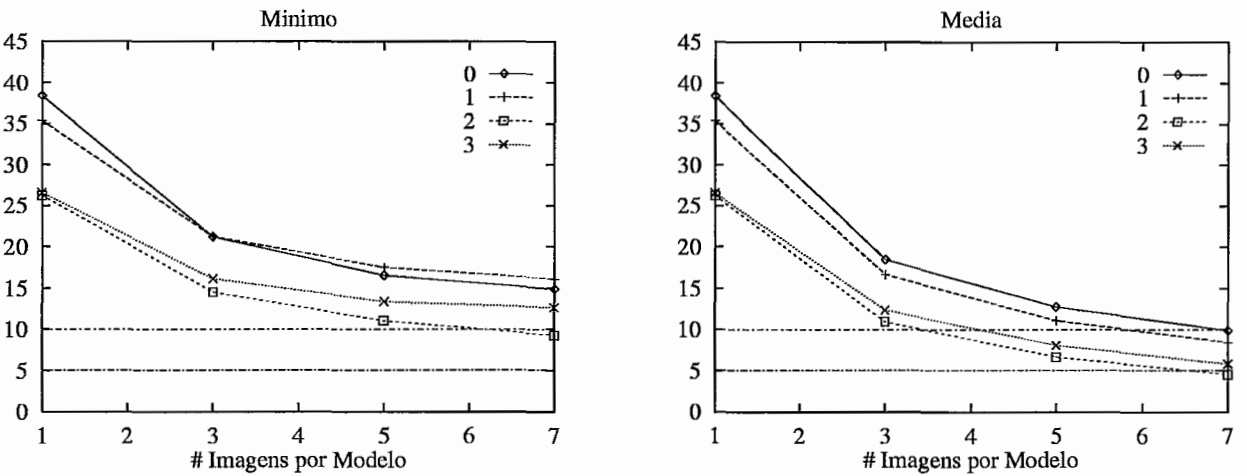


Figura 4.9: Taxa de erro *versus* tamanho do grupo para mínimos e médias utilizando a *função de ativação da rede ART* como medida de similaridade. Estes gráficos representam uma curva para cada nível da pirâmide, conforme os resultados das Tabelas 4.5 e 4.6

mínimo	7 imagens 120 entradas	5 imagens 200 entradas	3 imagens 280 entradas	1 imagem 360 entradas
nível 0	19.74%	21.83%	27.30%	43.62%
nível 1	17.99%	19.52%	23.35%	37.67%
nível 2	13.34%	15.27%	19.36%	27.22%
nível 3	12.75%	13.49%	16.63%	27.89%

Tabela 4.3: **Taxa de erro para mínimos.** Média dos resultados de todas as combinações, usando o *produto interno normalizado* como medida de similaridade.

media	7 imagens 120 entradas	5 imagens 200 entradas	3 imagens 280 entradas	1 imagem 360 entradas
nível 0	12.50%	15.66%	22.19%	43.62%
nível 1	9.41%	12.32%	18.27%	37.67%
nível 2	6.70%	9.38%	14.66%	27.22%
nível 3	5.97%	8.08%	12.58%	27.89%

Tabela 4.4: **Taxa de erro para médias.** Média dos resultados de todas as combinações, usando o *produto interno normalizado* como medida de similaridade.

mínimo	7 imagens 120 entradas	5 imagens 200 entradas	3 imagens 280 entradas	1 imagem 360 entradas
nível 0	14.88%	16.53%	21.20%	38.39%
nível 1	16.08%	17.52%	21.22%	35.39%
nível 2	9.27%	11.04%	14.53%	26.25%
nível 3	12.63%	13.38%	16.14%	26.58%

Tabela 4.5: **Taxa de erro para mínimos.** Média dos resultados de todas as combinações, usando a *função de ativação* como medida de similaridade.

media	7 imagens 120 entradas	5 imagens 200 entradas	3 imagens 280 entradas	1 imagem 360 entradas
nível 0	9.90%	12.76%	18.49%	38.39%
nível 1	8.44%	11.12%	16.68%	35.39%
nível 2	4.56%	6.65%	10.95%	26.25%
nível 3	5.82%	8.08%	12.37%	26.58%

Tabela 4.6: **Taxa de erro para médias.** Média dos resultados de todas as combinações, usando a *função de ativação* como medida de similaridade.

Para exemplificar a qualidade das respostas do sistema, nas Tabelas 4.7 e 4.8 são apresentados os melhores resultados para uma única combinação, tendo o produto interno como medida de similaridade. Enquanto que, nas Tabelas 4.9 e 4.10, temos os melhores resultados com a função de ativação.

mínimo	7 imagens 120 entradas	5 imagens 200 entradas	3 imagens 280 entradas	1 imagem 360 entradas
nível 0	14.17%	15.50%	21.43%	35.00%
nível 1	12.50%	13.50%	16.79%	29.72%
nível 2	8.33%	9.00%	14.28%	19.44%
nível 3	6.67%	8.50%	11.07%	17.78%

Tabela 4.7: **Menores taxas de erro para mínimos**, a melhor combinação para cada nível da pirâmide com 7, 5, 3 e 1 imagens, usando o *produto interno normalizado* como medida de similaridade.

media	7 imagens 120 entradas	5 imagens 200 entradas	3 imagens 280 entradas	1 imagem 360 entradas
nível 0	6.67%	10.00%	14.29%	35.00%
nível 1	4.17%	7.00%	10.35%	29.72%
nível 2	2.50%	4.00%	9.64%	19.44%
nível 3	0.83%	1.50%	6.79%	17.78%

Tabela 4.8: **Menores taxas de erro para médias**, a melhor combinação para cada nível da pirâmide com 7, 5, 3 e 1 imagens, usando o *produto interno normalizado* como medida de similaridade.

mínimo	7 imagens 120 entradas	5 imagens 200 entradas	3 imagens 280 entradas	1 imagem 360 entradas
nível 0	8.33%	11.00%	15.00%	31.39%
nível 1	10.00%	12.00%	14.64%	27.50%
nível 2	4.17%	5.00%	9.64%	16.11%
nível 3	7.50%	8.50%	11.07%	16.94%

Tabela 4.9: **Menores taxas de erro para mínimos**, a melhor combinação para cada nível da pirâmide com 7, 5, 3 e 1 imagens, usando a *função de ativação* como medida de similaridade.

Na tentativa de avaliar a qualidade dos resultados do nosso sistema, aproveitamos outros resultados obtidos com a mesma base de dados [34] (Tabela 4.11). Vale

media	7 imagens 120 entradas	5 imagens 200 entradas	3 imagens 280 entradas	1 imagem 360 entradas
nível 0	5.00%	7.50%	10.00%	31.39%
nível 1	2.50%	5.50%	8.57%	27.50%
nível 2	0.00%	1.00%	5.71%	16.11%
nível 3	0.00%	2.50%	6.79%	16.94%

Tabela 4.10: **Menores taxas de erro para médias**, a melhor combinação para cada nível da pirâmide com 7, 5, 3 e 1 imagens, usando a *função de ativação* como medida de similaridade.

ressaltar que nossos resultados foram obtidos com dados do nível 2 da pirâmide. Isto equivale a uma redução de custo computacional de até $4^3 = 64$ vezes em relação à imagem completa, já que cada nível tem imagens 4 vezes menores que o anterior, enquanto todos os outros métodos utilizam sempre a imagem inteira para criar seus modelos a partir de 1, 3 ou 5 imagens.

Imagens por pessoa	1	3	5
Eigenfaces – média por classes	38.6%	28.9%	26%
Eigenfaces – um por classe	38.6%	18.2%	10.5%
PCA + CN	34.2%	13.2%	7.5%
SOM + CN	30.0%	11.8%	3.8%
Melhor resultado: médias – nível 2	26.2%	10.9%	4.5%

Tabela 4.11: **Tabela comparativa** apresentada inicialmente no trabalho de Lawrence *et al.*. São apresentados os resultados utilizando 1, 3 ou 5 imagens para o estabelecimento do modelo de memória. Eatamos incluindo o nosso melhor resultado (dentre todas as combinações), que foi obtido utilizando média como forma de combinação, com a resolução do nível 2 da pirâmide de imagens e tendo como medida de similaridade a função de ativação.

Lawrence *et al.* [34] usaram um mapa auto-organizável, *Self-Organizing Map* (SOM), para reduzir a dimensionalidade da representação da entrada, e uma rede convolucional de 5 camadas (CN) para estabelecer invariância à translação e à deformação. eles afirmam que este procedimento é mais rápido que as abordagens convencionais de modelos de Markov, *Hidden Markov Models* (HMM), e tem uma performance parecida (ver [26]), mas ainda precisa de muito tempo para o treina-

mento.

Eles comparam a capacidade de redução de dimensão do SOM com Análise de Componentes Principais e a rede convolucional com um Perceptron de multi-camadas, *Multi-Layer Perceptron* (MLP). Esta última abordagem teve resultados muito ruins (cerca de 60%), principalmente quando muitas camadas intermediárias são utilizadas.

4.2 Objetos Rotacionados

A base de dados da Universidade de Columbia apresenta 20 objetos em 72 imagens de tamanho 128x128 pixels, em 256 tons de cinza (Figura 4.10). Nelas, os objetos estão dispostos em poses rotacionadas no eixo perpendicular ao plano do objeto (eixo z), de 5 em 5 graus, até a volta completa. A Figura 4.11 apresenta uma amostra simplificada da evolução desta rotação, tanto com as imagens originais quanto com a imagem pré-processada na maior resolução (nível 0 da pirâmide), enquanto a Figura 4.12 apresenta todas as poses encontradas para duas imagens desta base.

4.2.1 Testes e Resultados

No primeiro teste implementado, verificamos a capacidade de generalização do sistema. Foram criados vários modelos i com as imagens pré-processadas de cada objeto o , $M_{i,o}$ (Figura 4.13), representando conjuntos de poses, onde variam o número de imagens que compõe o modelo e o número de imagens de teste que devem ser classificadas por eles. Quanto maior o número de imagens utilizadas, tanto para o modelo quanto para teste, mais geral é a informação analisada, por envolver maiores rotações. Ao implementar os mesmos testes para diversos níveis da pirâmide, analisamos também a generalização criada pela variação na resolução dos modelos e das imagens. Nestes testes, aplicamos apenas a função de ativação descrita no Capítulo 3 como medida de similaridade.

Lembrando que estamos utilizando a estrutura de um sistema *feed forward* com



Figura 4.10: Todos os objetos da Universidade de Columbia. A base é formada por 72 imagens de cada objeto, variando rotação em profundidade e, para o caso dos objetos mais alongados (como os carros) com pequenos ajustes de escala. Ver Figura 4.12 a seguir.

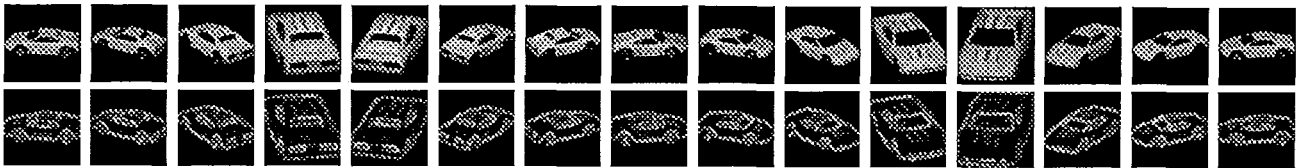


Figura 4.11: Rotações de 25 em 25 graus. Na primeira linha, as imagens originais e na segunda linha, as imagens utilizadas como entrada no sistema, filtradas conforme as definições do Capítulo 2.

uma camada de entrada e uma de saída, conforme o primeiro sistema apresentado na seção 3.2, temos todos os modelos armazenados como padrões em vetores de peso ($W_j = M_{i,o}$) conectando a camada de entrada à categoria que ele representa, como mostra o esquema na Figura 3.5. A saída deste sistema, é a categoria mais ativa, ou seja, qual modelo de que objeto se mostra mais similar a uma determinada entrada. Para este primeiro teste, uma resposta é considerada correta se o modelo escolhido for o mais próximo da imagem testada, dentre os vários modelos deste mesmo objeto (Figura 4.14).

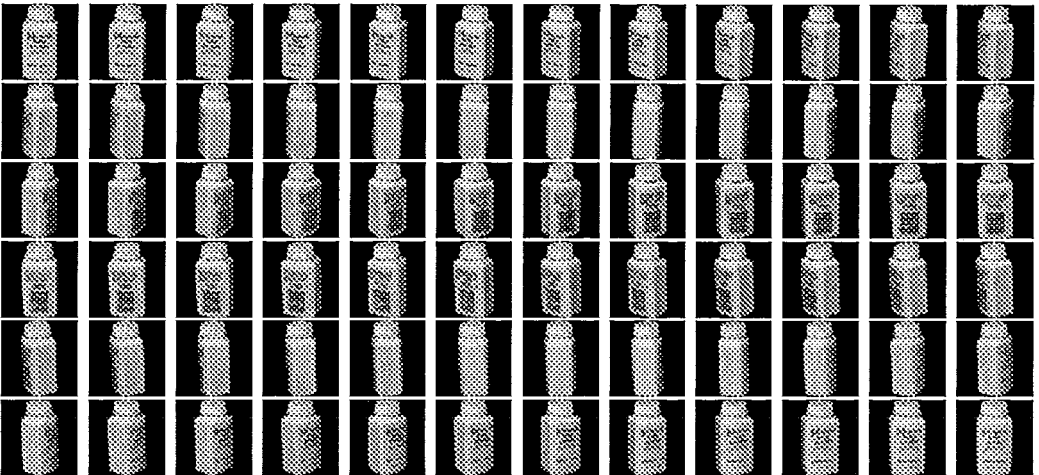
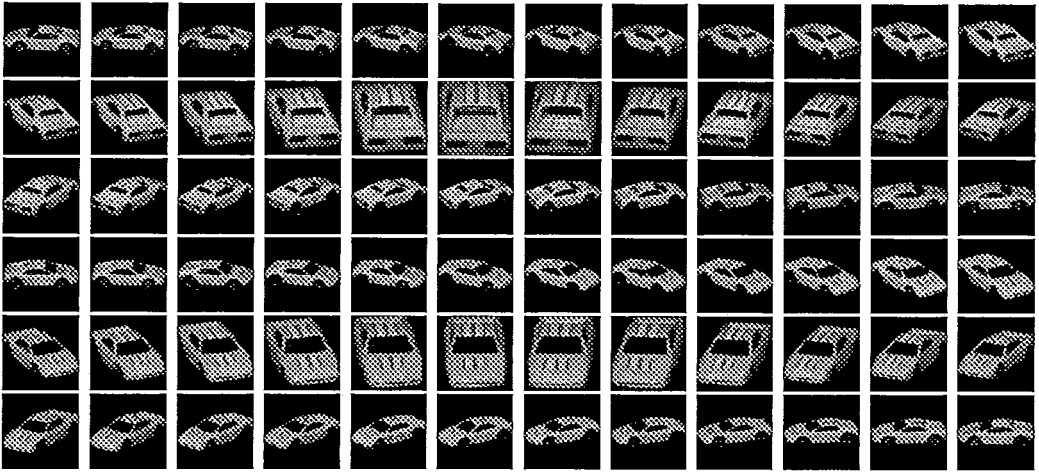


Figura 4.12: **Rotações de 5 em 5 graus.** No carro, um objeto alongado, a imagem sofre alterações de escala. Por outro lado, na embalagem de talco, um objeto simétrico em relação ao eixo de rotação, a escala se mantém fixa.

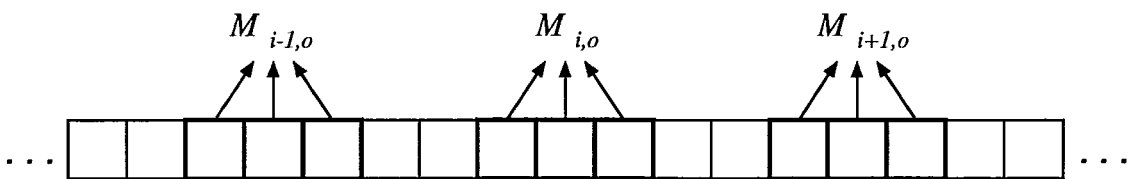


Figura 4.13: **Escolha de imagens para construção de cada modelo $M_{i,o}$.** As imagens que não são escolhidas para compor os modelos são usadas para teste. Os dois parâmetros tg (tamanho do grupo) e ts (tamanho do salto) definem, respectivamente, quantas imagens são usadas para a construção dos modelos e para teste, e, portanto, o número de modelos para cada objeto.

Nesta base de dados, temos 72 imagens de cada objeto em seqüência, dispostos de 5 em 5 graus. Desta forma, encontramos imagens do mesmo objeto de frente, de lado e de costas. Para tanto, um único modelo correria o risco de degenerar tal

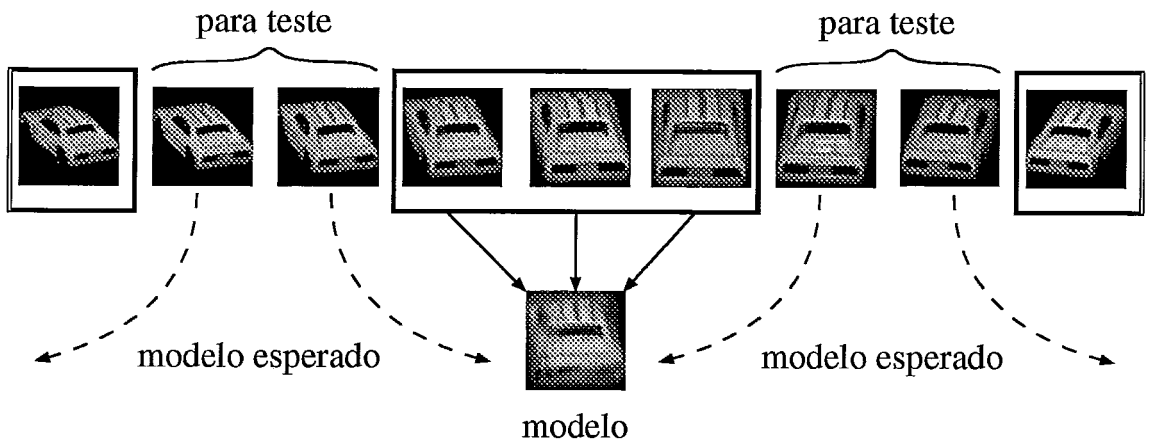


Figura 4.14: **Modelo esperado.** Construção de modelos a partir de um conjunto de imagens. Conjunto de teste é composto pelas imagens não utilizadas. A resposta esperada para cada teste é o modelo mais próximo da imagem. Considerando que nesta base de dados os objetos estão dispostos de 5 em 5 graus, o mais próximo deve ser sempre um dos modelos adjacentes na seqüência.

variedade de informações. Para evitar isto, criamos vários modelos para cada objeto, compostos de pequenas seqüências de imagens adjacentes, e conjuntos de testes escolhidos em posições intermediárias a estes conjuntos, como pode ser observado na Figura 4.14. Para dimensionar o número de modelos e de imagens para testes, criamos dois parâmetros definidos como o *tamanho do grupo* que deu origem ao modelo (tg), e o *tamanho do salto* entre estes grupos (ts). Desta forma, podemos ter conjuntos de teste de apenas 28 ($ts = 2$ e $tg = 3$) ou de até 60 imagens por objeto ($ts = 6$ e $tg = 1$). Os resultados são apresentados em valores relativos, usando a porcentagem de erros em relação à dimensão do conjunto de teste.

Nas Tabelas 4.12 e 4.13 encontramos os resultados destes primeiros testes e podemos acompanhar as mesmas respostas à variação dos parâmetros do sistema pelos gráficos das Figuras 4.15, 4.16 e 4.17.

Pela forte simetria de alguns objetos em relação ao eixo z , a rotação trouxe poucas mudanças, gerando modelos adjacentes com correlações muito altas. Como conseqüência, numa média de 46% dos erros, os modelos escolhidos eram adjacentes ao correto. Além disso, quanto maior o número de imagens em um grupo, maiores as chances de ocorrer uma classificação errada. Como mostram as Tabelas 4.12 e

nível 0	$tg = 3$	$tg = 2$	$tg = 1$
$ts = 2$	47.32%	31.53%	16.15%
$ts = 4$	39.12%	27.19%	23.66%
$ts = 6$	32.71%	33.52%	26.17%
nível 1	$tg = 3$	$tg = 2$	$tg = 1$
$ts = 2$	47.32%	30.69%	14.69%
$ts = 4$	38.00%	27.40%	22.59%
$ts = 6$	33.13%	33.43%	25.33%
nível 2	$tg = 3$	$tg = 2$	$tg = 1$
$ts = 2$	47.50%	31.39%	13.33%
$ts = 4$	39.50%	26.46%	20.45%
$ts = 6$	32.40%	33.06%	24.67%
nível 3	$tg = 3$	$tg = 2$	$tg = 1$
$ts = 2$	48.04%	29.72%	12.92%
$ts = 4$	38.75%	24.17%	19.29%
$ts = 6$	31.87%	32.04%	23.83%

Tabela 4.12: Taxas de erro para os testes com modelos de *Mínimos* de objetos rotacionados (cada modelo é uma categoria). Os resultados estão separados conforme o nível da pirâmide utilizado para o teste e organizados pelo tamanho do grupo (tg) e pelo tamanho do salto (ts).

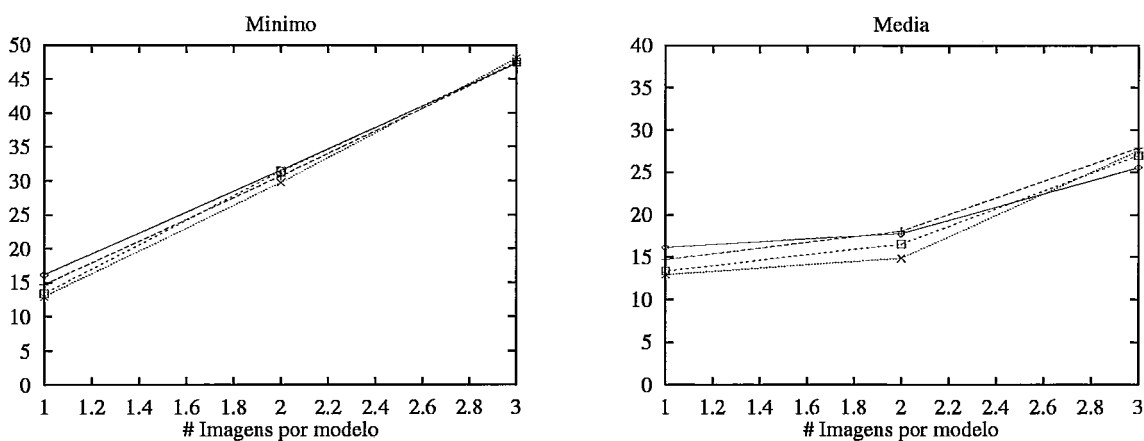


Figura 4.15: Taxa de erro *versus* tamanho do grupo utilizando modelos de mínimos e médias em um conjunto de teste com intervalos de tamanho 2. Cada curva representa um nível da pirâmide, ou seja, uma diferente resolução dos dados.

4.13, aumentando o número de imagens no intervalo entre os modelos, as taxas de erro também aumentam.

O único caso onde sempre houve melhoras com o uso de níveis mais baixos da pirâmide foi com modelos de uma única imagem. Neste caso, não havia uma

nível 0	$tg = 3$	$tg = 2$	$tg = 1$
$ts = 2$	25.54%	17.78%	16.15%
$ts = 4$	29.00%	21.46%	23.66%
$ts = 6$	25.52%	31.39%	26.17%
nível 1	$tg = 3$	$tg = 2$	$tg = 1$
$ts = 2$	27.86%	18.06%	14.69%
$ts = 4$	28.37%	20.42%	22.59%
$ts = 6$	26.56%	31.57%	25.33%
nível 2	$tg = 3$	$tg = 2$	$tg = 1$
$ts = 2$	26.96%	16.53%	13.33%
$ts = 4$	29.62%	20.31%	20.45%
$ts = 6$	29.06%	30.46%	24.67%
nível 3	$tg = 3$	$tg = 2$	$tg = 1$
$ts = 2$	27.50%	14.86%	12.92%
$ts = 4$	29.25%	20.31%	19.29%
$ts = 6$	26.56%	30.46%	23.83%

Tabela 4.13: Taxas de erro para os testes com modelos de *Médias de objetos rotacionados* (cada modelo é uma categoria). Os resultados estão separados conforme o nível da pirâmide utilizado para o teste e organizados pelo tamanho do grupo (tg) e pelo tamanho do salto (ts).

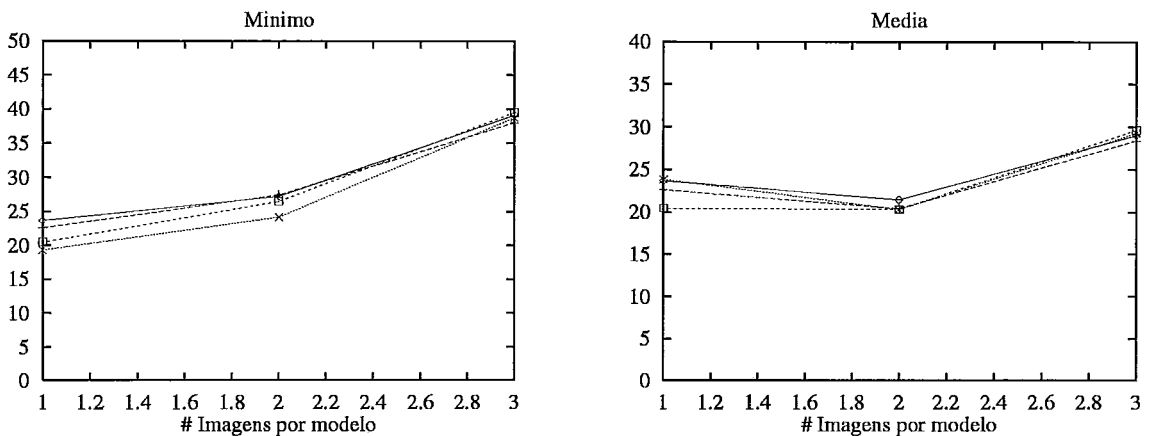


Figura 4.16: Taxa de erro *versus* tamanho do grupo utilizando modelos de mínimos e médias em um conjunto de teste com intervalos de tamanho 4. Cada curva representa um nível da pirâmide, ou seja, uma diferente resolução dos dados.

combinação de informações generalizando as imagens do intervalo, como na média e no mínimo. Assim, a baixa resolução tem seu papel enfatizado, os pixels passam a conter informações mais genéricas sobre os objetos de cada imagem.

Quando analisamos rotações no eixo z de objetos não simétricos, como faces hu-

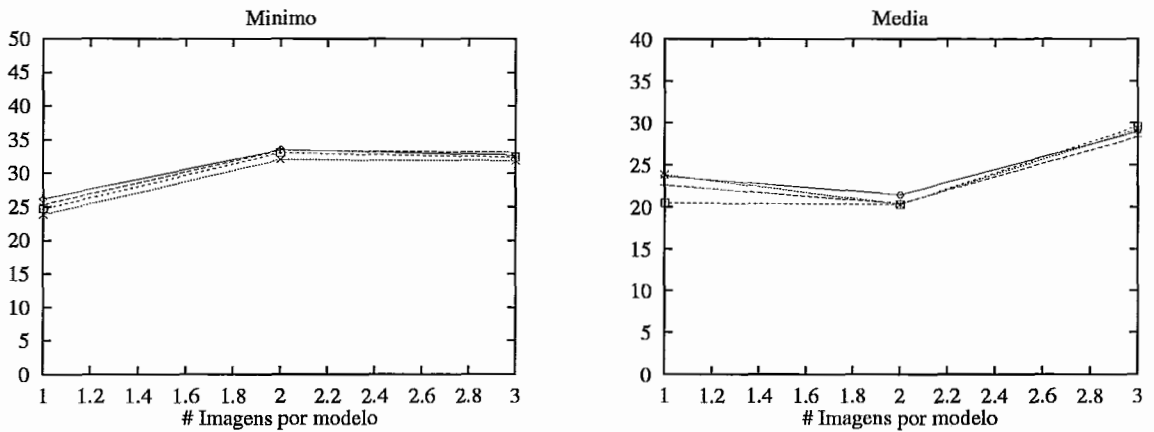


Figura 4.17: **Taxa de erro *versus* tamanho do grupo** utilizando modelos de mínimos e médias em um conjunto de teste com **intervalos de tamanho 6**. Cada curva representa um nível da pirâmide, ou seja, uma diferente resolução dos dados.

manas, criamos naturalmente categorias que classificam as faces que estamos vendo: de frente, de costas, de perfil, de meio-perfil, etc. Porém, qualquer que seja a *categoria*, as imagens são reconhecidas como poses da mesma pessoa. Ao invés de analisarmos cada modelo criado para o objeto, devemos considerar o conjunto de modelos como uma possível resposta para este objeto. Para tanto, devemos considerar como correta a classificação de uma nova imagem de um objeto por *qualquer* um dos modelos criados para ele. Desta forma, artificialmente criamos a categorização das posições aproveitando a conhecida disposição das poses em cada imagem. Alguns sistemas como o de Beymer [3] procura descobrir a pose da face representada na imagem para, em uma segunda fase, buscar o reconhecimento apenas dentre imagens da mesma categoria, ou seja, frontal com frontal, perfil com perfil, etc.

Os testes são feitos da mesma forma, armazenando os modelos nos vetores de peso de um sistema *feed forward* conforme o segundo sistema proposto no Capítulo 3 (Figura 3.6), onde uma entrada é apresentada e, por uma determinada medida de similaridade (função de ativação), a categoria do modelo mais parecido é ativada. No entanto, esta resposta serve como entrada para uma próxima camada do sistema, onde todas as categorias de um mesmo objeto estão conectadas a um único nó. Desta forma, todos os modelos de um objeto geram uma resposta única

nível 0	<i>tg</i> = 3	<i>tg</i> = 2	<i>tg</i> = 1
<i>ts</i> = 2	3.21%	1.81%	1.46%
<i>ts</i> = 4	6.63%	5.63%	4.11%
<i>ts</i> = 6	6.37%	6.30%	6.83%
nível 1	<i>tg</i> = 3	<i>tg</i> = 2	<i>tg</i> = 1
<i>ts</i> = 2	2.26%	2.08%	0.94%
<i>ts</i> = 4	6.63%	6.46%	3.93%
<i>ts</i> = 6	10.21%	5.93%	6.83%
nível 2	<i>tg</i> = 3	<i>tg</i> = 2	<i>tg</i> = 1
<i>ts</i> = 2	2.86%	2.78%	1.35%
<i>ts</i> = 4	7.13%	6.25%	3.48%
salto 6	9.58%	5.19%	5.33%
nível 3	<i>tg</i> = 3	<i>tg</i> = 2	<i>tg</i> = 1
<i>ts</i> = 2	2.86%	2.92%	1.15%
<i>ts</i> = 4	7.62%	5.83%	3.39%
<i>ts</i> = 6	9.27%	5.46%	5.92%

Tabela 4.14: **Taxas de erro para os testes com modelos de *Mínimos de objetos rotacionados***. Independente da pose da imagem modelo. Os resultados estão separados conforme o nível da pirâmide utilizado para o teste e organizados pelo tamanho do grupo (*tg*) e pelo tamanho do salto (*ts*).

no sistema, independente de qual deles foi escolhido. Além dos ótimos resultados, as Tabelas 4.14 e 4.15 apresentam o mesmo comportamento do teste anterior, onde as combinações pela média obtêm as menores taxas de erro, alcançando resultados menores do que 1%. Novamente, quanto maiores os conjuntos de imagens para criar cada modelo, maiores os erros. Da mesma forma, quanto maiores os conjuntos de teste selecionados, maior a variação das imagens representadas e, portanto, maiores os erros.

nível 0	<i>tg</i> = 3	<i>tg</i> = 2	<i>tg</i> = 1
<i>ts</i> = 2	3.75%	2.50%	1.46%
<i>ts</i> = 4	6.50%	4.58%	4.11%
<i>ts</i> = 6	7.60%	5.93%	6.83%
nível 1	<i>tg</i> = 3	<i>tg</i> = 2	<i>tg</i> = 1
<i>ts</i> = 2	4.46%	2.64%	0.94%
<i>ts</i> = 4	7.75%	5.31%	3.93%
<i>ts</i> = 6	8.75%	5.56%	6.83%
nível 2	<i>tg</i> = 3	<i>tg</i> = 2	<i>tg</i> = 1
<i>ts</i> = 2	5.18%	1.94%	1.35%
<i>ts</i> = 4	7.25%	5.00%	3.48%
<i>ts</i> = 6	10.00%	5.37%	5.33%
nível 3	<i>tg</i> = 3	<i>tg</i> = 2	<i>tg</i> = 1
<i>ts</i> = 2	4.82%	1.94%	1.15%
<i>ts</i> = 4	6.62%	4.58%	3.39%
<i>ts</i> = 6	9.17%	5.46%	5.92%

Tabela 4.15: Taxas de erro para os testes com modelos de *Médias de objetos rotacionados*. Independente da pose da imagem modelo. Os resultados estão separados conforme o nível da pirâmide utilizado para o teste e organizados pelo tamanho do grupo (*tg*) e pelo tamanho do salto (*ts*).

Capítulo 5

Reconhecimento com Redes Neurais RBF

Nos capítulos anteriores, embora tenhamos construído um sistema de comportamento bem parecido com uma rede *feed forward*, não chegamos a utilizar a parte mais importante de uma rede neural: o aprendizado. Esta fase foi substituída pela sintetização dos modelos a serem armazenados na memória em lugar dos protótipos que seriam gerados, pela rede neural, para a representação de cada categoria. Neste capítulo, nos aprofundaremos na implementação e discussão de uma rede neural do tipo *feed forward* denominada *Radial Basis Function* (RBF). Estas redes já foram utilizadas para reconhecimento de objetos tridimensionais tão simples quanto cliques retorcidos [5, 51] e outros muito mais complexos como animais, xícaras, carros, etc. [14, 18]

Vários casos abordam o problema de reconhecimento de faces com a translação e a rotação no plano ou em profundidade [26, 27]. Por exemplo, Edelman *et al.* [14] apresentam um sistema para reconhecer faces humanas em diferentes poses. Após a normalização da aparência das faces (baseado em operadores de simetria), executam uma redução de dimensão, pela simples aplicação de um conjunto de campos receptores Gaussianos, cujo resultado são apresentados para um classificador baseado em uma rede neural RBF [24, 25].

Encontramos vários estudos que aplicam diversos métodos para representação, classificação e identificação de objetos, mostrando uma abordagem interessante para

o tratamento de faces. Eles são centrados na noção de que um rosto é, afora qualquer outra transformação geométrica – translação, rotação, escala – um objeto de natureza deformável [6, 31, 34]. Portanto, se torna interessante procurar uma forma de criar sistemas insensíveis à mudança de expressão, presença e ausência de óculos e outras pequenas mudanças do dia-a-dia, que geram grandes problemas para o reconhecimento em um sistema artificial. De uma forma semelhante, o trabalho apresentado por Tomaz *et al.* [58] utiliza a base de dados de ORL (ver Figura 4.1) como entrada para um sistema de reconhecimento aplicando RBF. Para a representação interna, entretanto, eles optaram pela redução de dimensão oferecida pela Análise de Componentes Principais.

Seguindo a idéia de objetos deformáveis, para o caso do reconhecimento de faces num sistema real, como a identificação de visitantes [57], a interpolação entre várias expressões faciais seria mais interessante do que uma simples interpolação entre dois pontos de vista (ângulos em relação ao observador). Um trabalho de Beymer e Poggio [4] vai ainda mais além na questão da interpolação de expressões, sintetizando imagens intermediárias às faces estabelecidas como memória do sistema. Além disto, um exemplo de uso de RBF com a deformação das faces pode ser encontrado no trabalho de Rosenblum *et al.* [55] onde um sistema é desenvolvido para classificar a emoção expressa pelo rosto com uma rede neural RBF.

5.1 Outros Métodos

Uma vez que a base de dados foi coletada e a representação interna da imagens definida, deve ser determinado o método de comparação entre as entradas e os modelos conhecidos. Este método pode ser tão simples quanto uma comparação, como pode ser encontrado em diversos trabalhos [3, 6, 15], que utilizam como medidas de similaridade a distância euclideana, o produto interno normalizado ou a função de ativação de uma rede ART [10, 11, 22]. Os sistemas propostos no Capítulo

3, onde um vetor de pesos armazena o protótipo de cada categoria estabelecendo as representações internas dos modelos de memória, apresentam a aplicação deste método. Tradicionalmente, a comparação pode ser vista como uma técnica bastante útil em tarefas visuais de menor nível, tal como localização e identificação de padrões, baseadas em simples correlações entre vetores de imagens.

No entanto, o aprendizado é um fator importante em qualquer aplicação para evitar a inflexibilidade normalmente encontrada em sistemas com regras extraídas manualmente [26]. Mesmo tarefas visuais simples têm tanta complexidade, que as hipóteses iniciais de um sistema manual podem deixar de ser válidas, ou passarem a ser válidas apenas em determinadas circunstâncias. Além disto, uma abordagem deste tipo não é modificável em operações no dia-a-dia. Por exemplo, se uma tarefa se altera devido a mudanças no ambiente ou na base de dados envolvida, o sistema deve ser capaz de, automaticamente, se adaptar às novas condições.

Mesmo com as limitações computacionais, redes neurais têm uma longa história em aplicações para reconhecimento de objetos, em particular de faces. As redes associativas de Kohonen [33] desde muito tempo são capazes de demonstrar uma das maiores vantagens do processo distribuído em redes neurais, por gerar uma tolerância à ruídos ou dados incompletos. Millward e O'Toole [40], por exemplo, usaram um modelo de memória de Kohonen para codificar segmentos de bordas no lugar de tons de cinza (intensidade luminosa). Mel e Fiser [38, 39], por outro lado, utilizaram técnicas como a classificação pelo vizinho mais próximos (*Nearest Neighbor*).

5.2 Radial Basis Function

Um sistema baseado na rede neural RBF deve aprender a interpolar entre as diversas poses ou combinações representativas das condições em que os objetos se apresentam. Como já foi comentado, para detectar se duas imagens mostram o mesmo objeto

tridimensional, o sistema deve superar a influência dos diversos fatores que afetam suas representações bidimensionais. Uma solução proposta por Edelman [14, 18] é expandir as estratégias baseadas em comparação com modelos de memória para a categorização. Para tal, são criados módulos que devem responder com um valor próximo de 1 para qualquer imagem de um determinado objeto e com um valor próximo de 0 para todas as imagens de outros objetos do conjunto.

Neste trabalho, utilizamos o modelo de RBF Gaussiana proposto por Poggio e Girosi [52, 53]. Este modelo apresenta três camadas: entrada, intermediária e saída. Da forma como o sistema foi implementado, o estabelecimento de pesos entre as camadas intermediária e de saída se dá de forma supervisionada, enquanto a relação entre as camadas de entrada e intermediária se dá pelo estabelecimento de funções radiais ajustadas a partir do conjunto de treinamento (ver Figura 5.1), simulando o comportamento de campos receptores da entrada.

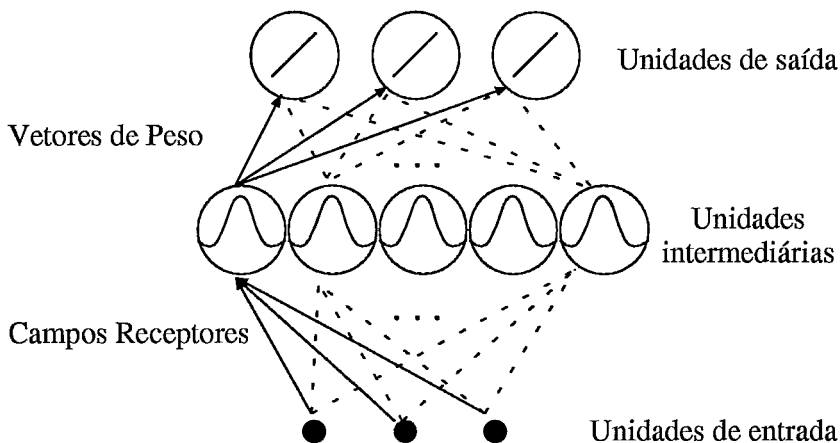


Figura 5.1: **Organização de uma rede neural RBF:** o aprendizado dos pesos entre as unidades intermediária e de saída (linear) é supervisionado, enquanto que, entre as unidades de entrada e intermediárias (radial) é estabelecido pelos ajustes imediatos dos parâmetros das funções radiais a partir do conjunto de treinamento.

5.2.1 Definições

O comportamento da rede neural RBF é motivado por *respostas ajustadas localmente* como pode ser observado pela biologia em várias partes do sistema nervoso [24]. Estes neurônios têm, por características, serem *seletivos* a um intervalo finito

do espaço do sinal de entrada. Desta forma, a rede RBF tem estrutura *feed forward* consistindo de uma única camada intermediária com H unidades ajustadas localmente e totalmente conectadas com as unidades lineares da camada de saída. Como pode ser acompanhado na Figura 5.2, todas as unidades intermediárias recebem simultaneamente o vetor n -dimensional de entrada I . Note a ausência de pesos ligando a camada de entrada e a camada intermediária. Isto ocorre porque as saídas dos neurônios da camada intermediária não usam o mecanismo de ativação de somatório e sigmóide como numa rede neural *Backpropagation* [24, 25]. Ao invés disto, cada saída intermediária é obtida pelo cálculo da “proximidade” da entrada I e um vetor n -dimensional c_h , associado a h -ésima unidade intermediária.

A fórmula mais geral¹ para qualquer função radial de uma rede RBF é dada pela Equação 5.1:

$$\phi_h(x) = \mathcal{K}((x - c_h)^T \mathbf{R}^{-1} (x - c_h)), \quad (5.1)$$

onde \mathcal{K} é uma função radialmente simétrica e estritamente positiva com um único valor máximo, c_h é o centro para a unidade h e \mathbf{R} é uma medida de similaridade. O termo $(x - c_h)^T \mathbf{R}^{-1} (x - c_h)$ é a distância entre a entrada I e os centros c_h na métrica definida por \mathbf{R} . Geralmente esta métrica é a distância Euclideana. Neste caso $\mathbf{R} = r^2 I$ para algum raio escalar r e a equação acima pode ser simplificada para:

$$\phi_h(x) = \mathcal{K}\left(\frac{\|x - c_h\|}{r^2}\right), \quad (5.2)$$

onde \mathcal{K} é uma função radialmente simétrica e estritamente positiva com um único valor máximo no seu centro c_h e que decai rapidamente para zero ao se afastar do mesmo [14, 44, 58].

O raio é a *largura* do campo receptivo no espaço de entrada para a unidade h .

¹Para eliminar a necessidade de uma unidade “bias” é feita a normalização das saída da camada intermediária.

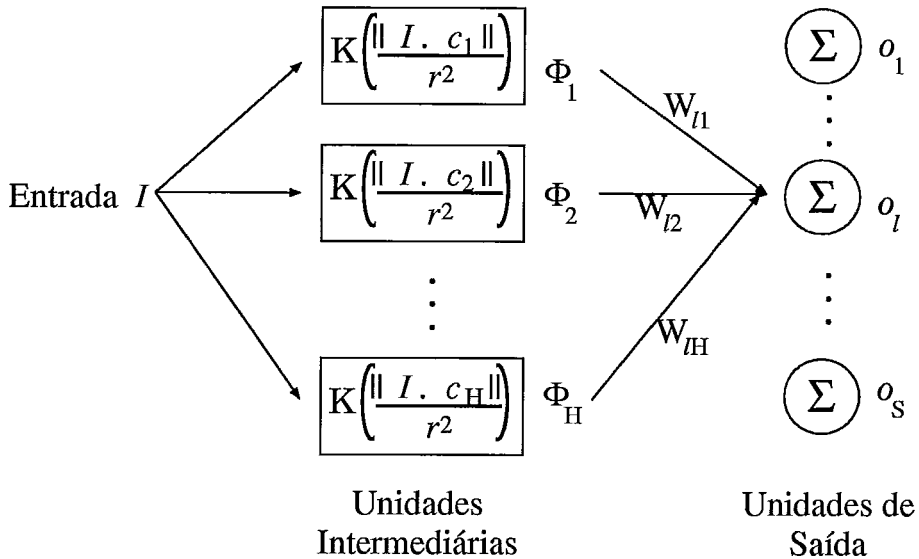


Figura 5.2: **Organização de uma rede neural RBF:** A saída da primeira camada é dada pela aplicação de uma função radial ao resultado da diferença entre a informação de entrada e um vetor de centros c_h definido durante o treinamento. A resposta da camada de saída é uma soma ponderada da saída da camada intermediária pelos vetores de peso, também estabelecidos durante o treinamento.

Isto implica em ϕ_h só ter valores interessantes quando a distância $\|I - c_h\|$ for menor do que a largura da função radial estabelecida. Dado um vetor de entrada I , a saída de uma rede RBF é o vetor S -dimensional de atividade, cujo i -ésimo componente é dado pela Equação 5.3.

$$o_i = \sum_{h=1}^H W_{ih} \phi_h(I), \quad (5.3)$$

onde S é o número de unidades de saída e ϕ_h é calculado pela Equação anterior.

Redes neurais RBF são adequadas para aproximação de funções $f : \mathcal{R}^n \rightarrow \mathcal{S}$, onde n é pequeno o suficiente; estes problemas de aproximação incluem a classificação como um caso especial. Pelas Equações 5.2 e 5.3, logo acima, a rede RBF pode ser vista como uma aproximação da função desejada $f(I)$ pela superposição de funções não ortogonais em forma de sino [44]. O grau de precisão pode ser controlado por três parâmetros: o número de funções utilizadas (unidades da camada intermediária), suas localizações (centros) e suas larguras. Poggio e Girosi [52] apresentam vários teoremas para mostrar que as redes RBF são aproximadores

universais de funções, como as demais redes neurais *feed forward* com uma única camada intermediária de unidades com funções sigmóides. Uma rede RBF especial, porém, muito usada assume a função Gaussiana como a função radial da camada intermediária, está descrita na Equação 5.4:

$$\phi_h(I) = \exp\left(-\frac{\|I - c_h\|^2}{2\sigma_h^2}\right), \quad (5.4)$$

tal que σ_h e c_h são o desvio padrão e a média do h -ésimo campo receptivo, e a norma $\|\cdot\|$ é calculada pela distância euclideana (Equação 5.5). Outras funções radiais são sugeridas por Orr [44], descritas na Equações 5.6, 5.7 e 5.8, podem ser comparadas a Equação Gaussiana (5.4) no gráfico apresentado pela Figura 5.3 a seguir.

$$\|A - B\| = \sqrt{\sum_{x=1}^N (A_x - B_x)^2}, \text{ onde A e B são vetores.} \quad (5.5)$$

$$\phi_h(I) = \sqrt{1 + \left(\frac{\|I - c_h\|^2}{2\sigma_h^2}\right)}. \quad (5.6)$$

$$\phi_h(I) = \sqrt{1 + \left(\frac{\|I - c_h\|^2}{2\sigma_h^2}\right)^{-1}}. \quad (5.7)$$

$$\phi_h(I) = \left(1 + \frac{\|I - c_h\|^2}{2\sigma_h^2}\right)^{-1}. \quad (5.8)$$

5.2.2 Algumas Vantagens

Pelos trabalhos de Ullman [60, 61] podemos deduzir que é possível modelar uma tarefa de reconhecimento baseada na pose usando combinações lineares de poses em imagens bidimensionais para representar qualquer imagem bidimensional de um objeto. Uma abordagem mais simples consistiria em estabelecer técnicas de interpolação de poses [5, 51] para aprender a tarefa explicitamente. As maiores vantagens de empregar uma rede neural como RBF para tentar resolver a questão do reconhecimento

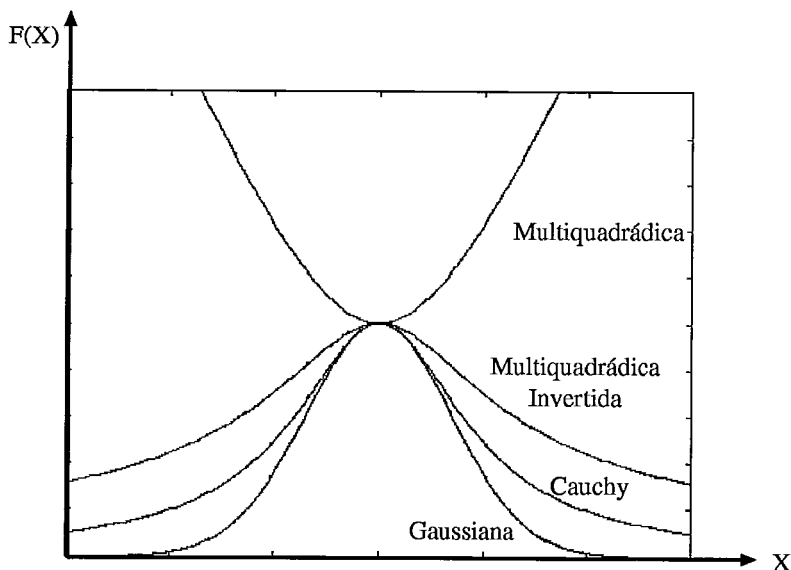


Figura 5.3: **Funções radialmente simétricas e estritamente positiva** com um único valor máximo no seu centro c_h e que decai rapidamente para zero ao se afastar do mesmo. São sugeridas por Orr (ver citação no texto) para serem empregadas nas camadas intermediárias de redes neurais RBF. A função mais utilizada é a Gaussiana.

são a simplicidade computacional, garantida por uma teoria matemática bem desenvolvida, e uma capacidade de generalização bastante robusta. Estas redes foram identificadas como ótimas para aplicações práticas de problemas visuais por Girosi [19], pois são eficientes para tratar com dados esparsos de alta dimensionalidade (como imagens) e porque conseguem contornar os efeitos dos ruídos, geralmente apresentados em dados reais.

Em resumo, uma vez que os exemplos de treinamento foram definidos como pares entrada-saída, ou seja, com a classe esperada associada a cada imagem, as tarefas podem ser simplesmente aprendidas pelo sistema. Este tipo de aprendizado supervisionado pode ser visto, em termos matemáticos, como uma aproximação de uma função multi-variável. Desta forma, estimativas dos valores da função podem ser feitas para dados de teste, cujos resultados não são conhecidos. Este processo pode ser feito pela rede RBF usando uma combinação linear de funções radiais (uma para cada exemplo de treinamento) por causa da suavidade do espaço n -dimensional formado por exemplos de poses de objetos num espaço de todas as possíveis deste

objeto [51].

A maior vantagem da rede RBF sobre outros tipos de redes neurais é o nível de confiança refletido diretamente no nível de cada unidade. Isto ocorre porque regiões no espaço de entrada que estejam longe dos vetores de treinamento são sempre mapeadas para gerar valores baixos, dada a natureza local das unidades intermediárias dos campos receptivos (funções que decaem rapidamente ao se afastar do centro). Além disso, a normalização da camada intermediária de rede RBF faz com que a saída represente a probabilidade da presença de suas classes.

5.3 Implementação

Utilizamos o modelo de RBF Gaussiana proposto por Poggio e Girosi [52, 53]. Este modelo combina uma camada supervisionada unindo as unidades intermediárias e de saída, com uma camada de ajustes imediatos unindo as unidades de entrada e intermediárias (ver Figura 5.1, na seção anterior). O modelo desta rede é caracterizado por funções Gaussianas individuais para cada unidade intermediária, simulando o comportamento de campos receptores da entrada.

5.3.1 Unidades de Reconhecimento

Neste trabalho, criamos um sistema com várias pequenas redes RBF, cada uma treinada para reconhecer um único objeto (Figura 5.4). Este sistema é facilmente adaptado a novos objetos, bastando treinar uma nova unidade para cada um destes e adicioná-las. Cada unidade é treinada com poses do objeto correspondente (evidências positivas) e algumas poses selecionadas de outros objetos (evidências negativas) e está conectada a apenas uma saída correspondendo ao quanto a entrada apresentada é semelhante ao objeto representado nesta unidade. Este treinamento usando exemplos negativos explícitos é usado em contraste com o esquema da rede apresentado por Edelman *et al.* [14] que preferem usar evidências negativas implícita.

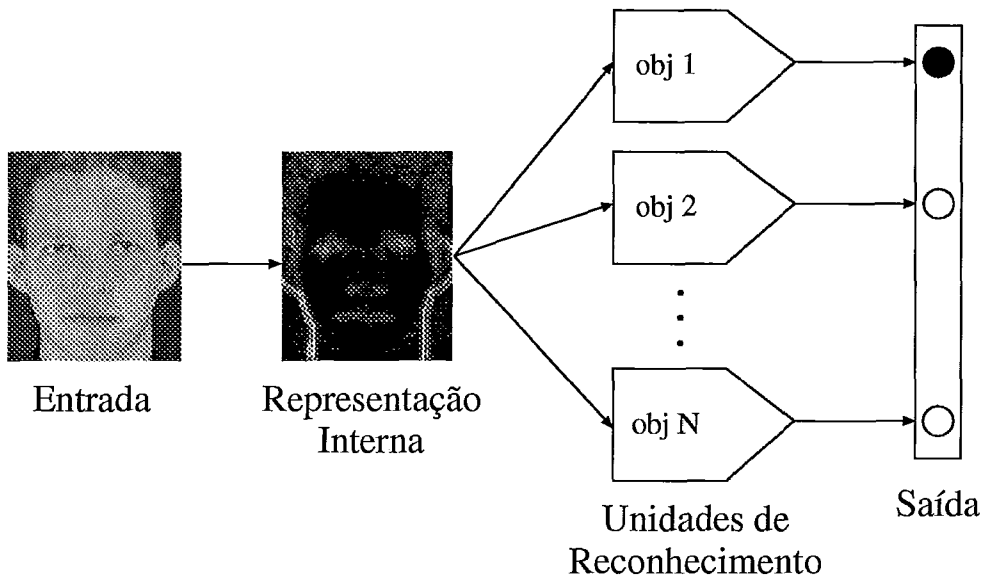


Figura 5.4: **Sistema combinado** utilizando uma RBF para cada objeto. A unidade com a maior saída classifica a entrada.

Para cada objeto, criamos uma unidade de reconhecimento na qual uma rede RBF é treinada para discriminar entre este e outros objetos selecionados na base de dados, Figura 5.5. Ao invés de usar todos os dados disponíveis dos outros objetos, adotamos uma estratégia de selecionar um único contra-exemplo para cada imagem selecionada. Este contra-exemplo foi escolhido como a imagem mais parecida com o exemplo de treinamento, ou seja, aquela que, dentro do conjunto de treinamento de todos os demais objetos, tiver a menor distância euclidiana. Estes dados seriam os mais difíceis de distinguir, por serem os mais ambíguos. Esta estratégia é baseada na hipótese que similaridade leva a confusão, então a inclusão deste tipo de evidências negativas tem o potencial de melhorar a discriminação. A unidade que tiver o maior valor de saída representa a classe que melhor classificou a entrada, sendo a saída final do sistema. Dizemos que o reconhecimento foi correto toda vez que a unidade escolhida estiver representando o mesmo objeto apresentado pela imagem de entrada.

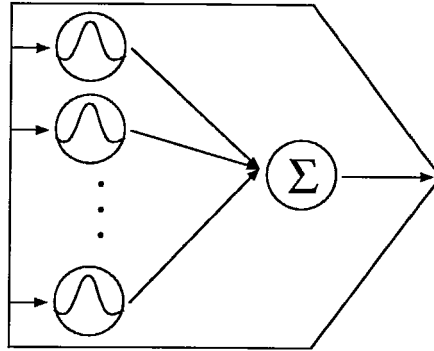


Figura 5.5: **Unidade de reconhecimento** uma rede RBF para cada objeto, cuja saída será comparada com as demais.

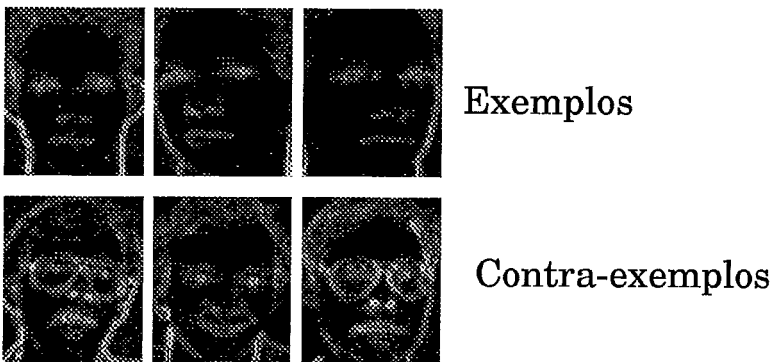


Figura 5.6: **Conjunto de treinamento** de uma unidade de reconhecimento. Para cada imagem do conjunto de treinamento é escolhido um contra-exemplo (imagem mais semelhante ao exemplo). Como pode ser visto, todas as imagens são filtradas conforme a representação interna definida.

5.3.2 Estabelecimento dos parâmetros

O tamanho e a performance da nossa rede RBF é determinada por: (1) tamanho da imagem, como o número de unidades de entrada; (2) o número de exemplos de treinamento (que determina o número de unidades intermediárias) e (3) o número de classes de objetos distintos (que representa o número de unidades de saída). A capacidade de uma rede RBF aproximar funções multi-dimensionais não lineares depende desta ter um número suficiente de unidades intermediárias e de uma distribuição adequada dos centros das funções radiais. Cada exemplo de treinamento é associado a uma unidade intermediária, com o vetor I de entrada definido como seu centro [6, 14].

Como no trabalho de Edelman *et al.* [14], o parâmetro σ das funções Gaussianas

é a largura do campo receptivo. Este parâmetro estabelece o compromisso entre uma performance satisfatória no conjunto de treinamento e a generalização para novas entradas. Seu valor foi estabelecido pelo cálculo da distância média entre os membros do conjunto de treinamento (exemplos positivos e negativos), como mostra a Equação 5.9, onde calculamos cada valor σ para cada uma das H unidades intermediárias pela distância média entre o centro de cada unidade intermediária c_α e todas as outras intermediárias c_h [14, 26]. A escolha deste σ fez com que a saída de cada unidade de reconhecimento, para uma dada pessoa, fosse sempre maior para as imagens (de treinamento) da mesma pessoa do que para imagens das demais pessoas da base de dados.

$$\sigma_\alpha = \frac{1}{H\sqrt{2}} \sum_h \sqrt{[c_\alpha - c_h]^T * [c_\alpha - c_h]} \quad (5.9)$$

5.3.3 Treinamento

O aprendizado supervisionado da rede RBF irá determinar os pesos W_{ih} entre as unidades da camada intermediária e da camada de saída. A saída o para cada unidade de saída i para uma determinada entrada l é dada pela equação:

$$o_i^l = \sum_h W_{ih} \phi_h^l \quad (5.10)$$

Abordamos duas formas de determinar estes pesos, sendo que as duas podem ser vistas como a minimização da medida de erro (função de custo) E da rede:

$$E = \sum_l E^l = \sum_l \sum_i [t_i^l - o_i^l]^2 \quad (5.11)$$

onde t_i^l é o valor de saída esperado para a unidade i para a entrada l . A primeira forma é usando gradiente descendente (pelo uso da regra delta) [25], enquanto a segunda é um método de aprendizado imediato usando Decomposição de Valores (*Singular Value Decomposition*) [54] para calcular uma matriz pseudo-inversa [14,

52], que permite uma solução exata para o cálculo de W_{ih} em uma única iteração. Este método de decomposição pode ser melhor explicado quando representamos as equações de treinamento usando notação de vetores. Sabemos que o erro E da rede (equação 5.11) será zero quando $o_i^l = t_i^l$, ou seja, quando a saída da rede atinge o valor esperado. Neste caso, a equação 5.10 pode ser reescrita como:

$$\sum_h W_{ih} \phi_h^l = t_i^l \quad (5.12)$$

Os valores de W_{ih} podem ser estimados a partir desta equação usando a decomposição SVD para encontrar a pseudo-inversa da matriz formada pelas saídas da camada intermediária para cada exemplo de treinamento [26]. Esta abordagem é mais interessante do que o uso do gradiente descendente pelo estabelecimento imediato dos pesos, qualquer que seja o tamanho da rede. Porém, os erros acumulados para o cálculo da pseudo-inversa da matriz a partir da decomposição SVD acabam por comprometer os resultados das simulações.

5.4 Simulações

Utilizando a base de dados de ORL, apresentada no Apêndice B, estabelecemos conjuntos de 1, 3, 5 e 7 poses para treinamento do sistema, associando um valor de saída esperada (*target*) próximo a 1 (i. e. 0,99). Para cada uma destas imagens é escolhido um anti-exemplo, uma evidência negativa à qual é associada um valor esperado próximo de 0 (i. e. 0,01).

Dentre as 10 poses oferecidas para cada pessoa, foram escolhidas 1, 3, 5 ou 7 imagens para compor o conjunto de evidências positivas e sua respectiva evidência negativa. Foram executados tantos testes quanto o número de combinações de 1, 3, 5 e 7 poses que puderam ser escolhidas. Para cada uma destas combinações, estabelecemos o treinamento de cada unidade de reconhecimento, calculando os parâmetros σ_h e c_h da unidades da camada intermediária. Para cada pose do objeto,

apresentamos o exemplo e o anti-exemplo, calculando todas as saídas da camada intermediária. Estes valores, juntamente com o valor esperado para cada exemplo, são usados para o cálculo do vetor de pesos, aplicando gradiente descendente.

Estabelecidos os pesos, passamos aos testes. Nesta fase, para cada objeto, todas as poses da combinação não utilizadas para o treinamento (mesmo as que tenham sido escolhidas como anti-exemplo para outro objeto) são apresentadas para o sistema. A resposta do sistema, a categoria que mais se assemelha à entrada, é dada pela unidade de reconhecimento com o maior valor de saída. Se esta classe representa o mesmo objeto apresentado na imagem de entrada, a saída é dita correta.

As Tabelas 5.1, 5.2 e 5.3 apresentam os resultados destes testes, apresentando a média das taxas de erro dentre as combinações usando 3, 5 e 10 objetos. Como pode ser notado, a quantidade de objetos envolvida foi um dos parâmetros com maior influência sobre os resultados. Quando temos poucos dados a serem comparados, os melhores resultados estão no nível 2 da pirâmide. Quanto mais informação utilizada pelos testes, fica mais evidente que a redução de dimensão e de resolução da entrada, mesmo no nível 3, não se mostrou suficiente para a correta classificação dos objetos. Um fator que pode estar influenciando nesta piora dos resultados é a escolha da evidência negativa de cada exemplo de treinamento. Quanto maior o número de imagens de treinamento, mais próximos tendem a ficar os exemplos de treinamento. Ao invés de colaborar para destacar os casos mais ambíguos, este método de escolha pode estar causando mais confusão por estarem muito próximos das demais imagens do mesmo objeto.

3 objetos	7 imagens	5 imagens	3 imagens	1 imagem
nível 0	25.83%	37.91%	40.87%	36.30%
nível 1	4.17%	10.71%	21.04%	16.30%
nível 2	0.93%	5.48%	10.71%	24.81%
nível 3	5.65%	13.07%	19.76%	25.56%

Tabela 5.1: **Taxa de erro:** média dos resultados de todas as combinações selecionando apenas 3 objetos da base de dados.

5 objetos	7 imagens	5 imagens	3 imagens	1 imagem
nível 0	33.11%	49.13%	57.93%	69.33%
nível 1	19.22%	24.00%	35.74%	53.111%
nível 2	45.06%	39.54%	44.81%	33.56%
nível 3	12.44%	34.33%	33.00%	28.00%

Tabela 5.2: **Taxa de erro:** média dos resultados de todas as combinações selecionando apenas 5 objetos da base de dados.

10 objetos	7 imagens	5 imagens	3 imagens	1 imagem
nível 0	48.83%	60.66%	71.90%	65.00%
nível 1	86.93%	82.93%	68.29%	53.00%
nível 2	59.67%	60.44%	71.74%	38.78%
nível 3	36.22%	57.89%	57.89%	36.22%

Tabela 5.3: **Taxa de erro:** média dos resultados de todas as combinações selecionando apenas 7 objetos da base de dados.

Capítulo 6

Conclusões

A principal motivação deste trabalho foi tentar emular uma tarefa desempenhada pelo no córtex cerebral: o reconhecimento visual de objetos. O cérebro extrai e armazena a *essência* dos objetos, as informações mais relevantes ao reconhecimento, o que é muito difícil para qualquer sistema artificial. Para tentarmos extrair estas informações, investigamos a capacidade de categorização de informação oferecida por diferentes combinações de informação. Utilizando imagens filtradas de forma a armazenar apenas os contornos gerados pelos contrastes luminosos, estabelecemos uma invariância luminosa. Algumas dessas imagens foram, então, combinadas em modelos que, por sua vez, foram transformados em categorias. Essas combinações foram feitas para tentar diminuir as variações das condições de um objeto nas imagens, facilitando o seu reconhecimento.

As variações de condição abordadas foram: rotação no eixo perpendicular ao plano da imagem (eixo z), mudança de expressão (para as faces humanas) e de iluminação. Aproveitamos para experimentar os resultados utilizando a variação de resolução das imagens em vários níveis de uma pirâmide, investigando como se comporta o reconhecimento para diversos níveis de detalhamento da informação. Modelos menos detalhados podem aceitar variações maiores na informação de entrada. Entretanto, em algum limite, a baixa resolução passa a confundir imagens de diferentes objetos. Além disso, quanto menos detalhes, menor o espaço necessário para o armazenamento da imagem, menor também o custo computacional de seu

processamento.

Os resultados mostraram que a filtragem estabelecendo extração de contornos e a variação de resolução definida pela estrutura de pirâmide, quando combinados com regras de construção de modelos de memória adequadas podem se comportar como meios eficientes de extração das informações mais importantes das imagens na tarefa do reconhecimento. O número de imagens utilizadas para gerar estes modelos também se mostrou um fator importante para a criação de modelos representativos. Quanto mais imagens de faces, mais vezes os mesmos pontos se repetem e melhores devem ser os resultados. Por outro lado, nos objetos rotacionados, quanto maior o número de ângulos representados, mais esparsas e variadas as imagens representadas pelos modelos, piorando o resultado.

O sistema proposto se apresenta bastante insensível a pequenas rotações planares e em profundidade e pequena variação de escala, além das mudanças de expressão, como as transformações encontradas na base de dados ORL (Veja a Figura 6.1 como exemplo). Este bom desempenho ocorre sem que seja empregado nenhum pré-processamento com o intuito de normalizar as imagens, eliminando ou amenizando estas transformações.

Além disto, este método obteve excelente performance para esta base de dados de faces, com resultados comparáveis a outros bons métodos da literatura e, ao mesmo tempo, tendo um custo computacional muito menor. Os mapas de células complexas são construídos por operações de filtragem simples que poderiam ser otimizadas em software ou mesmo hardware (i.e., sendo executadas no domínio de Fourier). O mesmo é verdade para as estruturas de pirâmide. Mas o principal aspecto que leva ao pequeno custo computacional é dado pelo fato de que redes neurais, a princípio, requerem procedimentos de treinamento muito custosos, enquanto que a síntese de protótipos proposta é feita em uma única (e simples) interação. Isto se mostra crítico em bases de dados que se modificam ao longo do tempo, com inclusão e exclusão de

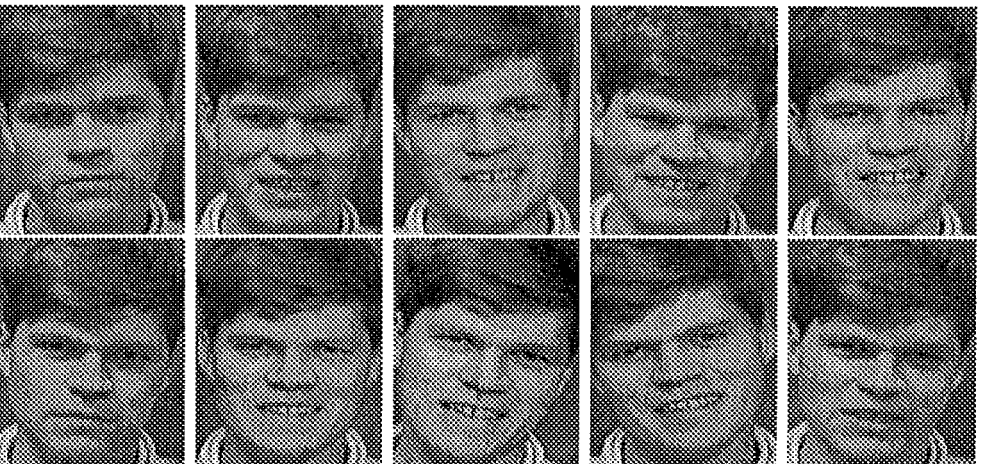


Figura 6.1: Transformações da base ORL: pequenas rotações planares e em profundidade, pequena variação de escala, variação de expressão e acessórios.

elementos. Note que, tanto na combinação pelo mínimo quanto na combinação pela média, um novo membro pode ser adicionado ou eliminado pela simples construção ou exclusão de seu modelo. Outros métodos, incluindo o popular *Eigenfaces* [59] e as redes neurais tradicionais, teriam que calcular novamente todos os modelos como se estivesse sendo apresentado uma base de dados completamente nova.

A proposta de generalização com objetos quaisquer em uma base de dados com grandes rotações em profundidade não obteve resultados tão bons. A base de dados de Columbia incluía a rotação completa, em ângulos de 5 graus, de objetos tão distintos como carros e xícaras, acrescentando a dificuldade de apresentar o mesmo objeto em poses tão distintas como de frente, de perfil e de costas. O primeiro problema seria criar um único modelo para cada objeto, como nos testes anteriores. Isto foi contornado pela criação de diversos modelos intermediários e por uma extensão no sistema proposto, que permitiu combinar os vários modelos do mesmo objeto em uma única saída. Embora o número de elementos para a composição de cada modelo também tenha se mostrado um importante fator na sua representatividade, os testes com esta base de dados demonstraram um comportamento muito suave em relação a variação da resolução da imagem estabelecido pela estrutura de pirâmide. Isto pode ter sido influenciado pela diferença das resoluções iniciais de cada base de dados (o tamanho inicial é quase o dobro) ou porque a variação da resolução de um nível para o outro ainda é insuficiente.

Em todas as simulações, encontramos na informação média a melhor forma de combinação de dados. Um teste qualitativo com uma única combinação cujos resultados são apresentados na Tabela 6.1 utiliza múltiplos modelos. Ou seja, ao invés de combinarmos a informação das categorias em modelos únicos, criamos um modelo para cada uma das imagens. Embora os resultados sejam promissores, este caso vai de encontro a tentativa de combinar informação armazenada na memória e acaba gerando um custo computacional muito maior pois o número de comparações cresce

proporcionalmente ao número de modelos estabelecidos para cada objeto e, como pode ser visto na Tabela 6.2, a melhora é pequena.

múltiplos modelos independentes	7 imagens 120 entradas	5 imagens 200 entradas	3 imagens 280 entradas	1 imagem 360 entradas
nível 0	4.6%	8.7%	14.1%	31.4%
nível 1	3.7%	5.5%	13.2%	27.8%
nível 2	2.1%	4.5%	7.5%	16.4%
nível 3	2.5%	4.5%	7.5%	16.9%

Tabela 6.1: **Taxas de erro para múltiplos modelos independentes.** Resultado de uma única combinação.

modelo único: média	7 imagens 120 entradas	5 imagens 200 entradas	3 imagens 280 entradas	1 imagem 360 entradas
nível 0	6.7%	10.5%	15.4%	31.4%
nível 1	6.7%	10.5%	15.7%	27.8%
nível 2	2.5%	7.0%	6.0%	16.4%
nível 3	6.7%	9.0%	9.3%	16.9%

Tabela 6.2: **Taxas de erro para um único modelo combinado** pela média. Resultado da mesma combinação anterior.

Numa segunda fase, apresentando um estudo aplicando uma rede neural, procurou-se manter a independência entre os modelos. Foram estabelecidas unidades de reconhecimento, cada qual contendo uma rede neural RBF, treinada para reconhecer um dos objetos da base de dados. O ajuste de pesos da primeira camada, entre as unidades de entrada e intermediárias, se dá de forma automática, baseando-se nos dados de treinamento. Por outro lado, a segunda camada, ligando as unidades intermediárias com as de saída, foi treinada pelo método de gradiente descendente. Existe aqui uma discussão de tentar substituir este treinamento por algum método automático, como a sugestão do uso da inversão de matrizes utilizando Decomposição de Valores (SVD).

Em qualquer rede neural, a qualidade do treinamento é sempre dependente da relação entre a dimensão de entrada e o número de exemplos utilizados. Numa

rede neural RBF [24, 25], entretanto oferece meios para se utilizar poucos exemplos de treinamento, mesmo com imagens (dados dimensão notadamente alta). Para reduzir a dimensão da entrada, trabalhos como o de Edelman *et al.* [14] e Tomaz *et al.* [58] apresentam métodos de redução de dimensão bastante diferentes, obtendo bons resultados. Porém, a redução oferecida pelos níveis mais baixos da pirâmide contruída ainda não se mostrou suficiente para a obtenção de bons resultados. Uma idéia a ser implementada é uma análise com níveis ainda mais baixos. Para que a variação da resolução não seja tão grande, uma solução seria a escolha de uma razão de crescimento entre os filtros menor do que a utilizada neste sistema.

Trabalhos futuros

Uma idéia derivada dos trabalhos apresentados em [47, 17, 48, 26], é implementar uma taxa de discriminação. Da forma que o sistema foi desenvolvido, a resposta para o reconhecimento é sempre a categoria que estiver mais próxima a imagem de entrada. Tanto resultados corretos quanto incorretos, podem ser questionados quando verificamos o segundo melhor resultado da comparação. Se este estiver muito próximo do primeiro, podemos dizer que houve empate, invalidando o resultado ao alegar que o sistema não foi capaz de escolher entre as duas melhores respostas.

Além de explorar outras formas de combinação de imagens, dependendo da base de dados envolvida, talvez seja necessário investigar outros níveis da pirâmide, ou mesmo outras razões de crescimento para os filtros, gerando níveis de resolução intermediários, na busca da resolução mais adequada para diversas bases de dados.

Para validar os resultados obtidos com a base de dados de Columbia, é o estabelecimento de várias combinações de conjuntos de treinamento e teste. Além disto, modelos sintetizados pela média ou pelo mínimo, representando uma combinação das imagens mais próximas, poderiam ser usados como entrada de uma rede neural RBF em substituição a um subconjunto da base de dados.

Apêndice

Implementação do Pré-processamento

A seguir, formalizamos os três estágios computacionais envolvidos no cálculo do mapa de resposta das células complexas. Para mais detalhes veja [23, 49]. O código de todos os sistemas desenvolvidos nesta tese foram desenvolvidos em C e executados em uma Estação de Trabalho UltraSparc Sun com sistema operacional Unix Solaris.

Estágio I: Células Concêntricas

A entrada inicial da distribuição de luminância L da imagem é processada por filtros sensíveis à variação do contraste, porém independentes da direção desta variação, ou seja, isotrópicos. A saída deste estágio é dada por:

$$R = \left[\frac{\beta E - \gamma I}{\alpha + E + I} \right]^+ \quad (.1)$$

onde $[x]^+ = \max(x, 0)$, $\beta = 0.5$, $\gamma = 0.5$, e $\alpha = 10.0$. Note que as respostas são calculadas para cada posição (i, j) de toda a imagem embora estes índices tenham omitidos (para simplificação).

As contribuições excitatória (E) e inibitória (I) são dadas por versões processadas da imagem de entrada L com filtros de passa-baixa (funções Gaussianas) utilizando uma pequena constante, σ_c , para definir o centro e outra maior, σ_s , para definir a vizinhança:

$$E = L \otimes G_c \quad \text{e} \quad I = L \otimes G_s \quad (.2)$$

onde as funções de peso do centro (G_c) e da vizinhança (G_s) são Gaussianas normalizadas na forma:

$$G = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-i)^2 + (y-j)^2}{2\sigma^2}\right) \quad (.3)$$

tal que (i, j) é a posição onde o filtro está centralizado na imagem; $\sigma = 0.5$ para o centro e $\sigma = 2.0$ para a vizinhança. O denominador na Equação .1 atua como a normalização do resposta ao contraste, criando maior insensibilidade à intensidade luminosa total.

Estágio II: Células Simples

A saída do primeiro estágio é processada por filtros alongados unisotrópicos sensíveis à orientação do contraste, tal como os modelos das células simples do córtex visual dos primatas. As respostas a transições claro-escuro (ce) e escuro-claro(ec) são geradas para um total de oito orientações k :

$$s_k^{ld} = [R \otimes S_k^{ld}]^+ \quad \text{e} \quad s_k^{dl} = [R \otimes S_k^{dl}]^+ \quad (.4)$$

onde os filtros S são calculados pela diferença de duas equações Gaussianas separadas por um *off-set* da forma:

$$G = \frac{1}{\sigma_x\sigma_y\sqrt{2\pi}} \exp\left(-\frac{(x-i)^2}{\sigma_x^2} + \frac{(y-j)^2}{\sigma_y^2}\right) \quad (.5)$$

sendo $\sigma_x = 0.5$ e $\sigma_y = 1.5$. Filtros em outras orientações são obtidos pela aplicação de uma matriz de rotação adequada para cada ângulo.

Estágio III: Células Complexas

Respostas insensíveis à direção da transição do contraste, como as produzidas pelas células complexas do córtex visual dos primatas são geradas pela combinação das respostas das células simples de mesma direção, neste caso as células *ce* e *ec* em todos os pontos. Ao final para gerar o *mapa de respostas das células complexas* sensível ao contorno em todas as orientações fazemos uma combinação de todas as orientações:

$$C = \sum_k s_k^{ld} + \sum_k s_k^{dl}. \quad (.6)$$

Representação em Pirâmides

O primeiro Estágio de pré-processamento foi aplicado em uma única escala espacial. Os cálculos efetuados pelos demais Estágios, especificados acima, são replicados em quatro diferentes tamanhos de escala espacial. Isto é feito dobrando o desvio padrão (σ) das equações Gaussianas dos filtros alongados a cada nível.

Depois de gerar cada mapa de respostas das células complexas, uma para cada escala espacial, suas dimensões são reduzidas construindo a estrutura de pirâmide [1, 8, 9], como pode ser acompanhado na seção 2.5 e pelo esquema da Figura 2.18. O primeiro nível da pirâmide – nível 0 – não sofre redução de dimensão (mantendo o mesmo tamanho da imagem original), enquanto os níveis 1 a 3 sofrem redução de dimensão por fatores 2, 4 e 8, respectivamente, em ambos os eixos. Com isto, a partir de cada imagem, contruímos uma pirâmide de quatro níveis.

Deve ser lembrado que todos os testes foram implementados para todos os quatro níveis das pirâmides de imagens.

Referências Bibliográficas

- [1] Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., & Odgen, J.M. (1984). Pyramid methods in image processing. *RCA Engineering*, Nov/Dec, 33–41.
- [2] Andersen, C.H. & Van Essen, D.C. (1987). Shift Circuits: a computational strategy for dynamic aspects of visual processing. *Proceedings of the National Academy of Science*, USA, **84**, 6297–6301.
- [3] Beymer, D.J. (1994). Face recognition under varying pose. *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, 756–761, Seattle, USA.
- [4] Beymer, D.J. & Poggio, T. (1995). Face recognition from one example view. *Proceedings of International Conference on Computer Vision*, 500-507, Cambridge, USA.
- [5] Brunelli, R. & Poggio, T. (1991). HiperBF networks for real object recognition. In Myopoulos, J. & Reiter, R (Eds), *Proceedings of International Joint Conference on Artificial Intelligence*, Sydney, Australia. Morgan Kaufmann, 1278–1284.
- [6] Brunelli, R. & Poggio, T. (1993). Face recognition: Features versus Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15** (10), 1042–1052.
- [7] Bülthoff, H.H., & Edelman, S. (1992). Psychophysical support for a 2-D view

- interpolation theory of object recognition. *Proceedings of the National Academy of Sciences, USA*, **89**, 60–64.
- [8] Burt, P.J. (1988). Smart Sensing within a Pyramid Vision Machine. *Proceedings IEEE*, **76**, #8, 1006–1015.
- [9] Burt, P.J., & Adelson, E.H. (1983). The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, **COM-31**, # 4.
- [10] Carpenter, G.A. (1994). Fuzzy ART. In B. Kosko (Ed.), *Fuzzy Engineering*. Carmel, IN: Prentice Hall. (Also appears as Technical Report CAS/CNS-TR-93-059, Department of Cognitive and Neural Systems, Boston University).
- [11] Carpenter, G.A., Grossberg, S., & Rosen, D.B. (1991). Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System. *Neural Networks*, **4**, 759–771.
- [12] Desimone, R. (1991). Face-selective cells in the temporal cortex of monkeys. *Journal of Cognitive Neuroscience*, **3**, # 1.
- [13] Edelman, S. (1997). Computational theories of object recognition. *Trends in Cognitive Science*, **1**, # 8, 296–304.
- [14] Edelman, S., Reisfield, D., & Yeshurun, Y. (1992). Learning to recognize faces from examples. *Proceedings of the 2nd European Conference on Computer Vision*, Santa Margherita Ligure, Italy, 787–791.
- [15] Edelman, S., Intrator, N., & Poggio, T. (1997). Complex Cells and Object Recognition. <http://barus.physics.brown.edu/people/nin/research.html>.
- [16] Exel, S. (1999). Reconhecimento Visual Atencional. Tese de Doutorado – COPPE/UFRJ, Engenharia de Sistemas de Computação.

- [17] Exel, S., & Pessoa, L. (1998). Attentive visual recognition. *Proceedings of the International Conference on Pattern Recognition (ICPR'98)*, August 16th–20th, Brisbane, Australia, 690–692.
- [18] Duvdevani-Bar, S., & Edelman, S. (1997) Visual recognition and categorization on the basis of similarities to multiple class prototypes. *AI Memo 1615*, Massachusetts Institute of Technology, Cambridge, MA, September 1997.
- [19] Girosi F. (1992). Some extensions of radial basis functions and their application in artificial intelligence. *Computers & Mathematics with Applications*, **24**, #12, 61–80.
- [20] Girosi, F., Jones, M., & Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation*, **7**, 219–269.
- [21] Goldstein, E.B. (1989). *Sensation and Perception*. Books/Cole Publishing Company, Pacific Grove, California.
- [22] Grossberg, S. (1987). Competitive Learning: From interactive activation to adaptative resonance. *Cognitive Science*, **11**, 23–63.
- [23] Grossberg, S., & Pessoa, L. (1998). Texture segregation, surface representation, and figure-ground separation. *Vision Research*, **38**, 2657-2684.
- [24] Hassoun, M. (1995). *Fundamentals of Artificial Neural Networks*, MIT Press, Cambridge, MA, USA.
- [25] Haykin, S., (1994). *Neural Networks*, Macmillan College Publishing Company, New York, USA.
- [26] Howell, A.J. (1997). Automatic Face Recognition using Radial Basis Function Networks. *Cognitive Science Research Papers*, University of Sussex, Brighton.

- [27] Howell, A.J., & Buxton, H. (1995). Invariance in Radial Basis Function Neural Networks in Human Face Classification. *Proceedings of the International Workshop on Face & Gesture Recognition*, Zurich, Switzerland, 221–226.
- [28] Hubel, D.H. & Wiesel, T.N. (1963). Receptive fields of cells in striate cortex of very young visually inexperienced kittens. *Journal of Neurophysiology*, **26**, 994–1002.
- [29] Hubel, D.H. & Wiesel, T.N. (1977). Ferrier lecture: functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society of London*, **198**, 1–59.
- [30] Hummel, J.E. & Biederman, I. (1992). Dynamic binding in a neural network for shape recognition. *Psychol. Rev.* **99**, 480–517.
- [31] Intrator, N., Reisfield, D., & Yeshurun, Y. (1996). Face Recognition using a Hybrid Supervised/Unsupervised Neural Network. *Pattern Recognition Letters*, **17**, 67–76.
- [32] Kandel, E.R., Schwartz, J.H. & Jessell (1995). *Essentials of Neuroscience and Behavior*, Appleton & Lange, Norwalk, CT, USA.
- [33] Kohonen, T. (1989). *Self-Organization and Associative Memory*, Springer-Verlag, Berlin, Germany.
- [34] Lawrence, S., Giles, C.L., Tsoi, A.C., & Back, A.D. (1997). Face Recognition: A Convolutional Neural Approach. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks and Pattern Recognition*, **8**, num. 1, 98–113.
- [35] Leitão, A.P. & Pessoa, L. (1999). Pyramid Representation for Face Recognition. *Proceedings of the III Workshop on Cybernetic Vision*, February 23rd–26th, Campinas, Brazil.

- [36] Marr, D. (1982). *Vision*. New York: Freeman & Company.
- [37] McIlwain, J.T. (1996). *An Introduction to the Biology of Vision*. Cambridge University Press, Cambridge. UK.
- [38] Mel, B. (1997). Combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation* **9**, 777–804. (VERIFICAR METODO DE CLASSIFICACAO)
- [39] Mel, B. & Fiser, J. (1998) Seeing with Spatially-Invariant Receptive Fields: When the ‘Binding Problem’ Isn’t. *Proceedings of the 5th Joint Symposium in Neural Computation*, UCSD.
- [40] Millward, R. & O’Toole, A. (1986). Recognition memory transfer between spatial-frequency analysed faces. In Ellis, H.d., Jeeves, M.A. Newcombe, F. & Young, A.W. (Eds), *Aspects of Face Processing*, 34–44, Nijhoff, Dordrecht, The Netherlands.
- [41] Nalwa, V.S. (1993) *A Guided Tour of Computer Vision*. Addison-Wesley Publishing Company, AT&T, USA.
- [42] Oram, M.W., & Perret, D.I. (1992). Time course of neural response discriminating different views of the face and the head. *Journal of Neurophysiology*, **68**, 70–84.
- [43] Oram, M.W., & Perret, D.I. (1994). Modeling visual recognition from neurobiological constraints. *Neural Networks*, **7**, #6/7, 945–972.
- [44] Orr, M.J.L. (1996). Introduction to Radial Basis Function Networks. <http://www.cns.ed.ac.uk/people/mark/intro/intro.html>
- [45] O’Toole, A.J., Edelman, S., & Bühlhoff, H.H. (1998). Stimulus-specific effects in face recognition over changes in viewpoint. *Vision Research*, **38**, 2351–2363.

- [46] Pessoa, L. (1990). Aprendizado Não-Supervisionado em Redes Neurais. Tese de Mestrado – COPPE/UFRJ, Engenharia de Sistemas de Computação.
- [47] Pessoa, L., Exel, S., Roque, A., & Leitão, A.P. (1998). Active Scene Exploration Vision System. *Proceedings of the International Conference on Neural Networks and Brain (ICNN&B'98)*, October 27th–30th, Beijing, China, 543–546.
- [48] Pessoa, L., Exel, S., Roque, A., & Leitão, A.P. (1998). Attentive visual recognition for scene exploration. *Proceedings of the International Conference on Neural Information Processing (ICONIP'98)*, October 21st–23rd, Kitakyushu, Japan, **3**, 1291–1294.
- [49] Pessoa, L., Mingolla, E., & Neumann, H. (1995). A contrast- and luminance-driven multiscale network model of brightness perception. *Vision Research*, **35**, 2201–2223.
- [50] Perret, Rolls & Caan (1982). Visual neurones responsive to faces in the monkey temporal cortex. *Experimental Brain Research*, **47**, 329–342.
- [51] Poggio, T., & Edelman, S. (1990). A network that learns to recognize three-dimensional objects. *Nature*, **343**, 263–266.
- [52] Poggio, T., & Girosi, F. (1990). Networks for approximation and learning. *Proceedings of IEEE*, **78**, 1481–1497.
- [53] Poggio, T., & Girosi, F. (1990). Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, **247**, 978–982.
- [54] Press, W.H., Flannery, B.P., Teukolsky, S.A., & Vetterling, W.T. (1986). *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK.
- [55] Rosenblum, M., Yacoob, Y., & Davis, L. (1994). Human Emotion Recognition from Motion Using a Radial Basis Function Network Architecture. *Presented at*

the IEEE Workshop on Motion of Non-Rigid and Articulated Objects, Austin, TX, November 1994.

- [56] Russell, J.S., & Norvig P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall.
- [57] Sim, T., Sukthankar, R., Mullin, M., & Baluja, S. (1999). High-Performance Memory-based Face Recognition for Visitor Identification. *JPRC-Technical Report-1999-01*. <http://www.cs.cmu.edu/baluja/techreps.html>.
- [58] Tomaz, C.E., Feitosa, R.Q., & Veiga, A. (1998). Design of Radial Basis Function Network as Classifier in Face REcognition Using Eigenfaces. *Proceedings of the Vth Brazilian Symposium on Neural Network*, December 9th – 11th, Belo Horizonte, Brazil, **1**, 118–123.
- [59] Turk, M. & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, **3**, 71–86.
- [60] Ullman, S. (1995). *High-level vision: Object Recognition and Visual Cognition*, MIT Press, Cambridge, MA.
- [61] Ullman, S., & Basri, R. (1991). Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 992–1006.
- [62] Valentin, D., Abdi, H., O’Toole, A.J. & Cottrell, G.W. (1994). Connectionist models of face processing: A survey. *Pattern Recognition*, **27**, 1208–1230.
- [63] Wiskott, L., Fellous, J., Krüger, N., & von der Malsburg, C. (1997). Recognizing faces by dynamic link matching. *Proceedings of the ICANN’95*, Paris, 347–352. (Also appears in *Neuroscience*, **4**, #3, s14–118, 1996).

- [64] Wiskott, L., Fellous, J., Krüger, N., & von der Malsburg, C. (1997). Face recognition by elastic graph matching. *7th International Conference on computer Analysis of Images and Patterns*, Kiel, Germany, September.
- [65] Yow, K.C., & Cipolla, R. (1997). Feature-based human face detection. *Image and Vision Computing*, **15**, 713–735.
- [66] Zeki, S. (1993). *A vision of the brain*. Blackwell Scientific Publications.