Projeto multirresolução de operadores morfológicos a partir de exemplos

Daniel André Vaquero

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO DE MESTRE
EM
CIÊNCIAS

Área de Concentração : **Ciência da Computação**
Orientador : **Prof. Dr. Junior Barrera**

O autor recebeu apoio financeiro da FAPESP, sob o processo
número 03/10066-7, para a elaboração deste trabalho.

- São Paulo, maio de 2006 -

Projeto multirresolução de operadores morfológicos a partir de exemplos

Este exemplar corresponde à redação
final da dissertação devidamente corrigida
e defendida por Daniel André Vaquero
e aprovada pela Comissão Julgadora.

São Paulo, 08 de maio de 2006.

Banca Examinadora :

Prof. Dr. Junior Barrera (orientador) – IME-USP

Prof. Dr. Roberto Marcondes Cesar Junior – IME-USP

Prof. Dr. Alexandre Xavier Falcão – IC-UNICAMP

aos meus pais

Agradecimentos

Aos meus pais, por todo o apoio, carinho e incentivo.

Ao professor Junior Barrera, pela amizade, paciência, encorajamento e por todas as oportunidades que me proporcionou nos últimos anos; aprendi muito com a sua excelente orientação, não somente em aspectos técnicos mas também sobre o processo científico em geral.

Ao professor Roberto Hirata Jr., pela intensa colaboração como co-orientador deste trabalho. Agradeço pelo incansável acompanhamento, assim como pelas incontáveis conversas e pelas inúmeras dicas sobre os mais variados assuntos. Tenho certeza de que encontrei um grande amigo.

Aos professores Ronaldo Fumio Hashimoto, Roberto Marcondes Cesar Jr. e Alexandre Xavier Falcão, pelos comentários e sugestões durante o exame de qualificação e a defesa, que contribuíram muito para a qualidade deste trabalho.

Aos amigos David da Silva Pires, Roberto Hirata Jr., Roberto Marcondes Cesar Jr. e Jesús Pascual Mena-Chalco, por tudo o que aprendi durante o tempo em que trabalhamos juntos na administração da rede do laboratório.

Aos membros do laboratório de processamento de imagens do IME-USP, pela receptividade e por proporcionar um ambiente estimulante para se fazer pesquisa. Aprendi bastante com todas as conversas e discussões.

A todos os meus amigos, especialmente à turma do BCC da época da graduação. Em particular, gostaria de agradecer aos que estiveram mais próximos durante os últimos anos: Daniel Cordeiro, Daniel Martin, Fernando Mário, Giuliano Mega, João Soares, Marcos Broinizi e Pedro Takecian.

À FAPESP, pelo apoio financeiro durante o período de mestrado.

Por fim, a todas as outras pessoas que direta ou indiretamente contribuíram para o desenvolvimento deste trabalho.

Muito obrigado!

Resumo

Resolver um problema de processamento de imagens pode ser uma tarefa bastante complexa. Em geral, isto depende de diversos fatores, como o conhecimento, experiência e intuição de um especialista, e o conhecimento do domínio da aplicação em questão. Motivados por tal complexidade, alguns grupos de pesquisa têm trabalhado na criação de técnicas para projetar operadores de imagens automaticamente, a partir de uma coleção de exemplos de entrada e saída do operador desejado.

A abordagem multirresolução [1] tem sido empregada com sucesso no projeto estatístico de W-operadores de janelas grandes. Esta metodologia usa uma estrutura piramidal de janelas para auxiliar na estimação das distribuições de probabilidade condicional para padrões não observados no conjunto de treinamento. No entanto, a qualidade do operador projetado depende diretamente da pirâmide escolhida. Tal escolha é feita pelo projetista a partir de sua intuição e de seu conhecimento prévio sobre o problema.

Neste trabalho, investigamos o uso da entropia condicional como um critério para determinar automaticamente uma boa pirâmide a ser usada no projeto do W-operador. Para isto, desenvolvemos uma técnica que utiliza o arcabouço piramidal multirresolução como um modelo na estimação da distribuição conjunta de probabilidades. Experimentos com o problema de reconhecimento de dígitos manuscritos foram realizados para avaliar o desempenho do método. Utilizamos duas bases de dados diferentes, com bons resultados. Além disso, outra contribuição deste trabalho foi a experimentação com mapeamentos de resolução da teoria de pirâmides de imagens no contexto do projeto de W-operadores multirresolução.

Abstract

The task of finding a good solution for an image processing problem is often very complex. It usually depends on the knowledge, experience and intuition of an image processing specialist. This complexity has served as a motivation for some research groups to create techniques for automatically designing image operators based on a collection of input and output examples of a desired operator.

The multiresolution approach [1] has been successfully used to statistically design W-operators for large windows. However, the success of this method directly depends on the adequate choice of a pyramidal window structure, which is used to aid in the estimation of the conditional probability distributions for patterns that do not appear in the training set. The choice is made by the designer, based on his intuition and previous knowledge of the problem domain.

In this work, we investigate the use of the conditional entropy criterion for automatically determining a good pyramid. In order to compute the entropy, we have developed a technique that uses the multiresolution pyramidal framework as a model in the estimation of the joint probability distribution. The performance of the method is evaluated on the problem of handwritten digits recognition. Two different databases are used, with good practical results. Another important contribution of this work is the experimentation with resolution mappings from image pyramids theory in the context of multiresolution W-operator design.

Índice

1	Introdução	1
1.1	Objetivos	2
1.2	Contribuições	3
1.3	Organização da Dissertação	4
2	Projeto de Operadores Morfológicos	5
2.1	Técnicas de Projeto	5
2.2	W-operadores	6
2.3	Aprendizado do Operador	7
2.3.1	Um Exemplo Ilustrativo	9
2.4	Problemas com Janelas Grandes	12
2.5	Projeto Utilizando Restrições	12
3	Projeto Piramidal Multirresolução	15
3.1	Restrição de Resolução	15
3.2	Operadores Multirresolução	19
3.3	Pirâmides de Imagens	21
3.3.1	Nota Sobre os Operadores de Adição e Subtração	24
3.3.2	Uso no Projeto de Operadores Multirresolução	24
3.4	Comentários	25
4	Estimação da Distribuição Conjunta	27
4.1	Algoritmo	27
4.2	Exemplo	30
4.3	Algumas Considerações	35

4.4	Representação	36
5	Escolha da Pirâmide	39
5.1	Entropia	40
5.2	Informação Mútua	41
5.3	Cr�terio de Escolha	41
5.4	C�culo da Entropia Condicional	41
5.4.1	Um Exemplo	42
6	Implementa��es	45
6.1	Especifica��o do Sistema	45
6.2	Coleta de Exemplos	46
6.3	Estima��o da Distribui��o Conjunta	46
6.4	C�culo da Entropia Condicional	47
6.5	Decis�o	47
6.6	Aplica��o	47
7	Resultados Experimentais	49
7.1	Reconhecimento de D�gitos Manuscritos	49
7.2	Experimentos com a Nossa Base de Dados	53
7.3	Experimentos com a Base de Dados MNIST	60
7.4	Experimentos com Mapeamentos da Teoria de Pir�mides de Imagens	64
7.5	Experimentos com Tamanhos Variados de Amostras	65
7.6	An�lise	67
7.6.1	Uso da Entropia Condicional na Escolha da Pir�mide	67
7.6.2	Outras Considera��es	68
8	Conclus�o	71
A	Pir�mides de Imagens	75
A.1	Fundamentos e Not��o	75
A.2	Exemplos	77
A.2.1	Pir�mide Usando Amostragem Di�dica Simples	77
A.2.2	Pir�mide de Burt-Adelson	77

A.2.3	Pirâmide de Toet	80
A.2.4	Pirâmide da Mediana	80
A.2.5	Pirâmide de Haar	81
A.2.6	Pirâmide de Haar Morfológica	82
A.2.7	Pirâmide de Heijmans-Toet	82
A.2.8	Pirâmide de Sun-Maragos	83
A.2.9	Pirâmide de Aberturas ou Fechamentos	83
A.2.10	Pirâmide Baseada em Amostragem “Quincunx”	86
A.2.11	Pirâmide de Filtros Alternados Seqüenciais	87
A.2.12	Pirâmide de Difusão Anisotrópica	89
A.2.13	Pirâmide da Diferença	89
A.2.14	Pirâmides com Quantização	90
A.3	Aplicações	91
A.3.1	Compressão e Codificação de Imagens	91
A.3.2	Transmissão Progressiva	92
A.3.3	Localização de Objetos	93
A.4	Algumas Considerações	94
B	Publicação Relacionada ao Tema da Dissertação	95
C	Lista de Símbolos	97
	Referências	99

Lista de Figuras

2.1	Coleta de exemplos: a) janela (com indicação do centro); b) imagem de entrada; c) imagem de saída; d) exemplo coletado.	10
2.2	Aplicação do operador: a) imagem de entrada; b) imagem de saída após algumas iterações do procedimento de aplicação.	11
2.3	Resultado da aplicação.	11
3.1	Exemplo de subamostragem.	16
3.2	Exemplo de partição.	17
3.3	Correspondência entre os operadores.	17
3.4	Exemplo de particionamento – subamostragem.	18
3.5	Outro exemplo de particionamento.	19
3.6	Pirâmide de janelas.	20
3.7	Estrutura piramidal.	21
3.8	a) transformação pirâmide; b) transformação pirâmide inversa.	23
3.9	Pirâmide válida para mapeamentos dependentes de uma vizinhança 3×3 . As janelas são compostas pelos pontos escuros.	24
4.1	Partições aninhadas.	31
4.2	Partição $\mathcal{X}^0$	31
4.3	Partição $\mathcal{X}$ - após o processamento da primeira resolução.	32
4.4	Partição $\mathcal{X}^1$	33
4.5	Partição $\mathcal{X}$ - após o processamento da segunda resolução.	34
4.6	Partição $\mathcal{X}^2$	34
4.7	Partição $\mathcal{X}$ - fim do algoritmo.	35
5.1	Entropia: a) alta entropia; b) baixa entropia.	40

6.1	Estrutura básica do sistema.	46
7.1	Classificação de dígitos: a) dígito a ser classificado; b) resultado da aplicação do W -operador (cada cor representa um valor de saída diferente); c) dígito classificado.	50
7.2	Outra abordagem: a) aproximação C do fecho convexo; b) resultado da aplicação do W -operador restrito à região C (cada cor representa um valor de saída diferente); c) dígito classificado.	50
7.3	Pirâmides de imagens usadas nos experimentos.	51
7.4	Pirâmides de imagens usadas nos experimentos.	52
7.5	Janelas da pirâmide 12.	52
7.6	Exemplos de dígitos manuscritos (resolução reduzida).	54
7.7	Segmentação dos dígitos: a) dígitos originais; b) dilatação por um disco de raio 5; c) componentes conexos da imagem dilatada; d) dígitos segmentados.	55
7.8	Normalização de espessura: a) dígitos normalizados pelo tamanho; b) abertura por um disco de raio 3 (em cinza) sobreposta aos dígitos originais; c) dilatação dos 3 dígitos superiores por um disco de raio 3; d) subamostragem da imagem (a) por um fator de 4; e) subamostragem da imagem (c) por um fator de 4. As imagens (a), (b) e (c) estão representadas em escala reduzida.	58
A.1	Funções equivalentes às máscaras.	78
A.2	Pirâmide de Burt-Adelson. A escala foi reduzida para facilitar a visualização, e as imagens de detalhe foram mapeadas para o intervalo $[0, 255]$	79
A.3	Pirâmide da mediana. A escala foi reduzida para facilitar a visualização, e as imagens de detalhe foram mapeadas para o intervalo $[0, 255]$	81
A.4	Pirâmide de Heijmans-Toet. A escala foi reduzida para facilitar a visualização.	84
A.5	Pirâmide de Sun-Maragos. A escala foi reduzida para facilitar a visualização.	85
A.6	Relações de equivalência. Os discos maiores representam os pontos selecionados pela amostragem <i>quincunx</i>	86
A.7	Pirâmide usando amostragem <i>quincunx</i> . Foi feita mudança de escala em algumas imagens para facilitar a visualização.	88
A.8	Transmissão progressiva. A imagem “entra em foco” gradativamente.	93

Lista de Tabelas

2.1	Tabela de exemplos.	10
2.2	Estimativas das distribuições de probabilidade condicional.	10
2.3	Operador projetado.	11
4.1	Tabela de exemplos - resolução de W_0	31
4.2	Tabela de exemplos - resolução de W_1	33
4.3	Tabela de exemplos - resolução de W_2	34
7.1	Entropias condicionais - Nosso BD.	55
7.2	Erros de classificação - Nosso BD - Máxima Verossimilhança.	56
7.3	Erros de classificação - Nosso BD - Moda.	56
7.4	Erros de classificação - Nosso BD - Abordagem Anterior.	57
7.5	Entropias condicionais - Nosso BD (normalizado).	58
7.6	Erros de classificação - Nosso BD (normalizado) - Máxima Verossimilhança.	59
7.7	Erros de classificação - Nosso BD (normalizado) - Moda.	59
7.8	Erros de classificação - Nosso BD (normalizado) - Abordagem Anterior.	60
7.9	Entropias condicionais - MNIST.	61
7.10	Erros de classificação - MNIST - Máxima Verossimilhança.	61
7.11	Erros de classificação - MNIST - Moda.	62
7.12	Erros de classificação - MNIST - Abordagem Anterior.	62
7.13	Comparação de algoritmos sobre a base de dados MNIST.	63
7.14	Taxas de erro por dígito, para o classificador projetado a partir da pirâmide 14 no experimento da Tabela 7.12.	64
7.15	Entropias condicionais - MNIST (Esqueletos).	65
7.16	Erros de classificação - MNIST (Esqueletos) - Máxima Verossimilhança.	65

7.17 Erros de classificação - MNIST (Esqueletos) - Moda.	65
7.18 Erros (em %) para números diferentes de amostras de treinamento - MNIST - Máxima Verossimilhança.	66
7.19 Erros (em %) para números diferentes de amostras de treinamento - MNIST - Moda. . . .	66

Capítulo 1

Introdução

Resolver um problema de processamento de imagens pode ser uma tarefa bastante complexa. Em geral, isto depende de diversos fatores, como o conhecimento, experiência e intuição de um especialista, e o conhecimento do domínio da aplicação em questão. Motivados por tal complexidade, alguns grupos de pesquisa têm trabalhado na criação de técnicas para projetar operadores de imagens automaticamente, através de otimização estatística a partir de uma coleção de exemplos de entrada e saída do operador desejado. Neste contexto, podemos citar alguns trabalhos anteriores que apresentam métodos para projetar operadores binários [2], classificadores de imagens binárias [3] e operadores em níveis de cinza [4] para resolver problemas de processamento de imagens.

No projeto estatístico de operadores morfológicos de janelas grandes¹, um dos problemas chave é a estimação das distribuições de probabilidade condicional. Do ponto de vista estatístico, este problema é difícil porque a quantidade de exemplos necessários para se obter uma boa estimação é muito grande. Do ponto de vista computacional, os requisitos de memória e tempo de processamento podem ser tão elevados de modo que o projeto do operador torna-se inviável. Recentemente, algumas abordagens propuseram o uso de restrições no espaço de busca do operador, com o objetivo de melhorar a precisão do operador projetado. Entre elas, podemos citar a busca por operadores crescentes [5] e o projeto de operadores multirresolução [1, 6]. Este último tem sido utilizado com sucesso no projeto de operadores de janelas grandes, e é o assunto do nosso trabalho.

O projeto multirresolução baseia-se na restrição de resolução, que parte do princípio de que projetar um operador observando as imagens em uma resolução maior em geral é melhor do que projetar um operador observando as imagens em uma resolução menor; porém, a utilização de uma janela maior implica em um aumento no número de variáveis aleatórias observadas sob a janela, conseqüentemente levando a um aumento no erro de estimação do operador ótimo. Esta metodologia usa uma estrutura piramidal de janelas para observar as imagens em diferentes níveis de resolução, e realiza a estimação

¹O conceito de janela grande depende de diversos fatores, como o número de pares de imagens de entrada e saída disponíveis para projetar o operador, a quantidade de memória disponível, e a velocidade do processador.

das distribuições condicionais utilizando dados observados em resoluções menores quando não há dados suficientes nas amostras para se obter uma boa estimativa em uma resolução maior.

No entanto, a qualidade do operador projetado depende diretamente da escolha de uma estrutura piramidal adequada, que determina a maneira como a redução de resolução é feita. No atual estado da arte, não existe uma técnica que permita obter de forma automática uma boa pirâmide a ser usada no projeto de um operador. Esta escolha tem sido feita pelo projetista a partir de sua intuição e de seu conhecimento prévio sobre o problema a ser resolvido. Isto serve como motivação para o nosso trabalho, que, em prosseguimento à pesquisa desenvolvida em trabalhos anteriores [7, 8], investiga um critério para determinar automaticamente uma boa pirâmide, a partir de um conjunto de pirâmides candidatas e uma coleção de pares de imagens de entrada e saída.

1.1 Objetivos

O principal objetivo deste trabalho é o desenvolvimento de uma metodologia de projeto de operadores morfológicos baseada no projeto multirresolução, mas que auxilie, através de alguma técnica automática, o projetista na escolha da pirâmide utilizada. Temos também como objetivos a experimentação com mapeamentos de resolução da teoria de pirâmides de imagens [9] no contexto de projeto de operadores, e o uso de um problema real de análise de imagens para validar a técnica criada.

Neste sentido, abordamos o problema de projeto estatístico de operadores como um problema composto por duas etapas sequenciais: a estimação das distribuições de probabilidade condicional e a escolha do operador ótimo, dentre uma família de operadores candidatos, segundo uma medida de erro. Os mesmos princípios usados em [1] para delimitar a família de operadores candidatos foram empregados na etapa de estimação das distribuições condicionais, através da utilização de uma pirâmide como um modelo que, a partir dos dados de treinamento, induz uma distribuição condicional. Então, o problema de escolha da pirâmide foi tratado como um problema de escolha de distribuições condicionais de massa concentrada, com base na hipótese de que, para cada padrão observado pela janela, as probabilidades condicionais dos valores de saída do operador concentram-se em torno de um dos possíveis valores de saída. É interessante observar que esta visão separa o problema de estimação das distribuições do problema de escolha do operador, sendo possível a utilização de diferentes restrições sobre o espaço de distribuições e a família de operadores candidatos de forma independente. Por exemplo, de posse da distribuição estimada usando uma pirâmide, se tivermos a informação de que o operador desejado para o problema em questão é um operador crescente, poderíamos restringir a família de operadores candidatos à família dos operadores crescentes durante a etapa de escolha do operador ótimo, com o objetivo de melhorar a precisão do operador projetado.

Em trabalhos recentes de redução de dimensionalidade [10, 11, 12, 13], a entropia condicional foi utilizada como uma função critério para a escolha de um subespaço de características cujas distribuições

condicionais têm massa concentrada. Inspirados em tais avanços, realizamos alguns experimentos com a entropia condicional como um critério para determinar automaticamente uma boa pirâmide. Além das distribuições de probabilidade condicional (que indicam as probabilidades de um determinado valor de saída ocorrer dado que um determinado padrão foi observado pela janela), para calcular a entropia condicional também é necessário conhecer a distribuição de probabilidade dos padrões possíveis de serem observados pela janela (que indica, para cada padrão, qual a probabilidade deste ser observado). Assim, para estimar tal distribuição introduzimos um algoritmo que baseia-se no arcabouço piramidal multirresolução dos trabalhos anteriores que, além de estimar as distribuições de probabilidade condicional, estima a distribuição dos possíveis padrões observados pela janela. Desta forma, com as duas distribuições temos uma representação da distribuição conjunta, que relaciona os padrões possíveis de serem observados pela janela com os possíveis valores de saída do operador. Combinando o algoritmo com o critério da entropia, temos uma técnica de estimação das distribuições condicionais que procura por distribuições que apresentam massa de probabilidade concentrada. O problema real escolhido para testar a metodologia foi o de reconhecimento de dígitos manuscritos, um problema clássico na área de visão computacional.

1.2 Contribuições

Como principais contribuições deste trabalho, podemos citar:

- A criação de um algoritmo de estimação da distribuição conjunta a partir de uma estrutura piramidal de janelas e um conjunto de exemplos de entrada e saída. Diferentemente dos trabalhos anteriores na área, o algoritmo também estima a distribuição de probabilidade dos possíveis padrões observados pela janela;
- A experimentação com a entropia condicional como um critério para auxiliar o projetista de operadores na escolha da pirâmide;
- Combinando o algoritmo de estimação da distribuição conjunta com o critério da entropia, temos uma técnica de estimação das distribuições condicionais que procura por distribuições que apresentam massa de probabilidade concentrada;
- A implementação e a experimentação com mapeamentos de resolução da teoria de pirâmides de imagens no projeto de operadores multirresolução;
- A aplicação do projeto multirresolução de operadores morfológicos a partir de exemplos ao problema de reconhecimento de dígitos manuscritos.

1.3 Organização da Dissertação

O restante deste texto está organizado como segue. Inicialmente, no Capítulo 2 apresentamos os fundamentos do projeto de operadores morfológicos a partir de exemplos (mais especificamente, dos W -operadores, que é a classe de operadores considerada neste trabalho), e algumas considerações de caráter prático. No Capítulo 3, revisamos a restrição de resolução e o conceito de operadores multirresolução, e apresentamos uma breve introdução à teoria de pirâmides de imagens, indicando como esta pode ser usada no projeto multirresolução de operadores morfológicos a partir de exemplos. Em seguida, no Capítulo 4 introduzimos um algoritmo baseado em uma estrutura piramidal multirresolução para estimar a distribuição conjunta de probabilidades no projeto de W -operadores. O Capítulo 5 apresenta o conceito da entropia condicional como um critério a ser usado na escolha da pirâmide. No Capítulo 6 expomos brevemente a especificação do *software* que implementamos para testar nossa metodologia, e no Capítulo 7 descrevemos os experimentos realizados com o problema de reconhecimento de dígitos manuscritos e analisamos os resultados obtidos. Por fim, no Capítulo 8 apresentamos as conclusões deste trabalho e apontamos algumas possíveis direções para pesquisa futura. Ao final do texto temos três apêndices: um levantamento de mapeamentos de resolução da teoria de pirâmides de imagens, que fizemos com o objetivo de aplicá-los ao projeto de operadores a partir de exemplos; uma referência a uma publicação resultante deste trabalho; e uma lista de símbolos utilizados ao longo do texto.

Capítulo 2

Projeto de Operadores Morfológicos

A área de Processamento Digital de Imagens [14] concentra-se no estudo de operadores de imagens, isto é, em transformações que mapeiam imagens digitais em imagens digitais. A Morfologia Matemática [15, 16, 17], teoria que estuda mapeamentos entre reticulados completos, é bastante útil para modelar transformações entre imagens, e tem sido utilizada com sucesso para resolver diversos problemas, como processamento de documentos, reconhecimento de caracteres e filtragem de ruído. Informalmente, podemos entender os operadores morfológicos como transformações cujos resultados dependem de como a forma de um dado conjunto, conhecido como elemento estruturante, relaciona-se com as formas dos objetos presentes na imagem.

Neste trabalho, abordamos alguns aspectos do problema de projeto de operadores morfológicos a partir de exemplos. No presente capítulo definiremos o problema e apresentaremos algumas considerações de caráter prático.

2.1 Técnicas de Projeto

Resolver um problema de processamento de imagens pode ser uma tarefa bastante complexa. Quando um especialista em imagens procura uma solução baseada em operadores morfológicos, tipicamente duas abordagens são utilizadas. A primeira, conhecida como projeto *heurístico* ou *ad-hoc*, é a mais tradicional. O projetista utiliza seu conhecimento prévio sobre as propriedades dos diversos tipos de operadores e sobre a família de imagens considerada para escolher uma seqüência de operadores e elementos estruturantes que resolva o problema satisfatoriamente. Em geral, isto depende da experiência e intuição do especialista e envolve tentativa e erro.

Motivados pela complexidade deste processo, alguns grupos de pesquisa têm trabalhado na criação de técnicas para projetar operadores de imagens automaticamente, a partir de uma coleção de exemplos de entrada e saída do operador desejado. Tais abordagens, conhecidas como projeto *a partir de exemplos*

ou projeto *baseado em aprendizado estatístico e computacional* dependem da disponibilidade de pares de imagens de entrada e saída que representam o resultado desejado. A idéia é criar um sistema que receba como entrada uma coleção de pares de imagens (os exemplos) e devolva um operador que quando aplicado a outras imagens tenha efeito semelhante ao observado nos exemplos. Um sistema deste tipo poderia também ser utilizado por profissionais que não tenham formação em processamento de imagens para projetar operadores úteis em seu domínio de aplicação. Por exemplo, um cardiologista poderia, usando uma interface apropriada, marcar as regiões correspondentes a vasos sanguíneos em algumas imagens de angiografia. A partir de tais imagens, o sistema encontraria um operador que ao ser aplicado a outras imagens similares detectaria as regiões dos vasos.

Combinando as duas abordagens, o projeto *híbrido* de operadores morfológicos propõe a utilização conjunta das duas metodologias. A mistura entre elas pode ser feita de diversas formas, como por exemplo, no projeto usando envelopes [18], onde dois operadores obtidos heurísticamente determinam um limitante superior e um inferior para o operador projetado usando aprendizado. Pode-se também projetar um operador por aprendizado e depois refinar a solução obtida usando um operador projetado heurísticamente; ou podemos dividir o problema em etapas e usar os dois tipos de projeto em etapas diferentes.

2.2 W-operadores

Neste trabalho, abordamos o problema de projeto automático de operadores morfológicos restrito à classe dos W -operadores [19]. Ao aplicar um W -operador sobre uma imagem, o resultado depende apenas de uma vizinhança do pixel sendo avaliado, definida por uma janela W , e a regra de transformação aplicada é a mesma para todos os pontos da imagem. Vamos agora apresentar a definição da classe dos W -operadores.

Imagens digitais podem ser formalmente definidas e representadas por funções de um retângulo finito E em um intervalo não vazio L . Usualmente, E é um subconjunto de $\mathbb{Z} \times \mathbb{Z}$ e L é um subconjunto $[0, l - 1]$ dos inteiros não-negativos, $l \in \mathbb{Z}^+$ (em geral, $l = 2^k$, com $k \in \mathbb{Z}^+$). Imagens binárias podem ser representadas por elementos da coleção de subconjuntos de E , denotada por $\mathcal{P}(E)$. Elas podem também ser representadas como uma função de E em $[0, 1]$. O conjunto de todas as funções de E em L será denotado por L^E . Um mapeamento Ψ de L^E em L'^E será chamado de *operador* de imagens ou *filtro*, onde L' é o intervalo $[0, l' - 1]$, com $l' \in \mathbb{Z}^+$.

Um subconjunto finito W de E será chamado de *janela* e o número de pontos em W será denotado por $|W|$. Uma *configuração* é uma função de W em L e o espaço de todas as possíveis configurações de W em L será denotado por L^W . Uma configuração, também conhecida como W -padrão, usualmente é obtida através da translação de uma janela W por t , $t \in E$, e da observação dos valores de uma imagem $h \in L^E$ sob a janela transladada, W_t . Se $W = \{w_1, w_2, \dots, w_n\}$, $n = |W|$, e associarmos os pontos de W

a uma n -upla $(w_1, w_2, \dots, w_n)$, então uma configuração $h(W_t)$ é dada por

$$h(W_t) = (h(t + w_1), h(t + w_2), \dots, h(t + w_n)). \quad (2.1)$$

Imagens digitais podem ser modeladas por funções aleatórias digitais [20], e, neste sentido, $h(W_t)$ é uma realização de um vetor aleatório $\mathbf{X} = (X_1, X_2, \dots, X_n)$, ou seja, $h(W_t) = \mathbf{x} = (x_1, x_2, \dots, x_n)$, onde $\mathbf{x}$ denota uma realização de $\mathbf{X}$.

Dizemos que um operador $\Psi : L^E \rightarrow L'^E$ é *invariante por translação* (i.t.) se e somente se $\Psi(h_t)(x) = [\Psi(h)](x - t)$, para todo $h \in L^E$ e $t \in E$, onde h_t representa a translação de h por t , isto é, $h_t(x) = h(x - t)$ para todo $x \in E$. Isto significa que transladar uma imagem por t e aplicar o operador é equivalente a aplicar o operador sobre a imagem e depois transladar o resultado por t . Outra interpretação possível é a de que objetos iguais, a menos de translação, na imagem de entrada serão mapeados para objetos iguais, a menos de translação, na imagem de saída.

Se h é uma função em L^E e $W \subseteq E$ é uma janela, definimos $\mathcal{F}_{h|W}$ como a família de funções em L^E que são iguais a h quando seu domínio é restrito a W . Ou seja, $\mathcal{F}_{h|W}$ é dada por $\mathcal{F}_{h|W} = \{g \in L^E : (g(w_1), g(w_2), \dots, g(w_n)) = (h(w_1), h(w_2), \dots, h(w_n))\}$. Um operador $\Psi : L^E \rightarrow L'^E$ é *localmente definido* (l.d.) em uma janela W se e somente se, para quaisquer $f \in L^E$ e $x \in E$, $\Psi(f)(x) = \Psi(g)(x)$, para todo $g \in \mathcal{F}_{f|W_x}$. Assim, o resultado do operador em um ponto x depende apenas de uma vizinhança ao redor do ponto, definida pela janela W .

Uma importante subclasse dos operadores de L^E em L'^E é a classe dos W -operadores [19]. Um operador $\Psi : L^E \rightarrow L'^E$ é um W -operador se e somente se Ψ é i.t. e l.d. em uma janela W . Os W -operadores possuem uma propriedade importante em relação à sua representação: se Ψ é um W -operador, então ele pode ser caracterizado por uma função $\psi : L^W \rightarrow L'$, chamada *função característica*, da seguinte forma:

$$\Psi(h)(t) = \psi(h(t + w_1), h(t + w_2), \dots, h(t + w_n)) = \psi(\mathbf{x}). \quad (2.2)$$

Portanto, projetar um W -operador é equivalente a encontrar uma função característica que defina o operador. No restante deste texto, algumas vezes utilizaremos o termo *operador* ao nos referirmos a uma função característica.

2.3 Aprendizado do Operador

Formalmente, o problema de projetar um W -operador a partir de exemplos pode ser enunciado da seguinte maneira: fixada uma janela W , e dadas duas imagens no domínio E , h a ser observada e g a ser estimada, encontrar um W -operador Ψ que minimiza uma medida de erro entre $\Psi(h)(t)$ e $g(t)$, $t \in E$. Mais especificamente, se $\mathbf{X}$ é uma variável aleatória sobre L^W , Y é uma variável aleatória sobre L' , $\Phi =$

$\{\psi : \psi \text{ é uma função de } L^W \text{ em } L'\}$ é a família de funções características equivalentes aos W -operadores e $l(y, \psi(\mathbf{x}))$ é a função de perda que quantifica a diferença entre o valor ideal $y \in \mathcal{A}_Y^1$ e o valor $\psi(\mathbf{x})$ devolvido pelo operador, o problema consiste em encontrar uma função característica $\psi_{opt} \in \Phi$ que minimize o erro médio

$$E[l(y, \psi(\mathbf{x}))] = \sum_{\mathbf{x} \in \mathcal{A}_X} \sum_{y \in \mathcal{A}_Y} P(\mathbf{x}, y) l(y, \psi(\mathbf{x})), \quad (2.3)$$

ou seja,

$$\sum_{\mathbf{x} \in \mathcal{A}_X} \sum_{y \in \mathcal{A}_Y} P(\mathbf{x}, y) l(y, \psi_{opt}(\mathbf{x})) = E[l(y, \psi_{opt}(\mathbf{x}))] \leq E[l(y, \psi(\mathbf{x}))] = \sum_{\mathbf{x} \in \mathcal{A}_X} \sum_{y \in \mathcal{A}_Y} P(\mathbf{x}, y) l(y, \psi(\mathbf{x})) \quad (2.4)$$

para todo $\psi \in \Phi$. Isto caracteriza um problema de otimização.

Como $P(\mathbf{x}, y) = P(y | \mathbf{x}) \cdot P(\mathbf{x})$, temos que

$$E[l(y, \psi(\mathbf{x}))] = \sum_{\mathbf{x} \in \mathcal{A}_X} \sum_{y \in \mathcal{A}_Y} P(y | \mathbf{x}) P(\mathbf{x}) l(y, \psi(\mathbf{x})) = \sum_{\mathbf{x} \in \mathcal{A}_X} P(\mathbf{x}) \sum_{y \in \mathcal{A}_Y} P(y | \mathbf{x}) l(y, \psi(\mathbf{x})). \quad (2.5)$$

Claramente, se o operador $\psi(\mathbf{x})$ for escolhido de forma que $\sum_{y \in \mathcal{A}_Y} P(y | \mathbf{x}) l(y, \psi(\mathbf{x}))$ seja mínimo para todo $\mathbf{x}$, então o erro médio é minimizado. Portanto, para decidirmos qual é o operador ótimo basta conhecermos as distribuições de probabilidade condicional² $p(Y | \mathbf{x})$, para todo $\mathbf{x} \in \mathcal{A}_X$.

A medida de erro usual para operadores de $\mathcal{P}(E)$ em $\mathcal{P}(E)$ (operadores binários) é o Erro Absoluto Médio (Mean Absolute Error, ou MAE) [20], que utiliza $l(y, \psi(\mathbf{x})) = |y - \psi(\mathbf{x})|$ como função de perda. Para operadores de $\mathcal{P}(E)$ em $[0, l - 1]^E$ (classificadores de imagens binárias), em geral usa-se o número total de objetos ou pontos incorretamente classificados. Para operadores de L^E em L'^E (operadores em níveis de cinza), costuma-se usar o Erro Quadrático Médio (Mean Square Error, ou MSE) [20], dado pela função de perda $l(y, \psi(\mathbf{x})) = (y - \psi(\mathbf{x}))^2$.

Na prática, as distribuições de probabilidade condicional são desconhecidas. Uma amostra de $p(\mathbf{X}, Y)$, obtida a partir dos pares de imagens de entrada e saída³, é utilizada para obter estimativas $\hat{p}(Y | \mathbf{x})$ das distribuições condicionais, e o problema de otimização é resolvido usando a estimativa $\hat{p}$ no lugar de p . Assim, encontra-se o operador ótimo $\hat{\psi}_{opt} \in \Phi$ de acordo com a distribuição $\hat{p}$, que poderá ter erro maior do que o operador ψ_{opt} devido a erros na estimação de $\hat{p}$. Com isso, fica evidente a importância de realizarmos uma boa estimação.

É interessante notar a semelhança do problema de projeto de W -operadores com a abordagem es-

¹Se X é uma variável aleatória, denotaremos por $\mathcal{A}_X$ o conjunto de valores que X pode assumir. Por exemplo, $\mathcal{A}_Y = L'$ neste caso.

²Neste texto, utilizaremos $p(\cdot)$ para denotar uma função distribuição de probabilidade e $P(\cdot)$ para denotar os valores de massa de probabilidade.

³Também conhecidos como imagens de treinamento.

tatística para o problema de reconhecimento de padrões [21]. Podemos entender uma configuração $\mathbf{x}$ como um vetor de características, e o operador ψ como um classificador que atribui uma classe $y \in \mathcal{A}_Y$ ao vetor $\mathbf{x}$. De fato, muitos dos conceitos apresentados nesta dissertação no contexto de projeto de W -operadores também valem para o problema de projeto de classificadores.

2.3.1 Um Exemplo Ilustrativo

O projeto de um W -operador por aprendizado computacional envolve as seguintes etapas:

- coleta de exemplos a partir dos pares de imagens;
- estimação das distribuições de probabilidade condicional;
- escolha do operador ótimo segundo uma medida de erro.

Vamos agora mostrar um exemplo bastante simples de como o processo funciona. Usaremos um par de imagens binárias como exemplo para projetar um operador $\psi : \{0, 1\}^W \rightarrow \{0, 1\}$, ou seja, um operador binário. A janela W utilizada será de apenas três pontos (para que a tabela de possíveis padrões observados não seja muito grande, de modo a facilitar a compreensão).

A etapa de coleta consiste em, fixada uma janela W , transladá-la de modo que o seu centro fique sobre os diversos pixels da imagem de entrada e observar o padrão sob a janela, acompanhado do valor de saída no pixel central. Com isso, monta-se uma tabela indicando quantas vezes cada padrão foi observado, com seus respectivos valores de saída. A Figura 2.1 ilustra o processo para a janela de 3 pontos sobre o par de imagens de treinamento, e a Tabela 2.1 contém o resultado das observações após os padrões terem sido coletados em todos os pontos nos quais a janela está contida na região da imagem. f_0 denota o número de vezes que uma configuração apareceu associada ao valor de saída 0, e f_1 denota o número de vezes que uma configuração apareceu associada ao valor de saída 1. Utilizamos a convenção de que pixels do fundo da imagem têm valor zero, e pixels correspondentes aos objetos têm valor um. Note que para o padrão 111 (última linha da Tabela 2.1) existem observações com ambos os valores de saída; tais exemplos são conhecidos como *exemplos contraditórios*.

De posse da tabela de exemplos, realiza-se a estimação das distribuições condicionais $p(Y | \mathbf{x})$, com $\mathbf{x} \in \mathcal{A}_X$. A Tabela 2.2 mostra a distribuição estimada no exemplo em questão. A estimação de $P(Y = b | \mathbf{x})$ é feita contando o número de vezes em que a configuração $\mathbf{x}$ apareceu associada ao valor de saída b e tomando a razão entre este número e o total de vezes em que $\mathbf{x}$ foi observada nos exemplos.

Finalmente, de posse das distribuições condicionais usa-se um critério de decisão que minimize uma medida de erro para determinar a função característica que define o operador. No caso binário, o critério que minimiza o Erro Absoluto Médio consiste em escolher, para uma determinada configuração $\mathbf{x}$, $\hat{\psi}(\mathbf{x}) = 0$ se $P(Y = 0 | \mathbf{x}) \geq 0,5$, e escolher $\hat{\psi}(\mathbf{x}) = 1$ caso contrário. A Tabela 2.3 mostra o operador projetado usando este critério.

Figura 2.1: Coleta de exemplos: a) janela (com indicação do centro); b) imagem de entrada; c) imagem de saída; d) exemplo coletado.

$\mathbf{x}$	f_0	f_1
000	24	0
100	6	0
010	0	5
001	6	0
110	0	6
101	2	0
011	0	6
111	17	5

Tabela 2.1: Tabela de exemplos.

$\mathbf{x}$	$P(Y = 0 \mathbf{x})$	$P(Y = 1 \mathbf{x})$
000	1	0
100	1	0
010	0	1
001	1	0
110	0	1
101	1	0
011	0	1
111	17/22	5/22

Tabela 2.2: Estimativas das distribuições de probabilidade condicional.

A aplicação do operador sobre uma imagem é feita através da translação da janela W para os diversos pontos da imagem, atribuindo ao pixel central da janela na imagem da saída o valor da função para a configuração observada. A Figura 2.2 ilustra o processo, e a Figura 2.3 mostra o resultado final da

$\mathbf{x}$	$\hat{\psi}(\mathbf{x})$
000	0
100	0
010	1
001	0
110	1
101	0
011	1
111	0

Tabela 2.3: Operador projetado.

aplicação do operador representado pela Tabela 2.3.

Figura 2.2: Aplicação do operador: a) imagem de entrada; b) imagem de saída após algumas iterações do procedimento de aplicação.

Figura 2.3: Resultado da aplicação.

2.4 Problemas com Janelas Grandes

Uma importante decisão a ser tomada ao se projetar um W -operador consiste na escolha da janela W . Quanto maior for a janela, melhor será a sua capacidade de distinção entre diferentes padrões observados na imagem. Por outro lado, o número de variáveis aleatórias no vetor $\mathbf{X}$ aumenta, tornando o problema de estimação das distribuições condicionais cada vez mais complicado, já que o número de exemplos em geral é limitado.

Do ponto de vista estatístico, a dificuldade vem da limitação na quantidade de amostras disponíveis para treinamento. O tamanho da janela, $|W|$, possui uma relação exponencial com o número de exemplos necessários para se obter uma boa estimativa das distribuições condicionais. Porém, o número de exemplos disponíveis é, em geral, muito menor do que $|\mathcal{A}_{\mathbf{X}}| = |L|^{|W|}$. Portanto, a técnica de aprendizado precisa ser capaz de determinar valores do operador $\psi(\mathbf{x})$ para configurações $\mathbf{x}$ que não aparecem no conjunto de treinamento. No contexto de projeto de classificadores, isto é conhecido como o problema da *generalização*.

Mesmo que tivéssemos uma quantidade muito grande de dados de treinamento, e o problema acima não ocorresse, existiria a limitação computacional. A quantidade de memória e o tempo de processamento necessários para projetar o operador seriam muito grandes. Outro problema de interesse refere-se à *representação* do operador. Por exemplo, para um operador binário de janela 17×17 , existem $2^{17 \times 17} = 2^{289}$ configurações possíveis de serem observadas pela janela, ou seja, temos aproximadamente 10^{87} possíveis configurações, o que é maior do que $3 \cdot 10^{74}$, uma estimativa conhecida para o número de átomos no universo! [22] Uma questão que surge é: como representar de maneira eficiente a função característica que define o operador?

2.5 Projeto Utilizando Restrições

Como vimos, a limitação na quantidade de imagens disponíveis para treinamento dificulta a obtenção de uma boa estimativa do operador ótimo. Ao projetar o operador a partir de probabilidades estimadas, o operador projetado $\hat{\psi}_{opt} \in \Phi$ é um estimador do operador ótimo ψ_{opt} . Isto significa que, devido a erros na estimação da distribuição, existe um aumento no erro $\Delta(\hat{\psi}_{opt}, \psi_{opt})$ associado a $\hat{\psi}_{opt}$, ou seja,

$$E[l(y, \hat{\psi}_{opt}(\mathbf{x}))] = E[l(y, \psi_{opt}(\mathbf{x}))] + \Delta(\hat{\psi}_{opt}, \psi_{opt}). \quad (2.6)$$

Se $\Delta(\hat{\psi}_{opt}, \psi_{opt})$ for pequeno, então o operador projetado é quase ótimo. Porém, $\Delta(\hat{\psi}_{opt}, \psi_{opt})$ é em geral significativo ao utilizarmos janelas grandes.

Com o objetivo de atenuar o problema, uma idéia que tem sido explorada é a de impor restrições à família de operadores Φ , realizando a busca pelo operador ótimo em uma subclasse $\Phi' \subset \Phi$. Considere o problema de projetar um operador ótimo ψ'_{opt} em Φ' . Devido aos erros de estimação, na verdade

projetamos um operador $\hat{\psi}'_{opt} \in \Phi'$ e, como anteriormente, temos um aumento no erro $\Delta(\hat{\psi}'_{opt}, \psi'_{opt})$:

$$E[l(y, \hat{\psi}'_{opt}(\mathbf{x}))] = E[l(y, \psi'_{opt}(\mathbf{x}))] + \Delta(\hat{\psi}'_{opt}, \psi'_{opt}). \quad (2.7)$$

Podemos ver $\Delta(\hat{\psi}_{opt}, \psi_{opt})$ e $\Delta(\hat{\psi}'_{opt}, \psi'_{opt})$ como variáveis aleatórias que dependem das amostras de treinamento. Portanto, existe o conceito de *aumento de erro médio*, dado por $E[\Delta(\hat{\psi}_{opt}, \psi_{opt})]$ e $E[\Delta(\hat{\psi}'_{opt}, \psi'_{opt})]$. Através da combinação das Equações 2.6 e 2.7, vemos que projetar um operador ótimo em Φ' pode ser mais vantajoso do que projetar um operador ótimo em Φ se

$$\Delta(\psi_{opt}, \psi'_{opt}) + E[\Delta(\hat{\psi}'_{opt}, \psi'_{opt})] \leq E[\Delta(\hat{\psi}_{opt}, \psi_{opt})], \quad (2.8)$$

mesmo que ψ'_{opt} seja subótimo em Φ . Isto significa que o erro de estimação do operador ótimo ψ_{opt} é maior do que a soma do aumento do erro de ψ'_{opt} em relação a ψ_{opt} , chamado de *custo de restrição* e denotado por $\Delta(\psi_{opt}, \psi'_{opt})$, com o erro de estimação $E[\Delta(\hat{\psi}'_{opt}, \psi'_{opt})]$ de ψ'_{opt} em Φ' . Assim, é interessante utilizar uma restrição quando a diferença nos erros de estimação do operador ótimo nos espaços restrito e não-restrito compensa o custo de restrição. Em geral, isto depende da classe de imagens sendo considerada e da quantidade de amostras de treinamento disponíveis. Tipicamente, impor uma restrição é vantajoso para conjuntos de amostras de até um certo tamanho, mas não para quantidades maiores de exemplos. Um caso particular ocorre quando a restrição é escolhida de forma que o operador ótimo ψ_{opt} está em Φ' (por exemplo, sabe-se de antemão que ψ_{opt} obedece a uma certa estrutura algébrica, e escolhe-se o conjunto Φ' como a família de operadores com tal propriedade); neste caso, não há custo de restrição.

Entre as restrições utilizadas recentemente no projeto de W-operadores, podemos citar a busca por operadores crescentes [5] e as restrições de resolução [1, 6]. Este tipo de restrição é do tipo algébrico, ou seja, as subclasses de operadores consideradas obedecem a alguma estrutura algébrica.

Uma outra maneira de reduzir os erros de estimação consiste em assumir que as distribuições de probabilidade condicional seguem algum modelo de probabilidade, restringindo a família de distribuições candidatas durante a estimação. Por exemplo, a partir de conhecimento sobre o problema poderíamos assumir que as distribuições condicionais $p(Y | \mathbf{x})$, com $\mathbf{x} \in \mathcal{A}_{\mathbf{X}}$ sejam distribuições Gaussianas, reduzindo o problema à estimação dos parâmetros μ (média) e σ (desvio padrão). Este tipo de procedimento não resulta em um custo de restrição, pois o espaço de operadores sendo considerado ainda é Φ ; no entanto, pode haver aumento no erro de estimação se o modelo de probabilidade escolhido não for apropriado.

Resumindo, as duas maneiras têm como objetivo aumentar a precisão do operador estimado, através da diminuição dos erros de estimação. No primeiro caso o espaço de busca do operador é delimitado por uma restrição algébrica. É vantajoso utilizar a restrição quando a diferença entre os erros de estimação do operador ótimo nos espaços restrito e não-restrito compensa o custo de restrição. Mesmo que o operador ótimo não esteja presente na subclasse considerada pela restrição, o operador encontrado pode ser melhor do que o encontrado sem usar restrições devido à diminuição no erro de estimação. Já no caso de utilização

de um modelo de probabilidade durante a etapa de estimação da distribuição, não existe restrição sobre os operadores considerados, mas apenas sobre a distribuição de probabilidade. No entanto, se o modelo de probabilidade escolhido não for adequado o erro de estimação pode aumentar ao invés de diminuir.

Do ponto de vista metodológico, as restrições sobre o espaço de operadores e a delimitação das distribuições de probabilidade segundo algum modelo podem ser vistas como duas formas de agregação de conhecimento a priori ao problema. Até o momento, não conhecemos nenhum trabalho que explore a utilização conjunta de um modelo sobre o espaço de distribuições considerado na estimação e uma restrição algébrica sobre o espaço de busca do operador. Como veremos nos próximos capítulos, neste trabalho usamos as estruturas piramidais, anteriormente empregadas para restringir o espaço de busca do operador pela restrição de resolução [1], como modelos que impõem uma certa estrutura às distribuições estimadas, que varia de acordo com as amostras de treinamento. Não definimos restrições sobre o espaço de operadores considerado. No futuro, pode ser interessante verificar a utilização de restrições sobre o espaço de operadores em conjunto com a nossa abordagem de estimação da distribuição.

Capítulo 3

Projeto Piramidal Multirresolução

Dentre as diversas restrições utilizadas no projeto de W -operadores, a restrição de resolução [1] é a que tem proporcionado melhores resultados para operadores de janelas grandes. Tal restrição parte do princípio de que o operador ótimo em uma resolução maior em geral é melhor do que o operador ótimo em uma resolução menor; porém, a utilização de uma janela maior implica em um aumento no número de variáveis aleatórias observadas sob a janela, conseqüentemente levando a um aumento no erro de estimação do operador ótimo. Para uma determinada quantidade de amostras de treinamento, é vantajoso projetar um operador em uma resolução menor se a diminuição do erro de estimação é maior do que o aumento do erro pelo custo de restrição. De uma maneira mais geral, podemos reduzir a resolução em diversas etapas (de forma piramidal) e determinar uma resolução onde a soma do custo de restrição com o erro de estimação seja mínima.

Neste capítulo descreveremos os principais conceitos do projeto de W -operadores baseado na delimitação do espaço de busca do operador pela restrição de resolução [1] e faremos uma breve introdução à teoria de pirâmides de imagens [9], mostrando como ela pode ser utilizada no projeto multirresolução de W -operadores.

3.1 Restrição de Resolução

Qualitativamente, se W_0 e W_1 são janelas correspondentes à mesma região euclidiana D , então a resolução de W_1 é menor do que a de W_0 se W_1 contém menos pixels do que W_0 . A restrição de resolução parte do princípio de projetar um W -operador observando a região vista por W em resoluções menores. Assim, o número de pontos nas configurações observadas será menor, o que resulta em um número menor de distribuições de probabilidade condicional a estimar.

Vamos iniciar definindo um mapeamento de resolução para o caso específico da subamostragem e para duas janelas W_0 e W_1 , $W_1 \subseteq W_0$. Depois estenderemos a definição para mapeamentos de resolução

genéricos e enunciaremos a restrição de resolução. Finalmente, faremos a extensão para múltiplas janelas (esquema piramidal). Sejam $D_0 = L^{W_0}$ e $D_1 = L^{W_1}$ os espaços das configurações que podem ser observadas através das janelas W_0 e W_1 , respectivamente, onde $L = [0, l - 1]$, $l \in \mathbb{Z}^+$, e seja $\rho : D_0 \rightarrow D_1$ um mapeamento de subamostragem, isto é, um mapeamento que atribui a cada configuração $\mathbf{x} \in D_0$ uma configuração $\mathbf{z} = \rho(\mathbf{x}) \in D_1$ através da seleção dos valores de $\mathbf{x}$ nos pontos de W_1 . Por exemplo, na Figura 3.1 podemos ver uma subamostragem de uma janela 3×3 para uma janela 2×2 . Cada padrão $\mathbf{x} = (x_1, \dots, x_9)$ em D_0 é mapeado para um padrão $\mathbf{z} = (z_1, z_2, z_3, z_4)$ em D_1 por $z_1 = x_1$, $z_2 = x_3$, $z_3 = x_7$, $z_4 = x_9$. Em relação ao projeto de operadores, se o objetivo é estimar o valor de uma variável aleatória Y usando as amostras, o operador poderia ser projetado usando as amostras em alta resolução (observadas por W_0) ou as amostras em baixa resolução (observadas por W_1).

Figura 3.1: Exemplo de subamostragem.

O mapeamento de resolução ρ acima foi definido de acordo com o procedimento de subamostragem. No entanto, em termos matemáticos ele poderia ser livremente especificado. Vamos agora estender a definição para mapeamentos arbitrários. Seja $D_0 = L^{W_0}$ o espaço das configurações que podem ser observadas através da janela W_0 , e seja $D_1 \subseteq L^{W_1}$ um espaço de configurações de menor resolução, onde $L = [0, l - 1]$, $l \in \mathbb{Z}^+$. Note que nesta formulação mais geral, D_1 não é necessariamente igual a L^{W_1} . Considerando um mapeamento de resolução $\rho : D_0 \rightarrow D_1$, podemos dizer que ele determina uma relação de equivalência $\sim$ em D_0 , por $\mathbf{x} \sim \mathbf{x}' \Leftrightarrow \rho(\mathbf{x}) = \rho(\mathbf{x}')$, para $\mathbf{x}, \mathbf{x}' \in D_0$. Esta relação induz uma partição $\mathcal{X}^1 = \{\mathcal{X}_1^1, \mathcal{X}_2^1, \dots, \mathcal{X}_{n_1}^1\}$, do espaço D_0 , com $n_1 = |D_1|$: $\mathbf{x}$ e $\mathbf{x}'$ estão no mesmo conjunto $\mathcal{X}_k^1$ de $\mathcal{X}^1$ ($1 \leq k \leq n_1$) se e somente se $\mathbf{x} \sim \mathbf{x}'$. A partição $\mathcal{X}^1$ consiste de n_1 subconjuntos disjuntos $\mathcal{X}_1^1, \mathcal{X}_2^1, \dots, \mathcal{X}_{n_1}^1$ de D_0 e a união dos subconjuntos em $\mathcal{X}^1$ é igual a D_0 . A Figura 3.2 mostra a partição $\mathcal{X}^1$ para $L = \{0, 1\}$, $W_0 = \{w_1, w_2, w_3\}$, $W_1 = \{w_2, w_3\}$, e o mapeamento ρ definido como a subamostragem de W_0 em W_1 .

Em outras palavras, se $D_1 = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{n_1}\}$, onde $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{n_1}$ são configurações distintas em L^{W_1} , podemos definir a correspondência $\mathbf{z}_k \leftrightarrow \mathcal{X}_k^1$, com $1 \leq k \leq n_1$; a partição é definida através dos conjuntos inversos de ρ , ou seja, $\mathcal{X}_k^1 = \rho^{-1}(\mathbf{z}_k)$. Para cada $\mathbf{z}$ em D_1 a classe de equivalência correspondente será denotada por $C[\mathbf{z}] = \rho^{-1}(\mathbf{z})$. No exemplo da Figura 3.2, $D_1 = \{00, 10, 01, 11\}$ e $C[00] = \{100, 000\} = \mathcal{X}_1^1$, $C[10] = \{010, 110\} = \mathcal{X}_2^1$, $C[01] = \{001, 101\} = \mathcal{X}_3^1$, e $C[11] = \{011, 111\} = \mathcal{X}_4^1$.

Um operador $\phi : D_1 \rightarrow L$ induz um operador $\psi_\phi : D_0 \rightarrow L$ por $\psi_\phi(\mathbf{x}) = \phi(\rho(\mathbf{x}))$. ψ_ϕ é restrito por resolução de acordo com a partição $\mathcal{X}^1$: se $\mathbf{x} \sim \mathbf{x}'$, então $\psi_\phi(\mathbf{x}) = \psi_\phi(\mathbf{x}')$. Por outro lado, se ψ é um operador de D_0 em L que satisfaz a restrição de resolução, então ele induz um operador ϕ_ψ de D_1 em L

Figura 3.2: Exemplo de partição.

por $\phi_\psi(\mathbf{z}) = \psi(\mathbf{x})$, onde $\mathbf{x}$ é alguma configuração em $C[\mathbf{z}]$. Sejam O_0 e O_1 as classes de operadores de D_0 em L e de D_1 em L , respectivamente. O mapeamento $\psi \rightarrow \phi_\psi$ é sobrejetor de O_0 em O_1 e o mapeamento $\phi \rightarrow \psi_\phi$ é injetor de O_1 em O_0 [1]. Se R_0 é o subconjunto de O_0 dado pelos operadores que satisfazem a restrição de resolução e $\psi \in R_0$, então $\psi \rightarrow \phi_\psi$ define uma bijeção $R_0 \rightarrow O_1$ cuja inversa é dada por $\phi \rightarrow \psi_\phi$. Esta bijeção permite identificarmos operadores em D_1 com operadores restritos por resolução em D_0 . Em particular, é possível mostrar que o operador ótimo em O_1 corresponde ao operador ótimo restrito por resolução em R_0 [6]. A Figura 3.3 ilustra estes conceitos.

Figura 3.3: Correspondência entre os operadores.

Vamos agora caracterizar um esquema piramidal, que permite a utilização sucessiva da restrição por resolução. Seja $W_0, W_1, \dots, W_r$ uma seqüência de janelas tal que $W_{i+1} \subseteq W_i$, para $0 \leq i \leq r-1$, seja $D_0 = L_0^{W_0}$ o espaço das configurações que podem ser observadas pela janela W_0 , e sejam $D_1 \subseteq L_1^{W_1}, \dots, D_r \subseteq L_r^{W_r}$ espaços de configurações de menor resolução correspondentes às janelas $W_1, \dots, W_r$, onde L_i , para $i \in \{0, \dots, r\}$ são intervalos da forma $[0, l_i - 1]$, $l_i \in \mathbb{Z}^+$. Podemos definir uma seqüência de mapeamentos de resolução $\rho_{01} : D_0 \rightarrow D_1, \rho_{12} : D_1 \rightarrow D_2, \dots, \rho_{(r-1)r} : D_{r-1} \rightarrow D_r$ que induzem

uma seqüência de partições aninhadas $\mathcal{X}^1, \mathcal{X}^2, \dots, \mathcal{X}^r$ do espaço D_0 , determinadas pelas relações de equivalência $\mathbf{x} \sim_1 \mathbf{x}' \Leftrightarrow \rho_{01}(\mathbf{x}) = \rho_{01}(\mathbf{x}')$, para $\mathbf{x}, \mathbf{x}' \in D_0$, $\mathbf{x} \sim_2 \mathbf{x}' \Leftrightarrow \rho_{12}(\rho_{01}(\mathbf{x})) = \rho_{12}(\rho_{01}(\mathbf{x}'))$, e assim por diante. Além disso, defina a partição $\mathcal{X}^0 = \{\{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \dots, \{\mathbf{x}_{|D_0|}\}\}$, onde cada conjunto em $\mathcal{X}^0$ contém um único elemento de D_0 . Por construção, se $\mathbf{x} \sim_i \mathbf{x}'$ então $\mathbf{x} \sim_{i+1} \mathbf{x}'$, para $1 \leq i \leq r-1$.

Para ilustrar os conceitos do parágrafo anterior, vamos agora mostrar dois exemplos de particionamento do espaço D_0 . Suponha que $L_0 = L_1 = L_2 = \{0, 1\}$, temos a seqüência de janelas $W_0 = \{w_1, w_2, w_3\}$, $W_1 = \{w_2, w_3\}$ e $W_2 = \{w_3\}$, e os mapeamentos de resolução ρ_{01} e ρ_{12} são definidos como a subamostragem de W_0 em W_1 e a subamostragem de W_1 em W_2 , respectivamente. Então, $D_0 = \{100, 000, 010, 110, 001, 101, 011, 111\}$, $D_1 = \{00, 10, 01, 11\}$ e $D_2 = \{0, 1\}$. Tais mapeamentos de resolução induzem as partições aninhadas $\mathcal{X}^0$, $\mathcal{X}^1$ e $\mathcal{X}^2$, dadas por:

- $\mathcal{X}^0 = \{\{100\}, \{000\}, \{010\}, \{110\}, \{001\}, \{101\}, \{011\}, \{111\}\}$;
- $\mathcal{X}^1 = \{\{100, 000\}, \{010, 110\}, \{001, 101\}, \{011, 111\}\}$;
- $\mathcal{X}^2 = \{\{100, 000, 010, 110\}, \{001, 101, 011, 111\}\}$.

A Figura 3.4 ilustra o particionamento.

Figura 3.4: Exemplo de particionamento – subamostragem.

O próximo exemplo ilustra que os espaços de configurações $D_1, \dots, D_r$ não precisam ser necessariamente iguais a $L^{W_1}, \dots, L^{W_r}$, pois isto depende dos mapeamentos de resolução utilizados. Novamente, suponha que $L_0 = L_1 = L_2 = \{0, 1\}$, e que temos a seqüência de janelas $W_0 = \{w_1, w_2, w_3\}$, $W_1 = \{w_2, w_3\}$ e $W_2 = \{w_3\}$. Vamos definir os mapeamentos de resolução ρ_{01} e ρ_{12} como $\rho_{01}(100) = \rho_{01}(000) = \rho_{01}(010) = 00$, $\rho_{01}(110) = \rho_{01}(101) = 11$, $\rho_{01}(001) = \rho_{01}(011) = \rho_{01}(111) = 01$, $\rho_{12}(00) = \rho_{12}(01) = 0$ e $\rho_{12}(11) = 1$. Então, $D_0 = \{100, 000, 010, 110, 001, 101, 011, 111\}$, $D_1 = \{00, 01, 11\}$ e $D_2 = \{0, 1\}$. Tais mapeamentos de resolução induzem as partições aninhadas $\mathcal{X}^0$, $\mathcal{X}^1$ e $\mathcal{X}^2$, dadas por:

- $\mathcal{X}^0 = \{\{100\}, \{000\}, \{010\}, \{110\}, \{001\}, \{101\}, \{011\}, \{111\}\};$
- $\mathcal{X}^1 = \{\{100, 000, 010\}, \{110, 101\}, \{001, 011, 111\}\};$
- $\mathcal{X}^2 = \{\{100, 000, 010, 001, 011, 111\}, \{110, 101\}\}.$

A Figura 3.5 ilustra o particionamento.

Figura 3.5: Outro exemplo de particionamento.

Considerando a restrição por resolução no caso piramidal, esta pode ser entendida como um procedimento de particionamento progressivo do espaço de observações da janela W para obtermos uma seqüência de espaços de observações $D_0, D_1, \dots, D_r$ de resolução cada vez menor, que podem ser determinados pela aplicação dos mapeamentos de resolução $\rho_{01}, \rho_{12}, \dots, \rho_{(r-1)r}$. Aos espaços de observações corresponde uma seqüência de operadores ótimos restritos por resolução $\psi_0, \psi_1, \dots, \psi_r$. Para uma determinada quantidade de amostras de treinamento, deverá ser escolhido o operador ótimo da resolução que minimiza a soma do custo de restrição com o erro de estimação.

No restante desta dissertação, iremos nos referir a uma seqüência de janelas $W_0, \dots, W_r$ acompanhada por uma seqüência de mapeamentos de resolução $\rho_{01} : D_0 \rightarrow D_1, \rho_{12} : D_1 \rightarrow D_2, \dots, \rho_{(r-1)r} : D_{r-1} \rightarrow D_r$ como uma *pirâmide de janelas*. A Figura 3.6 ilustra esta estrutura para uma pirâmide de três janelas.

3.2 Operadores Multirresolução

Fixada uma pirâmide de janelas, a técnica descrita na seção anterior propõe a escolha de um dos espaços de configurações $D_0, \dots, D_r$ para projetar o operador. Digamos que a melhor resolução, isto é, a resolução em que a soma do custo de restrição com o erro de estimação é mínima, seja t , com $1 \leq t \leq r$. O operador ótimo é então projetado no espaço D_t e induz um operador ótimo restrito por resolução em D_0 , por

Figura 3.6: Pirâmide de janelas.

$\psi'_{opt}(\mathbf{x}) = \psi_t(\rho_{(t-1)t}(\dots \rho_{12}(\rho_{01}(\mathbf{x})))$. Assim, mesmo que para algumas configurações de D_0 exista uma boa quantidade de amostras em alguma resolução de D_0 a D_{t-1} , apenas as amostras na resolução de D_t são utilizadas para projetar o operador. Em outras palavras, para algumas configurações $\mathbf{x}$ em D_0 , podemos ter uma estimativa da distribuição condicional $p(Y | \mathbf{x})$ em alguma resolução de 0 a $t - 1$ que seja melhor do que a estimativa na resolução t , mas tal estimativa não será usada, pois o projeto é feito usando apenas amostras em D_t para toda configuração $\mathbf{x} \in D_0$.

Tendo isto em mente, em [1] os autores também propuseram uma abordagem híbrida, que permite utilizar amostras em diferentes resoluções no projeto do operador. É apresentada a definição de *operador multirresolução*: se M é o número de amostras de treinamento, sejam $\hat{\psi}_{M,0}, \hat{\psi}_{M,1}, \dots, \hat{\psi}_{M,r}$ os operadores ótimos projetados sobre os espaços $D_0, D_1, \dots, D_r$, e seja $N(\mathbf{x})$ o número de vezes que a configuração $\mathbf{x}$ foi observada nos exemplos. Se $\rho_i = \rho_{(i-1)i} \dots \rho_{12}\rho_{01}$, o operador multirresolução é definido como

$$\hat{\psi}_{M,(0,1,\dots,r)}(\mathbf{x}) = \begin{cases} \hat{\psi}_{M,0}(\mathbf{x}), & \text{se } N(\mathbf{x}) > 0; \\ \hat{\psi}_{M,1}(\rho_1(\mathbf{x})), & \text{se } N(\mathbf{x}) = 0, N(\rho_1(\mathbf{x})) > 0; \\ \vdots & \\ \hat{\psi}_{M,r-1}(\rho_{r-1}(\mathbf{x})), & \text{se } N(\mathbf{x}) = 0, \dots, N(\rho_{r-2}(\mathbf{x})) = 0, N(\rho_{r-1}(\mathbf{x})) > 0; \\ \hat{\psi}_{M,r}(\rho_r(\mathbf{x})), & \text{se } N(\mathbf{x}) = 0, \dots, N(\rho_{r-1}(\mathbf{x})) = 0. \end{cases} \quad (3.1)$$

A definição acima significa que para uma configuração $\mathbf{x} \in D_0$ escolhe-se a janela de maior resolução em que existam exemplos observados nas amostras para realizar a estimação de $\psi(\mathbf{x})$. A definição pode ser ligeiramente alterada para exigir que $N(\mathbf{x}) \geq \alpha$, para algum limiar $\alpha \in \{1, 2, \dots\}$, de modo a evitar uma estimação ruim do operador desejado. O *projeto de operadores multirresolução* pode ser visto como uma técnica de generalização; a generalização para as configurações não observadas em um nível da pirâmide é dada pelos níveis subseqüentes.

3.3 Pirâmides de Imagens

Ao analisar uma imagem, em algumas situações pode ser útil fazermos uma decomposição desta em partes separadas de modo que não haja perda de informação. Uma importante teoria na área de Processamento de Imagens é a teoria de pirâmides, que provê maneiras de realizar a decomposição de imagens em múltiplos níveis de resolução [23]. As decomposições piramidais têm sido utilizadas em diversas aplicações em imagens (ver o Apêndice A para alguns exemplos); um dos objetivos deste trabalho é realizar experimentos com mapeamentos de resolução da teoria de pirâmides de imagens no contexto do projeto multirresolução de operadores. Vamos agora descrever os seus principais conceitos.

Considere uma coleção de representações de uma imagem em resoluções espaciais distintas, empilhadas uma sobre a outra, com a imagem de maior resolução na base da pilha e as imagens subseqüentes aparecendo sobre ela em ordem decrescente de resolução. Isto gera uma estrutura semelhante a uma pirâmide, como pode ser visto na Figura 3.7. O procedimento tradicional para obtenção de uma imagem de menor resolução consiste em realizar uma filtragem passa-baixas seguida por uma amostragem [9]. Em [23, 24, 25, 26], Goutsias e Heijmans apresentam o conceito de pirâmide sob um enfoque mais formal, que reproduzimos abaixo. Esta formalização vale para sinais de dimensões arbitrárias (isto é, não apenas para imagens, que são sinais em duas dimensões).

Figura 3.7: Estrutura piramidal.

Suponha que temos uma seqüência de espaços de sinais $V_0, V_1, V_2, \dots$ e uma seqüência de espaços de sinais $W_1, W_2, \dots$ tais que o domínio dos sinais em W_{j+1} é igual ao domínio dos sinais em V_j , para $j \geq 0$. Suponha também que para cada $j \geq 0$ temos operadores $\psi_j^\uparrow : V_j \rightarrow V_{j+1}$, $\omega_j^\uparrow : V_j \rightarrow W_{j+1}$ e $\Psi_j^\downarrow : V_{j+1} \times W_{j+1} \rightarrow V_j$. Estes operadores devem ser tais que $\Psi_j^\downarrow(x, y) = \psi_j^\downarrow(x) + y$, para $x \in V_{j+1}$ e

$y \in W_{j+1}$, e $\psi_j^\downarrow : V_{j+1} \rightarrow V_j$ é um operador escolhido de modo que a condição de reconstrução perfeita seja satisfeita: $\Psi_j^\downarrow(\psi_j^\uparrow(x), \omega_j^\uparrow(x)) = x$, para $x \in V_j$; ou seja, $\psi_j^\downarrow \psi_j^\uparrow(x) + \omega_j^\uparrow(x) = x$. Então¹, $\omega_j^\uparrow(x) = x - \psi_j^\downarrow \psi_j^\uparrow(x)$.

Os operadores $\psi_j^\uparrow$ e $\omega_j^\uparrow$ são chamados de *operadores de análise*, e os operadores $\Psi_j^\downarrow$ e $\psi_j^\downarrow$ são denominados *operadores de síntese*². Um sinal $x_j \in V_j$ pode ser decomposto em sinais $x_{j+1} \in V_{j+1}$ e $y_{j+1} \in W_{j+1}$ através da aplicação dos operadores de análise, e a condição de reconstrução perfeita garante que o sinal original x_j pode ser reconstruído sem perda de informação a partir dos sinais x_{j+1} e y_{j+1} usando os operadores de síntese. Podemos interpretar x_{j+1} como uma aproximação ou simplificação do sinal x_j , de modo que x_{j+1} herda muitas das propriedades de x_j . O sinal y_{j+1} pode ser visto como um sinal de detalhe ou erro, que contém (pelo menos) a informação descartada para obter tal simplificação. O sinal de detalhe é necessário para obtermos a reconstrução perfeita de x_j , pois a transformação de x_j para x_{j+1} em geral implica em perda de informação, o que faz com que a operação $\psi_j^\downarrow(x_{j+1})$ resulte apenas em uma aproximação de x_j em V_j , denotada por $\hat{x}_j$.

A decomposição de um sinal de entrada $x_0 \in V_0$ em diversas resoluções é dada por:

$$x_{j+1} = \psi_j^\uparrow(x_j) \in V_{j+1} \quad (3.2)$$

$$y_{j+1} = x_j - \psi_j^\downarrow(x_{j+1}) \in W_{j+1} \quad (3.3)$$

com $j = 0, 1, \dots, k-1$. Tal processo é denominado *transformação pirâmide* de x_0 . A decomposição pode ser feita recursivamente:

$$x_0 \rightarrow \{x_1, y_1\} \rightarrow \{x_2, y_2, y_1\} \rightarrow \dots \rightarrow \{x_k, y_k, y_{k-1}, \dots, y_1\} \quad (3.4)$$

A reconstrução perfeita do sinal x_0 a partir dos sinais x_k e $y_1, y_2, \dots, y_k$ é dada pelo seguinte esquema recursivo de síntese:

$$x_j = \Psi_j^\downarrow(x_{j+1}, y_{j+1}) = \psi_j^\downarrow(x_{j+1}) + y_{j+1}, \quad j = k-1, k-2, \dots, 0 \quad (3.5)$$

Tal processo é denominado *transformação pirâmide inversa*. A Figura 3.8 mostra de forma esquemática três níveis da transformação pirâmide e de sua inversa.

O processo de *amostragem* de uma imagem consiste em gerar uma nova imagem composta por um

¹Em decomposições piramidais, como os domínios W_{j+1} e V_j coincidem, e o operador de análise $\omega_j^\uparrow$ é escrito em termos dos operadores $\psi_j^\downarrow$ e $\psi_j^\uparrow$, pode-se definir a decomposição sem fazer uso dos espaços W_{j+1} e dos operadores de análise $\omega_j^\uparrow$. No entanto, mantemos a notação do artigo [27], que considera as pirâmides no contexto mais amplo de esquemas de decomposição com reconstrução perfeita, que também inclui as *wavelets* – neste caso, os elementos citados são necessários.

²Apesar de muitos trabalhos na área de processamento de sinais utilizarem o símbolo $\downarrow$ para denotar subamostragem (*downsampling*) e o símbolo $\uparrow$ para denotar interpolação (*upsampling*), neste texto manteremos a notação de Goutsias e Heijmans [24], onde $\uparrow$ denota uma operação de análise e tipicamente envolve *downsampling*, enquanto $\downarrow$ denota uma operação de síntese e tipicamente envolve *upsampling*. Intuitivamente, podemos entender esta notação como “ir de baixo para cima” na pirâmide no caso da operação de análise, e “ir de cima para baixo” no caso da operação de síntese.

Figura 3.8: a) transformação pirâmide; b) transformação pirâmide inversa.

subconjunto dos pixels da imagem original. Na prática, o tipo mais utilizado de amostragem substitui os pixels $(2m, 2n)$, $(2m + 1, 2n)$, $(2m, 2n + 1)$ e $(2m + 1, 2n + 1)$ da imagem original por um único pixel (m, n) na imagem de saída. Tal processo é conhecido como *amostragem diádica* [28]. Assim, o número de pixels da imagem resultante é aproximadamente um quarto do número de pixels da imagem original.

A formulação tradicional de pirâmides de imagens consiste em, a cada nível da pirâmide, realizar uma filtragem passa-baixas seguida de amostragem diádica [9]. A maioria dos esquemas piramidais propostos também usa amostragem diádica, e algoritmos multirresolução podem tirar proveito do fato de que o volume de dados é reduzido em cada nível da pirâmide, permitindo que implementações eficientes possam ser realizadas [9]. Porém, vale ressaltar que algumas decomposições que realizam amostragem de outras maneiras, ou mesmo não incluem amostragem também podem ser modeladas de acordo com a teoria apresentada por Goutsias e Heijmans [23].

3.3.1 Nota Sobre os Operadores de Adição e Subtração

A escolha dos operadores de adição e subtração entre sinais depende da aplicação que temos em mãos [23]. Abaixo, mostramos duas alternativas em que a condição de reconstrução perfeita é válida. Nos dois casos, supomos que os sinais estão em $\mathcal{T}^E$, para algum conjunto de níveis de cinza $\mathcal{T}$. Desta forma, é suficiente definir operações de adição e subtração em $\mathcal{T}$.

1. Suponha que $\mathcal{T} \subseteq \mathbb{R}$ e seja $\mathcal{T}' = \{t - s \mid t, s \in \mathcal{T}\}$. Definimos o operador de subtração $(t, s) \rightarrow t - s$ de $\mathcal{T} \times \mathcal{T} \rightarrow \mathcal{T}'$, e o operador de adição como a adição usual.
2. Suponha que $\mathcal{T} = \{0, 1, \dots, N - 1\}$. Defina as operações de adição e subtração como a adição e a subtração no grupo abeliano $\mathbb{Z}_N$, isto é, a soma e a subtração *módulo* N . Note que no caso em que $\mathcal{T} = \{0, 1\}$ (imagens binárias), as operações de adição e subtração correspondem ao operador “ou exclusivo” (XOR).

3.3.2 Uso no Projeto de Operadores Multirresolução

A teoria exposta nesta seção trata de decomposições multirresolução de imagens. Propomos a sua utilização no projeto de operadores da seguinte forma: usar os operadores de análise como mapeamentos de resolução entre janelas; muitos destes operadores são da forma $\sigma\beta_B$, onde β_B é um filtro que depende de uma vizinhança B e σ é um operador de amostragem (apresentamos alguns exemplos no Apêndice A). Assim, para mapear um padrão de uma resolução maior em um padrão de uma resolução menor, basta aplicar o filtro β_B ao padrão e então amostrar o resultado usando σ . Porém, deve-se ter o cuidado de que, para cada ponto em uma janela W_i , $i > 0$, os pontos correspondentes à vizinhança B estejam presentes na janela W_{i-1} , para que o mapeamento seja uma função válida. A Figura 3.9 mostra uma seqüência válida de três janelas, para quando a vizinhança B é o quadrado 3×3 e o operador σ é a amostragem diádica.

Figura 3.9: Pirâmide válida para mapeamentos dependentes de uma vizinhança 3×3 . As janelas são compostas pelos pontos escuros.

3.4 Comentários

Neste capítulo, descrevemos o estado da arte na área de projeto multirresolução de W -operadores, assunto de alguns trabalhos recentes [1, 6]. A técnica depende da escolha de uma pirâmide de janelas, que exerce influência direta sobre a qualidade do operador projetado. Tal estrutura é determinada pelo especialista em processamento de imagens de maneira *ad-hoc*, com base em seu conhecimento prévio sobre a aplicação em questão. Vimos também que o número de possíveis mapeamentos de resolução é muito grande. Tendo em vista esta limitação, o foco de nossa pesquisa foi no desenvolvimento de uma técnica de projeto de operadores similar ao projeto multirresolução, mas que auxilie o projetista na escolha da pirâmide utilizada. A idéia é que, a partir de uma coleção de pirâmides, a técnica utilize algum critério para determinar automaticamente a partir dos exemplos uma boa pirâmide para a aplicação em questão. Nos próximos capítulos, descreveremos os detalhes da técnica desenvolvida e dos experimentos com a entropia condicional como critério na escolha da pirâmide.

Capítulo 4

Estimação da Distribuição Conjunta

No projeto de W -operadores multirresolução [1], o espaço de busca do operador é delimitado de acordo com a restrição de resolução. Implicitamente, as distribuições condicionais $p(Y | \mathbf{x})$, para $\mathbf{x} \in D_0$ são estimadas usando as partições aninhadas, induzidas por uma pirâmide de janelas. Para estimar $p(Y | \mathbf{x})$ para uma configuração $\mathbf{x} \in D_0$ é feita uma análise multirresolução. Esta consiste em realizar a estimação usando amostras na maior resolução para a qual temos observações suficientes de $\mathbf{x}$ nos exemplos, isto é, se não temos uma boa estimativa de $p(Y | \mathbf{x})$ na resolução de D_0 , o mapeamento de resolução ρ_{01} é aplicado a $\mathbf{x}$ e a estimativa na resolução de D_1 é verificada. Se ainda não temos uma boa estimativa, ρ_{12} é aplicado a $\rho_{01}(\mathbf{x})$, e assim por diante.

Uma das contribuições deste trabalho foi o desenvolvimento de um algoritmo para estimar a distribuição conjunta $p(\mathbf{X}, Y)$. Para tal, além de estimar $p(Y | \mathbf{x})$, para $\mathbf{x} \in D_0$, o algoritmo também realiza a estimação de $p(\mathbf{X})$ usando o arcabouço multirresolução dado por uma pirâmide. Veremos no Capítulo 5 que a estimativa de $p(\mathbf{X})$ será utilizada no cálculo da entropia condicional. Neste capítulo, mostramos o algoritmo e apresentamos uma breve discussão comparando-o com a metodologia de [1].

4.1 Algoritmo

Antes de descrever o algoritmo de estimação, vamos introduzir a notação que será utilizada. Seja W uma janela finita e $p(\mathbf{X})$ uma distribuição de probabilidade em D_0 . Supomos que existe uma partição¹

¹Como veremos, esta partição será determinada pelo algoritmo de estimação. Não se deve confundir os conjuntos $\mathcal{X}_i$ desta partição com as partições $\mathcal{X}^0, \mathcal{X}^1, \dots, \mathcal{X}^r$ resultantes do particionamento progressivo do espaço D_0 usando uma pirâmide, introduzido no Capítulo 3.

$\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n\}$ de D_0 e uma distribuição de probabilidade $\gamma(\mathcal{X})$ em $\mathcal{X}$ tais que²

$$\forall \mathbf{x}_i \in \mathcal{X}_i, \quad P(Y | \mathbf{x}_i) = P(Y | \mathcal{X}_i), \quad (4.1)$$

e

$$\Gamma(\mathcal{X}_i) = \sum_{\mathbf{x} \in \mathcal{X}_i} P(\mathbf{x}), \quad (4.2)$$

para todo i em $\{1, \dots, n\}$. A Equação 4.1 significa que todas as configurações em uma parte $\mathcal{X}_i$ de $\mathcal{X}$ têm a mesma distribuição condicional $p(Y | \mathcal{X}_i)$, e a Equação 4.2 determina que a probabilidade de um conjunto $\mathcal{X}_i$ é a soma das probabilidades de cada configuração no conjunto.

Seja $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)$ uma amostra de $p(\mathbf{X}, Y)$ (obtida a partir dos pares de imagens de entrada e saída), e seja Δ uma pirâmide de janelas. Denotaremos as partições aninhadas induzidas por Δ , em ordem decrescente de resolução, por $\mathcal{X}^0 = \{\mathcal{X}_1^0, \dots, \mathcal{X}_{n_0}^0\}, \dots, \mathcal{X}^r = \{\mathcal{X}_1^r, \dots, \mathcal{X}_{n_r}^r\}$. Agora consideremos um subconjunto não vazio $\mathcal{S}$ de D_0 . Definimos

$$N_{\mathcal{S}} = \sum_{k=1}^m c_{\mathcal{S}}(\mathbf{x}_k), \quad (4.3)$$

onde

$$c_{\mathcal{S}}(\mathbf{x}) = \begin{cases} 1, & \text{se } \mathbf{x} \in \mathcal{S} \\ 0, & \text{caso contrário.} \end{cases} \quad (4.4)$$

Ou seja, $N_{\mathcal{S}}$ é o número total de vezes que as configurações em $\mathcal{S}$ aparecem nos pares de exemplo.

Para $\mathcal{S}$ em $\mathcal{X}$, a estimação de $p(Y | \mathcal{S})$ é dada por

$$P(Y | \mathcal{S}) = \frac{\sum_{k=1}^m a_{\mathcal{S}}((\mathbf{x}_k, y_k))}{N_{\mathcal{S}}}, \quad (4.5)$$

onde

$$a_{\mathcal{S}}((\mathbf{x}, y)) = \begin{cases} 1, & \text{se } \mathbf{x} \in \mathcal{S} \text{ e } Y = y \\ 0, & \text{caso contrário.} \end{cases} \quad (4.6)$$

Isto significa que a distribuição de probabilidade condicional para $\mathcal{S}$ é calculada simplesmente tomando a distribuição de Y para as configurações em $\mathcal{S}$.

Para estimar $p(\mathbf{X})$, é necessário atribuir massas de probabilidade às configurações em D_0 . Nosso algoritmo não o faz para todas as configurações em D_0 ; isto é feito para grupos de configurações (os

²Assim como $p(\cdot)$ e $P(\cdot)$ denotam uma distribuição de probabilidade e um valor de massa de probabilidade, usaremos $\gamma(\mathcal{X})$ e $\Gamma(\mathcal{X}_i)$ para denotar a distribuição sobre $\mathcal{X}$ e o valor da probabilidade de um determinado conjunto $\mathcal{X}_i$.

conjuntos em $\mathcal{X}$), ou seja, estimaremos $\gamma(\mathcal{X})$. Vamos então definir uma função que será utilizada pelo algoritmo de estimação. Para um subconjunto não vazio $\mathcal{S}$ de D_0 , definimos

$$M_{\mathcal{S}} = \frac{T_{\alpha}(N_{\mathcal{S}})}{\sum_{k=1}^{n_r} T_{\alpha}(N_{\mathcal{X}_k^r})}, \quad (4.7)$$

onde

$$T_{\alpha}(n) = \begin{cases} n, & \text{se } n \geq \alpha \\ 0, & \text{caso contrário.} \end{cases} \quad (4.8)$$

O número $\alpha \in \{2, 3, 4, \dots\}$ é um parâmetro para o algoritmo, que serve para, durante o processo de estimação, em um certo nível de resolução i levarmos em conta apenas os conjuntos $\mathcal{X}_k^i$ de $\mathcal{X}^i$ tais que o número de observações em $\mathcal{X}_k^i$ é maior ou igual a α . Conforme vamos da resolução 0 até a resolução r , a partição $\mathcal{X}^i$ torna-se cada vez mais grosseira, logo o número de elementos nos conjuntos $\mathcal{X}_k^i$ tende a aumentar.

Vamos agora mostrar um algoritmo para estimar $\gamma(\mathcal{X})$ e $p(Y | \mathcal{X})$. Ele recebe α , Δ e $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)$ como entradas, e devolve um conjunto $\mathcal{R}$ de triplas no formato $(\mathcal{X}_i, \Gamma(\mathcal{X}_i), p(Y | \mathcal{X}_i))$, isto é, cada tripla contém um conjunto $\mathcal{X}_i$ de $\mathcal{X}$, sua massa de probabilidade $\Gamma(\mathcal{X}_i)$ e a distribuição condicional $p(Y | \mathcal{X}_i)$. O conjunto $\mathcal{R}$ é inicializado como um conjunto vazio, e é construído durante a execução do algoritmo de forma que, ao final desta, $\mathcal{R}$ é uma representação da distribuição conjunta, e os conjuntos $\mathcal{X}_i$ que aparecem nas triplas de $\mathcal{R}$ são os conjuntos de $\mathcal{X}$ tais que $\Gamma(\mathcal{X}_i) > 0$. Trabalharemos sob a hipótese de que $\sum_{k=1}^{n_r} T_{\alpha}(N_{\mathcal{X}_k^r}) > 0$, o que significa que a partição $\mathcal{X}^r$ tem pelo menos um conjunto $\mathcal{X}_i^r$ cujos elementos aparecem pelo menos α vezes nas amostras.

ESTIMAÇÃO-DA-DISTRIBUIÇÃO-CONJUNTA($\alpha, \Delta, (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)$)

01. $\mathcal{R}$ = conjunto vazio;
02. PARA CADA conjunto $\mathcal{X}_i^0, i \in \{1, \dots, n_0\}$ FAÇA
03. SE $M_{\mathcal{X}_i^0} > 0$, ENTÃO
04. $\hat{p} = p(Y | \mathcal{X}_i^0)$ calculada como na Equação 4.5;
05. $\mathcal{R} = \mathcal{R} \cup \{(\mathcal{X}_i^0, M_{\mathcal{X}_i^0}, \hat{p})\};$
06. PARA CADA resolução j de 1 até r FAÇA
07. PARA CADA conjunto $\mathcal{X}_i^j, i \in \{1, \dots, n_j\}$ FAÇA
08. $\mathcal{K} = \{\mathcal{Z} : \text{existe uma tripla } (\mathcal{Z}, \cdot, \cdot) \in \mathcal{R} \text{ e } \mathcal{Z} \subseteq \mathcal{X}_i^j\};$
09. $M = M_{\mathcal{X}_i^j} - \sum_{K \in \mathcal{K}} \Gamma(K);$
10. SE $M > 0$, ENTÃO
11. $\hat{p} = p(Y | \mathcal{X}_i^j)$ calculada como na Equação 4.5;
12. $\mathcal{U} = \bigcup_{K \in \mathcal{K}} K;$
13. $\mathcal{R} = \mathcal{R} \cup \{(\mathcal{X}_i^j - \mathcal{U}, M, \hat{p})\};$
14. DEVOLVA $\mathcal{R}$.

Nos passos 02 a 05, é processada a primeira resolução, isto é, a resolução da janela da base da pirâmide. Para cada conjunto $\mathcal{X}_i^0$, $i \in \{1, \dots, n_0\}$, de massa $M_{\mathcal{X}_i^0}$ positiva, estimamos a distribuição condicional com base na distribuição dos valores de saída, usamos a massa $M_{\mathcal{X}_i^0}$ como estimativa de $\Gamma(\mathcal{X}_i^0)$ e adicionamos uma tripla a $\mathcal{R}$.

Em seguida, nos passos 06 a 13 processamos as demais resoluções iterativamente. Para cada resolução j de 1 a r , para cada conjunto $\mathcal{X}_i^j$ calculamos a massa M descontando de $M_{\mathcal{X}_i^j}$ a massa dos conjuntos em $\mathcal{K}$, isto é, as massas dos subconjuntos de $\mathcal{X}_i^j$ que já foram atribuídas em resoluções anteriores não são contabilizadas. Isto garante que, ao final do algoritmo, a soma das massas de probabilidade atribuídas aos conjuntos que aparecem nas triplas de $\mathcal{R}$ seja igual a 1. Se a massa M for positiva, uma tripla representando o conjunto $\mathcal{X}_i^j - \mathcal{U}$, com massa M e distribuição condicional calculada com base na distribuição dos valores de saída em $\mathcal{X}_i^j$ é adicionada a $\mathcal{R}$.

Ao final do algoritmo, $\mathcal{R}$ é uma representação da distribuição conjunta, pois cada tripla representa um dos conjuntos $\mathcal{X}_i$ de $\mathcal{X}$ tais que $\Gamma(\mathcal{X}_i) > 0$, com sua distribuição de probabilidade condicional associada e o valor de $\Gamma(\mathcal{X}_i)$. Uma observação importante é que, como dito anteriormente, estamos estimando $\gamma(\mathcal{X})$, mas não a distribuição completa $p(\mathbf{X})$. No próximo capítulo, veremos que para a aplicação considerada neste trabalho a estimação de $\gamma(\mathcal{X})$ é suficiente. Mas se os valores de $P(\mathbf{x})$ forem necessários em uma aplicação particular, pode-se considerar qualquer modelo de probabilidade que satisfaça a Equação 4.2.

4.2 Exemplo

Com o objetivo de mostrar de forma mais concreta a dinâmica de funcionamento do algoritmo, vamos agora apresentar a sua execução sobre um conjunto de amostras fictício. Novamente utilizaremos uma janela de apenas três pontos e um operador binário para facilitar a compreensão. Mesmo que em aplicações reais seja muito difícil termos configurações não observadas nas amostras de treinamento com uma janela deste tamanho, neste exemplo simularemos este fato para mostrar como é feita a atribuição de probabilidades para padrões não observados. Vamos supor que o parâmetro α é igual a 2.

Consideremos a pirâmide de janelas do exemplo de particionamento da Seção 3.1, dada pelas janelas $W_0 = \{w_1, w_2, w_3\}$, $W_1 = \{w_2, w_3\}$ e $W_2 = \{w_3\}$ e pelos mapeamentos de resolução definidos por simples subamostragem. A Figura 4.1 recorda as partições aninhadas geradas pela pirâmide e mostra a divisão das partições em conjuntos.

W_0 □□□	W_1 □□□	W_2 □□□
100 000	100 000	100 000
010 110	010 110	010 110
001 101	001 101	001 101
011 111	011 111	011 111
$\mathcal{X}^0$	$\mathcal{X}^1$	$\mathcal{X}^2$
$\mathcal{X}_1^0$ $\mathcal{X}_2^0$	$\mathcal{X}_1^1$	$\mathcal{X}_1^2$
$\mathcal{X}_3^0$ $\mathcal{X}_4^0$	$\mathcal{X}_2^1$	
$\mathcal{X}_5^0$ $\mathcal{X}_6^0$	$\mathcal{X}_3^1$	$\mathcal{X}_2^2$
$\mathcal{X}_7^0$ $\mathcal{X}_8^0$	$\mathcal{X}_4^1$	

Figura 4.1: Partições aninhadas.

Vamos supor que, a partir das imagens de treinamento, foram observados os seguintes padrões na resolução de W_0 (novamente, f_0 é o número de vezes em que um padrão apareceu associado ao valor de saída 0, e f_1 é o número de vezes em que um padrão apareceu associado ao valor de saída 1):

$\mathbf{x}$	f_0	f_1	Total
000	0	0	0
100	1	0	1
010	0	0	0
110	3	1	4
001	0	1	1
101	3	2	5
011	1	0	1
111	0	2	2

Tabela 4.1: Tabela de exemplos - resolução de W_0 .

A Figura 4.2 mostra a partição $\mathcal{X}^0$ e suas respectivas observações. Os círculos representam padrões com valor de saída 0, enquanto os quadrados representam padrões com valores de saída 1.

●	
	● ● ● ■
■	● ● ● ■ ■
●	■ ■

Figura 4.2: Partição $\mathcal{X}^0$.

A partição $\mathcal{X}$ é construída conforme a execução do algoritmo progride, e inicialmente está vazia (passo 01). A construção é feita progressivamente, através da inclusão de triplas no conjunto $\mathcal{R}$. Apresentaremos o estado da partição $\mathcal{X}$ conforme a execução do algoritmo progride, na forma de diagramas em que, para cada conjunto $\mathcal{X}_i$, temos as amostras usadas na estimação de sua distribuição de probabilidade condicional, e o valor da probabilidade $\Gamma(\mathcal{X}_i)$.

No passo 02, o algoritmo processa a primeira resolução, examinando a partição $\mathcal{X}^0$. A condição do passo 03 é verdadeira quando $i = 4, 6$ e 8 ; os valores de $M_{\mathcal{X}_i^0}$ para cada um dos conjuntos são, respectivamente, $4/14$, $5/14$ e $2/14$. Assim, para cada execução do passo 04, uma das seguintes distribuições condicionais é estimada:

- $P(Y = 0 | \mathcal{X}_4^0) = 3/4$, $P(Y = 1 | \mathcal{X}_4^0) = 1/4$;
- $P(Y = 0 | \mathcal{X}_6^0) = 3/5$, $P(Y = 1 | \mathcal{X}_6^0) = 2/5$;
- $P(Y = 0 | \mathcal{X}_8^0) = 0$, $P(Y = 1 | \mathcal{X}_8^0) = 1$.

Em cada execução do passo 05, uma das seguintes triplas é adicionada a $\mathcal{R}$:

- $(\mathcal{X}_1 = \mathcal{X}_4^0 = \{110\}, \Gamma(\mathcal{X}_1) = 4/14, [P(Y = 0 | \mathcal{X}_1) = 3/4, P(Y = 1 | \mathcal{X}_1) = 1/4])$;
- $(\mathcal{X}_2 = \mathcal{X}_6^0 = \{101\}, \Gamma(\mathcal{X}_2) = 5/14, [P(Y = 0 | \mathcal{X}_2) = 3/5, P(Y = 1 | \mathcal{X}_2) = 2/5])$;
- $(\mathcal{X}_3 = \mathcal{X}_8^0 = \{111\}, \Gamma(\mathcal{X}_3) = 2/14, [P(Y = 0 | \mathcal{X}_3) = 0, P(Y = 1 | \mathcal{X}_3) = 1])$.

A Figura 4.3 mostra o diagrama da partição $\mathcal{X}$ com os conjuntos adicionados. Na região de cada conjunto estão representadas as amostras que são usadas na estimação da distribuição condicional, e o valor de $\Gamma(\mathcal{X})$.

Figura 4.3: Partição $\mathcal{X}$ - após o processamento da primeira resolução.

Em seguida, no passo 06 processa-se as demais resoluções. Quando $j = 1$, a distribuição dos padrões é dada pela Tabela 4.2, e a distribuição das amostras na partição $\mathcal{X}^1$ pode ser vista na Figura 4.4.

$\mathbf{x}$	f_0	f_1	Total
00	1	0	1
10	3	1	4
01	3	3	6
11	1	2	3

Tabela 4.2: Tabela de exemplos - resolução de W_1 . Figura 4.4: Partição $\mathcal{X}^1$.

Os passos 08 e 09 são repetidos para os conjuntos $\mathcal{X}_1^1$, $\mathcal{X}_2^1$, $\mathcal{X}_3^1$ e $\mathcal{X}_4^1$. Para o conjunto $\mathcal{X}_1^1 = C[00] = \{100, 000\}$, no passo 08 o conjunto $\mathcal{K}$ é vazio, e no passo 09, $M = 0$. Portanto, a condição do passo 10 é falsa e os passos 11, 12 e 13 não são executados. Para o conjunto $\mathcal{X}_2^1 = C[10] = \{010, 110\}$, temos $\mathcal{K} = \{\mathcal{X}_1\}$ no passo 08, e no passo 09 a massa $M = 4/14 - \Gamma(\mathcal{X}_1) = 0$. A massa M é igual a zero porque estamos descontando a massa do subconjunto $\mathcal{X}_1$, que já recebeu massa quando processamos a resolução anterior. Logo, a condição do passo 10 é falsa e os demais passos também não são executados para o conjunto em questão. Já para o conjunto $\mathcal{X}_3^1 = C[01] = \{001, 101\}$, no passo 08 obtemos $\mathcal{K} = \{\mathcal{X}_2\}$, e no passo 09 calculamos $M = 6/14 - \Gamma(\mathcal{X}_2) = 1/14$. A condição no passo 10 é verdadeira, e calcula-se a seguinte distribuição condicional no passo 11:

$$P(Y = 0 \mid \mathcal{X}_3^1) = 1/2, P(Y = 1 \mid \mathcal{X}_3^1) = 1/2.$$

No passo 12, $\mathcal{U} = \mathcal{X}_2$, e no passo 13 adicionamos a seguinte tripla a $\mathcal{R}$:

$$(\mathcal{X}_4 = \mathcal{X}_3^1 - \mathcal{X}_2 = \{001\}, \Gamma(\mathcal{X}_4) = 1/14, [P(Y = 0 \mid \mathcal{X}_4) = 1/2, P(Y = 1 \mid \mathcal{X}_4) = 1/2]).$$

Finalmente, para o conjunto $\mathcal{X}_4^1 = C[11] = \{011, 111\}$, no passo 08 obtemos $\mathcal{K} = \{\mathcal{X}_3\}$ e no passo 09 calculamos $M = 3/14 - \Gamma(\mathcal{X}_3) = 1/14$. A condição no passo 10 é verdadeira, e estima-se a seguinte distribuição condicional no passo 11:

$$P(Y = 0 \mid \mathcal{X}_4^1) = 1/3, P(Y = 1 \mid \mathcal{X}_4^1) = 2/3.$$

No passo 12, $\mathcal{U} = \mathcal{X}_3$, e no passo 13 adicionamos a seguinte tripla a $\mathcal{R}$:

$$(\mathcal{X}_5 = \mathcal{X}_4^1 - \mathcal{X}_3 = \{011\}, \Gamma(\mathcal{X}_5) = 1/14, [P(Y = 0 | \mathcal{X}_5) = 1/3, P(Y = 1 | \mathcal{X}_5) = 2/3]).$$

A Figura 4.5 mostra o diagrama de $\mathcal{X}$ após esta iteração.

Figura 4.5: Partição $\mathcal{X}$ - após o processamento da segunda resolução.

Por fim, quando $j = 2$ no passo 06, temos as amostras da Tabela 4.3, e a partição $\mathcal{X}^2$, representada na Figura 4.6:

$\mathbf{x}$	f_0	f_1	Total
0	4	1	5
1	4	5	9

Tabela 4.3: Tabela de exemplos - resolução de W_2 .

Figura 4.6: Partição $\mathcal{X}^2$.

Os passos 08 e 09 são repetidos para os conjuntos $\mathcal{X}_1^2$ e $\mathcal{X}_2^2$. Para o conjunto $\mathcal{X}_1^2 = C[0] = \{000, 010, 100, 110\}$, no passo 08 o conjunto $\mathcal{K}$ é igual a $\{\mathcal{X}_1\}$, e, no passo 09, M é igual a $5/14 - \Gamma(\mathcal{X}_1) = 1/14$. Portanto, no passo 10 a condição é verdadeira, e no passo 11 a distribuição condicional

$$P(Y = 0 | \mathcal{X}_1^2) = 4/5, P(Y = 1 | \mathcal{X}_1^2) = 1/5$$

é estimada. No passo 12, $\mathcal{U} = \mathcal{X}_1$, e no passo 13 a tripla

$$(\mathcal{X}_6 = \mathcal{X}_1^2 - \mathcal{X}_1 = \{000, 100, 010\}, \Gamma(\mathcal{X}_6) = 1/14, [P(Y = 0 | \mathcal{X}_6) = 4/5, P(Y = 1 | \mathcal{X}_6) = 1/5])$$

é adicionada a $\mathcal{R}$. Para o conjunto $\mathcal{X}_2^2 = C[1] = \{001, 101, 011, 111\}$, no passo 08 o conjunto $\mathcal{K}$ é igual a $\{\mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4, \mathcal{X}_5\}$, no passo 09 calcula-se $M = 9/14 - (\Gamma(\mathcal{X}_2) + \Gamma(\mathcal{X}_3) + \Gamma(\mathcal{X}_4) + \Gamma(\mathcal{X}_5)) = 0$, e no passo 10 a condição é falsa. Assim, os demais passos não são executados. Finalmente, o algoritmo termina no passo 14. A Figura 4.7 mostra o resultado final do algoritmo.

Figura 4.7: Partição $\mathcal{X}$ - fim do algoritmo.

Uma observação importante do ponto de vista de implementação é o fato de que, em cada resolução, não é necessário analisar os conjuntos para os quais não há amostras observadas. Para estes conjuntos, as condições nos passos 03 e 10 serão sempre falsas, logo não serão adicionadas triplas. Assim, basta analisar os conjuntos em que há padrões observados. No exemplo apresentado nesta seção, os conjuntos $\mathcal{X}_2^0$ e $\mathcal{X}_3^0$ não precisam ser analisados.

4.3 Algumas Considerações

Apresentamos um algoritmo para estimar a distribuição de probabilidade conjunta através da estimação de $\gamma(\mathcal{X})$ e $p(Y | \mathbf{x})$, para $\mathbf{x} \in D_0$. Embora o algoritmo utilize o mesmo arcabouço piramidal dos trabalhos anteriores, existem algumas diferenças entre o estimador que desenvolvemos e o estimador usado no projeto de operadores multirresolução de [1] no que se refere à estimação das distribuições condicionais:

- Em [1], estima-se as distribuições de probabilidade condicional $p(Y | \mathbf{x})$, para $\mathbf{x} \in D_0$. Para cada configuração $\mathbf{x} \in D_0$, sua respectiva distribuição condicional é estimada na maior resolução em que $\mathbf{x}$ for observada pelo menos α vezes nos exemplos. Já o estimador descrito neste capítulo não define distribuições condicionais $p(Y | \mathbf{x})$ para configurações $\mathbf{x}$ tais que $P(\mathbf{x}) = 0$;
- Ainda no caso do estimador apresentado neste capítulo, para as configurações $\mathbf{x}$ tais que $P(\mathbf{x}) > 0$ a estimação de $p(Y | \mathbf{x})$ não é necessariamente feita na maior resolução onde $\mathbf{x}$ for observada pelo

menos α vezes nos exemplos. A resolução dependerá dos dados de treinamento, pois a estimação só é feita quando uma tripla é incluída em $\mathcal{R}$, isto é, quando $M > 0$. O exemplo mostrado na seção anterior ilustra este fato. De acordo com o estimador deste capítulo, $P(Y = 0 | \mathbf{X} = 010) = 4/5$ e $P(Y = 1 | \mathbf{X} = 010) = 1/5$, pois esta distribuição condicional foi estimada com amostras na resolução de W_2 . Já o estimador do trabalho anterior utilizaria as amostras na resolução de W_1 para fazer a estimação, o que resultaria em $P(Y = 0 | \mathbf{X} = 010) = 3/4$ e $P(Y = 1 | \mathbf{X} = 010) = 1/4$.

4.4 Representação

Um ponto positivo da metodologia que apresentamos neste capítulo é a facilidade de representação da distribuição estimada. A idéia é utilizar diversas tabelas de configurações observadas, uma para cada resolução, e a pirâmide de janelas. Se gostaríamos de obter a distribuição condicional para um determinado padrão $\mathbf{x} \in D_0$, iniciamos buscando tal configuração na tabela de maior resolução; caso ela não seja encontrada, o mapeamento de resolução ρ_{01} é aplicado a $\mathbf{x}$ e a configuração resultante é buscada na tabela seguinte; caso ela ainda não seja encontrada, ρ_{12} é aplicado a $\rho_{01}(\mathbf{x})$ e o resultado é buscado na próxima tabela. O processo é repetido até que se encontre um padrão, ou todas as tabelas sejam examinadas (esta última situação ocorre quando $P(\mathbf{x}) = 0$).

Mais especificamente, podemos representar o conjunto $\mathcal{R}$ usando diversas tabelas $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_r$ de configurações observadas nas resoluções de $D_0, D_1, \dots, D_r$; cada entrada de uma tabela $\mathcal{T}_i$ é composta por uma configuração $\mathbf{z}_k$, sua distribuição condicional associada $p(Y | \mathcal{X}_k)$ e o valor de $\Gamma(\mathcal{X}_k)$ correspondente ao conjunto $\mathcal{X}_k$. Se $i > 0$, $\mathcal{X}_k$ é dado pelos padrões $\mathbf{x}$ de D_0 tais que $\rho_{(i-1)i} \cdots \rho_{12}\rho_{01}(\mathbf{x}) = \mathbf{z}_k$ e não existe $j \in \{0, \dots, i-1\}$ tal que $\rho_{(j-1)j} \cdots \rho_{12}\rho_{01}(\mathbf{x})$ está na tabela $\mathcal{T}_j$; se $i = 0$, $\mathcal{X}_k = \{\mathbf{z}_k\}$. Note que não armazenamos explicitamente os conjuntos $\mathcal{X}_k$ – dependendo da estrutura das partições aninhadas, enumerar tais conjuntos pode ser uma tarefa muito custosa. Nas aplicações consideradas neste trabalho não é necessário enumerá-los.

Desta forma, temos uma representação eficiente em termos de espaço, já que o maior número possível de entradas nas tabelas será o número de padrões distintos observados nos exemplos. A utilização dos mapeamentos de resolução permite obter os valores da distribuição para configurações não observadas no treinamento.

A mesma representação pode ser utilizada para o operador projetado a partir da distribuição estimada. Neste caso, a mesma estrutura de tabelas se mantém, mas cada entrada é composta apenas por uma configuração e pelo seu correspondente valor de saída. O valor de saída tipicamente é obtido utilizando um critério de decisão a partir das distribuições condicionais, como por exemplo, escolher o valor que é mais provável (critério da moda). A aplicação do operador para uma configuração $\mathbf{x}$ também é feita da mesma maneira que o acesso à distribuição condicional: inicia-se buscando a configuração em $\mathcal{T}_0$; caso ela não seja encontrada, o mapeamento de resolução ρ_{01} é aplicado a $\mathbf{x}$ e a configuração resultante é buscada

em $\mathcal{T}_1$; caso ela ainda não seja encontrada, ρ_{12} é aplicado a $\rho_{01}(\mathbf{x})$ e o resultado é buscado em $\mathcal{T}_2$. O processo é repetido até que um padrão seja encontrado (o respectivo valor de saída é atribuído ao pixel em questão na imagem de saída), ou todas as tabelas sejam examinadas (neste caso, pode-se atribuir um valor arbitrário, ou um valor fora do intervalo de possíveis valores de saída, dependendo da aplicação).

Capítulo 5

Escolha da Pirâmide

Conforme vimos até o momento, a escolha da pirâmide de janelas determina a estrutura das partições aninhadas, que exerce influência direta sobre a distribuição de probabilidade estimada e, portanto, contribui significativamente para a qualidade do operador projetado. Na prática, o uso de uma pirâmide específica pode ser uma má escolha para alguns problemas, mas pode levar a bons resultados em outros; ou seja, uma boa escolha também depende do problema que temos em mãos.

Fixada uma janela W , o número de possíveis pirâmides de janelas que têm W como base cresce muito rapidamente conforme o número de pontos em W aumenta. Assim, examinar todas as possíveis pirâmides e escolher a melhor segundo algum critério é uma tarefa computacionalmente inviável. No entanto, uma ferramenta que permita ao projetista de operadores especificar um conjunto limitado de pirâmides e uma coleção de exemplos de treinamento e, a partir desta informação, determine automaticamente uma boa pirâmide a ser usada no projeto de um W -operador seria bastante útil. O projetista poderia usar seu conhecimento prévio sobre as pirâmides ou mesmo uma biblioteca de mapeamentos de resolução conhecidos da literatura (como os da teoria de pirâmides de imagens, por exemplo) para fornecer a entrada para o programa.

Neste capítulo, descreveremos o critério da entropia condicional, cuja utilização na escolha da pirâmide foi um dos tópicos que investigamos neste trabalho. Tal critério baseia-se em conceitos de teoria da informação [29] e, recentemente, foi utilizado com sucesso no problema de redução de dimensionalidade [10, 11, 12, 13], com aplicações em bioinformática e processamento de imagens.

5.1 Entropia

Se X é uma variável aleatória discreta e $p(X)$ é a sua distribuição de probabilidade, a entropia $H(X)$ de X é definida como¹

$$H(X) = - \sum_{x \in \mathcal{A}_X} P(x) \log_2(P(x)). \quad (5.1)$$

A entropia pode ser vista como uma medida de concentração da massa de probabilidade. Se as probabilidades concentram-se em um valor $x \in \mathcal{A}_X$, então a entropia deve ser baixa. Já a distribuição de probabilidade uniforme é a que tem entropia máxima. A Figura 5.1 ilustra este fato.

Figura 5.1: Entropia: a) alta entropia; b) baixa entropia.

Se X e Y são variáveis aleatórias, a entropia condicional de Y dado que $X = x$ é a entropia da distribuição de probabilidade $p(Y | x)$:

$$H(Y | X = x) = - \sum_{y \in \mathcal{A}_Y} P(y | x) \log_2(P(y | x)). \quad (5.2)$$

Já a **entropia condicional** [30] de Y dado X é a média das entropias condicionais das distribuições $p(Y | x)$ ponderada pelas probabilidades $P(x)$, para $x \in \mathcal{A}_X$:

$$H(Y | X) = \sum_{x \in \mathcal{A}_X} P(x) \cdot H(Y | x), \quad (5.3)$$

ou seja,

$$H(Y | X) = \sum_{x \in \mathcal{A}_X} P(x) \cdot \left[- \sum_{y \in \mathcal{A}_Y} P(y | x) \log_2(P(y | x)) \right]. \quad (5.4)$$

¹No cálculo da entropia, por convenção usa-se $\log_2(0) = 0$.

Esta medida quantifica a incerteza média sobre os valores de Y quando X é conhecida.

5.2 Informação Mútua

Informação mútua é uma medida de dependência entre duas variáveis aleatórias X e Y . Ela é definida como [30]:

$$I(X; Y) = H(Y) - H(Y | X), \quad (5.5)$$

e satisfaz $I(X; Y) = I(Y; X)$, e $I(X; Y) \geq 0$. Se X e Y são independentes, $I(X; Y) = 0$.

Uma consequência imediata da Equação 5.5 é a equivalência entre minimizar a entropia condicional e maximizar a informação mútua. Conforme a dependência entre X e Y cresce, as distribuições condicionais $p(Y | x)$, $x \in \mathcal{A}_X$ tendem a possuir massa de probabilidade mais concentrada em alguns valores de Y .

5.3 Critério de Escolha

Em problemas de classificação, em geral deseja-se que as distribuições condicionais verdadeiras $p(Y | \mathbf{x})$ tenham massas de probabilidade concentradas em uma das possíveis classes $y \in \mathcal{A}_Y$. Desta forma, para cada vetor de características $\mathbf{x} \in \mathcal{A}_X$ existirá uma classe $y \in \mathcal{A}_Y$ que será mais provável, o que implica em menor incerteza na classificação. O mesmo raciocínio vale para o projeto de W -operadores: deseja-se que para cada configuração $\mathbf{x} \in \mathcal{A}_X$ exista um valor de saída $y \in \mathcal{A}_Y$ que seja mais provável. Em tal situação, a informação mútua é alta, ou, equivalentemente, a entropia condicional é baixa.

A fim de estabelecer um critério para escolher uma pirâmide de janelas, trabalharemos com a hipótese de que as distribuições condicionais verdadeiras (as que gostaríamos de estimar a partir dos exemplos) possuem massa de probabilidade concentrada em um dos valores de saída. Esta hipótese é razoável para problemas que admitem uma boa solução. Partindo deste princípio, e sabendo que cada pirâmide de janelas induz uma coleção de distribuições condicionais, propomos a busca por uma pirâmide que induza distribuições condicionais de baixa entropia. Em outras palavras, procuramos pela pirâmide que induz distribuições condicionais que, em média, proporcionam uma melhor predição do valor de saída $y \in \mathcal{A}_Y$ dado que um padrão $\mathbf{x} \in \mathcal{A}_X$ foi observado. Assim, o problema passa a ser encontrar uma pirâmide de janelas que resulte em uma distribuição de baixa entropia condicional.

5.4 Cálculo da Entropia Condicional

Como podemos ver na Equação 5.4, para calcular a entropia condicional é necessário conhecer as distribuições de probabilidade $p(\mathbf{X})$ e $p(Y | \mathbf{x})$, para $\mathbf{x} \in \mathcal{A}_X$. O cálculo será feito com base nas distribuições

estimadas pelo algoritmo da Seção 4.1.

Recordemos a partição $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n\}$, cujos elementos de probabilidades maiores do que zero são determinados pelo algoritmo de estimação. A entropia condicional de uma distribuição induzida por uma pirâmide de janelas pode ser calculada a partir de $\mathcal{X}$ por

$$\begin{aligned}
 H(Y | \mathbf{X}) &= \sum_{\mathbf{x} \in \mathcal{A}_{\mathbf{X}}} P(\mathbf{x}) \cdot H(Y | \mathbf{x}) \\
 &= \sum_{i=1}^n \sum_{\mathbf{x} \in \mathcal{X}_i} P(\mathbf{x}) \cdot H(Y | \mathbf{x}) \\
 &= \sum_{i=1}^n \left(\sum_{\mathbf{x} \in \mathcal{X}_i} P(\mathbf{x}) \right) \cdot H(Y | \mathcal{X}_i) \\
 &= \sum_{i=1}^n \Gamma(\mathcal{X}_i) \cdot H(Y | \mathcal{X}_i). \tag{5.6}
 \end{aligned}$$

As quatro igualdades mostradas acima vêm, respectivamente, da Equação 5.3, do fato de que $\mathcal{X}$ é uma partição de D_0 , da Equação 4.1 e da Equação 4.2. Como o conjunto $\mathcal{R}$ (o resultado do algoritmo de estimação) contém triplas para todos os conjuntos $\mathcal{X}_i$ tais que $\Gamma(\mathcal{X}_i) > 0$, a obtenção da entropia condicional é imediata: basta calcular, para cada tripla $(\mathcal{X}_i, \Gamma(\mathcal{X}_i), p(Y | \mathcal{X}_i))$ em $\mathcal{R}$, o valor de $\Gamma(\mathcal{X}_i) \cdot H(Y | \mathcal{X}_i) = \Gamma(\mathcal{X}_i) \cdot \left[- \sum_{y \in \mathcal{A}_Y} p(y | \mathcal{X}_i) \log_2(p(y | \mathcal{X}_i)) \right]$ e somar os resultados.

5.4.1 Um Exemplo

Vamos recordar da Seção 4.2 o resultado do exemplo de estimação. Para a instância do problema considerada, o algoritmo devolveu a distribuição conjunta representada pelas seguintes triplas:

- $(\mathcal{X}_1 = \{110\}, \Gamma(\mathcal{X}_1) = 4/14, [P(Y = 0 | \mathcal{X}_1) = 3/4, P(Y = 1 | \mathcal{X}_1) = 1/4])$;
- $(\mathcal{X}_2 = \{101\}, \Gamma(\mathcal{X}_2) = 5/14, [P(Y = 0 | \mathcal{X}_2) = 3/5, P(Y = 1 | \mathcal{X}_2) = 2/5])$;
- $(\mathcal{X}_3 = \{111\}, \Gamma(\mathcal{X}_3) = 2/14, [P(Y = 0 | \mathcal{X}_3) = 0, P(Y = 1 | \mathcal{X}_3) = 1])$;
- $(\mathcal{X}_4 = \{001\}, \Gamma(\mathcal{X}_4) = 1/14, [P(Y = 0 | \mathcal{X}_4) = 1/2, P(Y = 1 | \mathcal{X}_4) = 1/2])$;
- $(\mathcal{X}_5 = \{011\}, \Gamma(\mathcal{X}_5) = 1/14, [P(Y = 0 | \mathcal{X}_5) = 1/3, P(Y = 1 | \mathcal{X}_5) = 2/3])$;
- $(\mathcal{X}_6 = \{000, 100, 010\}, \Gamma(\mathcal{X}_6) = 1/14, [P(Y = 0 | \mathcal{X}_6) = 4/5, P(Y = 1 | \mathcal{X}_6) = 1/5])$.

Para calcular a entropia condicional para tal distribuição, basta obtermos o valor de

$$H(Y | \mathbf{X}) = \sum_{i=1}^6 \Gamma(\mathcal{X}_i) \cdot H(Y | \mathcal{X}_i).$$

No exemplo,

- $H(Y | \mathcal{X}_1) = -[P(Y = 0 | \mathcal{X}_1) \cdot \log_2(P(Y = 0 | \mathcal{X}_1)) + P(Y = 1 | \mathcal{X}_1) \cdot \log_2(P(Y = 1 | \mathcal{X}_1))] = -(3/4 \cdot \log_2(3/4) + 1/4 \cdot \log_2(1/4)) = 0,81;$
- $H(Y | \mathcal{X}_2) = -[P(Y = 0 | \mathcal{X}_2) \cdot \log_2(P(Y = 0 | \mathcal{X}_2)) + P(Y = 1 | \mathcal{X}_2) \cdot \log_2(P(Y = 1 | \mathcal{X}_2))] = -(3/5 \cdot \log_2(3/5) + 2/5 \cdot \log_2(2/5)) = 0,97;$
- $H(Y | \mathcal{X}_3) = 0$ (pois um dos valores de $\mathcal{A}_Y$ tem probabilidade 1);
- $H(Y | \mathcal{X}_4) = 1$ (distribuição uniforme, a entropia é igual a $\log_2(|\mathcal{A}_Y|)$);
- $H(Y | \mathcal{X}_5) = -[P(Y = 0 | \mathcal{X}_5) \cdot \log_2(P(Y = 0 | \mathcal{X}_5)) + P(Y = 1 | \mathcal{X}_5) \cdot \log_2(P(Y = 1 | \mathcal{X}_5))] = -(1/3 \cdot \log_2(1/3) + 2/3 \cdot \log_2(2/3)) = 0,91;$
- $H(Y | \mathcal{X}_6) = -[P(Y = 0 | \mathcal{X}_6) \cdot \log_2(P(Y = 0 | \mathcal{X}_6)) + P(Y = 1 | \mathcal{X}_6) \cdot \log_2(P(Y = 1 | \mathcal{X}_6))] = -(4/5 \cdot \log_2(4/5) + 1/5 \cdot \log_2(1/5)) = 0,72.$

Portanto, $H(Y | \mathbf{X}) = 4/14 \cdot 0,81 + 5/14 \cdot 0,97 + 2/14 \cdot 0 + 1/14 \cdot 1 + 1/14 \cdot 0,91 + 1/14 \cdot 0,72 = 0,76.$

Capítulo 6

Implementações

Como parte do projeto de mestrado, foram realizadas diversas implementações de *software*. Tais implementações foram feitas como extensões ao sistema PAC [31, 32], desenvolvido por outros membros do grupo de Processamento de Imagens do IME-USP sobre o sistema KHOROS [33]. O sistema PAC provê uma biblioteca e um conjunto de programas para projetar operadores morfológicos a partir de exemplos.

O módulo de projeto multirresolução da PAC [7, 8] permitia apenas a especificação de mapeamentos de resolução dados por simples subamostragem espacial e restrição no intervalo de níveis de cinza [6]. Inicialmente, foi realizada a extensão deste módulo para que fosse possível utilizar mapeamentos da forma $\rho = \sigma\beta_B$, onde β_B é um filtro dependente de uma vizinhança B e σ é a subamostragem q -ádica¹. Após esta modificação, implementamos vários dos mapeamentos da teoria de pirâmides de imagens (ver o Apêndice A). Com isso, pudemos realizar experimentos com mapeamentos de resolução baseados em diversos filtros.

A PAC também foi usada como base para desenvolvermos um software de projeto de operadores morfológicos que implementa a metodologia desenvolvida neste trabalho. Neste capítulo, descreveremos brevemente a especificação do software desenvolvido.

6.1 Especificação do Sistema

Podemos dividir a funcionalidade do *software* implementado em duas partes: a de **projeto** de operadores e a de **aplicação** de um operador. O projeto de operadores é composto pelas seguintes etapas: **coleta de exemplos**, **estimação da distribuição conjunta** e uso de um critério de **decisão** para determinar o valor de saída do operador. A partir da distribuição conjunta estimada pode-se também **calcular a entropia condicional**. Já a aplicação de um operador consiste em, a partir da representação de um

¹Subamostragem da forma $q : 1$, ou seja, seleciona-se um pixel de cada q pixels, tanto na horizontal quanto na vertical (por exemplo, a subamostragem diádica é da forma $2 : 1$).

operador projetado anteriormente, aplicá-lo a uma imagem. A Figura 6.1 mostra a estrutura básica do sistema de forma esquemática.

Figura 6.1: Estrutura básica do sistema.

6.2 Coleta de Exemplos

Para projetar um operador, primeiramente é necessário extrair pares de entrada e saída a partir das imagens de treinamento. Esta etapa utiliza os programas de coleta de exemplos do sistema PAC, que recebem como entrada um conjunto de pares de imagens de entrada e saída e uma janela W , e devolvem uma tabela de exemplos observados através da translação da janela aos diversos pixels das imagens. No nosso programa, W é a janela da base da pirâmide. Opcionalmente, pode-se fornecer uma imagem máscara que determina um subconjunto dos pixels das imagens. A coleta é então realizada apenas para as janelas transladadas com centro em um dos pixels do subconjunto.

6.3 Estimação da Distribuição Conjunta

A partir da tabela de exemplos, é feita a estimação da distribuição de probabilidade conjunta. Para isto, foi implementado um programa que recebe como entrada uma pirâmide de janelas, um conjunto de

exemplos coletados pela janela na base da pirâmide e o parâmetro α , e executa o algoritmo descrito no Capítulo 4 para estimar a distribuição conjunta de probabilidades. Esta é devolvida na forma de tabelas, como explicado na Seção 4.4.

6.4 Cálculo da Entropia Condicional

A fim de realizarmos experimentos com o critério da entropia condicional, implementamos também um programa que recebe uma representação multirresolução da distribuição conjunta (dada por uma pirâmide de janelas e um conjunto de tabelas de padrões em diversas resoluções) e devolve o valor da sua entropia condicional, calculada como na Seção 5.4.

6.5 Decisão

Este programa recebe como entrada a representação multirresolução da distribuição conjunta e um critério de decisão (moda ou máxima verossimilhança – ver o Capítulo 7) e devolve a representação multirresolução de um operador morfológico projetado de acordo com o critério escolhido. Tal representação é feita na forma de tabelas de padrões nos diversos níveis de resolução e uma pirâmide de janelas, conforme apresentamos na Seção 4.4.

6.6 Aplicação

De posse da representação de um operador multirresolução, este programa aplica o operador a uma imagem, devolvendo a imagem processada. A aplicação é feita da maneira descrita na Seção 4.4. Este programa também aceita como entrada uma imagem máscara. Caso esta seja fornecida, a aplicação do operador é restrita aos pixels pertencentes à máscara.

Capítulo 7

Resultados Experimentais

Aplicamos a nossa técnica ao problema de reconhecimento de dígitos manuscritos, a fim de analisar o seu comportamento em uma aplicação real. Para tal, implementamos um *software* de acordo com as especificações do capítulo anterior, e conduzimos experimentos com duas bases de dados diferentes. Neste capítulo, descreveremos os experimentos e analisaremos os resultados obtidos.

7.1 Reconhecimento de Dígitos Manuscritos

O problema de *reconhecimento de dígitos manuscritos* é um problema clássico na área de análise de imagens. Este consiste em, de posse da imagem de um dígito, determinar qual é o dígito presente na imagem, isto é, determinar automaticamente a classe no conjunto $\{0, 1, \dots, 9\}$ que corresponde ao dígito observado. A grande variabilidade de formas possíveis devido à diversidade de estilos de escrita o torna um problema interessante. Além disso, existem aplicações em diversas áreas, como leitura automática de cheques, identificação de códigos postais em envelopes e leitura automática de formulários.

Nos experimentos realizados, projetamos classificadores de dígitos baseados em W -operadores. Estes são construídos da seguinte forma: para um objeto (dígito) que gostaríamos de classificar, aplicamos um W -operador $\psi : \{0, 1\}^W \rightarrow \{0, 1, \dots, 9\}$ à região do dígito, isto é, analisamos todos os W -padrões cujo centro é um dos pixels do dígito. Assim, teremos como saída uma imagem em que cada pixel na região do dígito recebeu um rótulo no conjunto $\{0, 1, \dots, 9\}$. Então, atribuímos o rótulo que mais aparece entre os pixels do dígito a todos os pixels do dígito. Tal rótulo é o resultado do classificador. A Figura 7.1 ilustra as etapas do processo. Desta maneira, o problema reduz-se ao projeto do W -operador ψ . Durante a etapa de aprendizado de ψ , apenas as configurações cujo centro da janela encontra-se na região do dígito, ou seja, possui valor 1, são usadas no treinamento.

Uma pequena variação da abordagem acima consiste em, antes de aplicar o operador a um objeto (dígito), obter a região C correspondente a uma aproximação de seu fecho convexo, dada pelo fechamento

Figura 7.1: Classificação de dígitos: a) dígito a ser classificado; b) resultado da aplicação do W -operador (cada cor representa um valor de saída diferente); c) dígito classificado.

morfológico [16] do dígito por um disco euclidiano de raio 15. Em seguida, realizar a aplicação do operador restrita à região C , e obter o resultado do classificador escolhendo o rótulo que mais aparece entre os pixels de C . Na etapa de treinamento, o mesmo procedimento é realizado: antes de coletarmos os exemplos é feito um fechamento por um disco de raio 15, e os padrões com centro na região C são utilizados. Além dos padrões que têm o pixel central com valor 1, esta abordagem também considera alguns padrões que têm o pixel central com valor zero, e proporcionou uma pequena redução no erro dos classificadores projetados. A Figura 7.2 mostra as etapas do processo, para o dígito da Figura 7.1.

Figura 7.2: Outra abordagem: a) aproximação C do fecho convexo; b) resultado da aplicação do W -operador restrito à região C (cada cor representa um valor de saída diferente); c) dígito classificado.

O projeto de ψ foi realizado de acordo com a abordagem multirresolução proposta no Capítulo 4. Para isto, um conjunto de 14 pirâmides de janelas baseadas em operadores de subamostragem foi especificado, com janelas grandes na base. Cada pirâmide foi rotulada por um inteiro de 1 a 14. As Figuras 7.3 e 7.4 mostram as especificações das pirâmides, onde cada pirâmide é representada como uma janela que determina a forma de W_0 (a janela da base). As janelas subseqüentes W_i , para $i \geq 1$ podem ser obtidas

tomando apenas os pontos cujos valores são maiores ou iguais a $i + 1$. A Figura 7.5 mostra a sequência de janelas correspondente à pirâmide número 12.

1	1	1	1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2	2	2	1
1	2	3	3	3	3	3	3	3	2	1
1	2	3	4	4	4	4	4	3	2	1
1	2	3	4	5	5	5	4	3	2	1
1	2	3	4	5	6	5	4	3	2	1
1	2	3	4	5	5	5	4	3	2	1
1	2	3	4	4	4	4	4	3	2	1
1	2	3	3	3	3	3	3	3	2	1
1	2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1

Pirâmide 1

1	1	1	1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2	2	2	1
1	2	3	3	3	3	3	3	2	1	1
1	2	3	4	4	4	4	3	2	1	1
1	2	3	4	5	4	3	2	1	1	1
1	2	3	4	4	4	3	2	1	1	1
1	2	3	3	3	3	3	2	1	1	1
1	2	2	2	2	2	2	2	2	1	1
1	1	1	1	1	1	1	1	1	1	1

Pirâmide 2

Pirâmide 3

Pirâmide 4

1	1	1	2	2	2	2	1	1	1	1
1	2	2	3	4	4	4	3	2	2	1
1	2	4	5	6	6	6	5	4	2	1
2	3	5	7	8	9	8	7	5	3	2
2	4	6	8	10	11	10	8	6	4	2
2	4	6	9	11	12	11	9	6	4	2
2	4	6	8	10	11	10	8	6	4	2
2	3	5	7	8	9	8	7	5	3	2
1	2	4	5	6	6	6	5	4	2	1
1	2	2	3	4	4	4	3	2	2	1
1	1	1	2	2	2	2	2	1	1	1

Pirâmide 5

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

Pirâmide 6

1	1	1	1	1	1	1	1	1	1	1
1	3	1	2	1	3	1	2	1	3	1
1	1	1	1	1	1	1	1	1	1	1
1	2	1	2	1	2	1	2	1	2	1
1	1	1	1	1	1	1	1	1	1	1
1	3	1	2	1	4	1	2	1	3	1
1	1	1	1	1	1	1	1	1	1	1
1	2	1	2	1	2	1	2	1	2	1
1	1	1	1	1	1	1	1	1	1	1
1	3	1	2	1	3	1	2	1	3	1
1	1	1	1	1	1	1	1	1	1	1

Pirâmide 7

3	1	2	1	3	1	2	1	3		
1	1	1	1	1	1	1	1	1		
2	1	2	1	2	1	2	1	2		
1	1	1	1	1	1	1	1	1		
3	1	2	1	4	1	2	1	3		
1	1	1	1	1	1	1	1	1		
2	1	2	1	2	1	2	1	2		
1	1	1	1	1	1	1	1	1		
3	1	2	1	3	1	2	1	3		

Pirâmide 8

2	1	2	1	2	1	2	1	2	1	2
1	1	1	1	1	1	1	1	1	1	1
2	1	3	1	2	1	3	1	2	1	3
1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	2	1	2	1	2	1	2
1	1	1	1	1	1	1	1	1	1	1
2	1	3	1	2	1	4	1	2	1	3
1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	2	1	2	1	2	1	2
1	1	1	1	1	1	1	1	1	1	1
2	1	3	1	2	1	3	1	2	1	3
1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	2	1	2	1	2	1	2

Pirâmide 9

Figura 7.3: Pirâmides de imagens usadas nos experimentos.

4	1	2	1	3	1	2	1	4	1	2	1	3	1	2	1	4
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	2	1	3	1	2	1	3	1	2	1	3	1	2	1	3
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	2	1	3	1	2	1	5	1	2	1	3	1	2	1	4
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	2	1	3	1	2	1	3	1	2	1	3	1	2	1	3
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	2	1	3	1	2	1	3	1	2	1	3	1	2	1	3
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	2	1	3	1	2	1	4	1	2	1	3	1	2	1	4

Pirâmide 10

8	1	3	1	5	1	3	1	7	1	3	1	5	1	3	1	8
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
3	1	4	1	3	1	4	1	3	1	4	1	3	1	4	1	3
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
5	1	3	1	6	1	3	1	5	1	3	1	6	1	3	1	5
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
3	1	4	1	3	1	4	1	3	1	4	1	3	1	4	1	3
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
7	1	3	1	5	1	3	1	9	1	3	1	5	1	3	1	7
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
3	1	4	1	3	1	4	1	3	1	4	1	3	1	4	1	3
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
5	1	3	1	6	1	3	1	5	1	3	1	6	1	3	1	5
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
3	1	4	1	3	1	4	1	3	1	4	1	3	1	4	1	3
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
5	1	3	1	6	1	3	1	5	1	3	1	6	1	3	1	5
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
3	1	4	1	3	1	4	1	3	1	4	1	3	1	4	1	3
1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
8	1	3	1	5	1	3	1	7	1	3	1	5	1	3	1	8

Pirâmide 11

12	1	4	1	7	1	4	1	10	1	4	1	7	1	4	1	12
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
5	1	6	1	5	1	6	1	5	1	6	1	5	1	6	1	5
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
8	1	4	1	9	1	4	1	8	1	4	1	9	1	4	1	8
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
5	1	6	1	5	1	6	1	5	1	6	1	5	1	6	1	5
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
11	1	4	1	7	1	4	1	13	1	4	1	7	1	4	1	11
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
5	1	6	1	5	1	6	1	5	1	6	1	5	1	6	1	5
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
8	1	4	1	9	1	4	1	8	1	4	1	9	1	4	1	8
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
5	1	6	1	5	1	6	1	5	1	6	1	5	1	6	1	5
2	3	2	3	2	3	2	3	2	3	2	3	2	3	2	3	2
12	1	4	1	7	1	4	1	10	1	4	1	7	1	4	1	12

Pirâmide 12

Pirâmide 14

9	1	4	1	7	1	4	1	9
2	3	2	3	2	3	2	3	2
5	1	6	1	5	1	6	1	5
2	3	2	3	2	3	2	3	2
8	1	4	1	10	1	4	1	8
2	3	2	3	2	3	2	3	2
5	1	6	1	5	1	6	1	5
2	3	2	3	2	3	2	3	2
9	1	4	1	7	1	4	1	9

Pirâmide 13

Figura 7.4: Pirâmides de imagens usadas nos experimentos.

Figura 7.5: Janelas da pirâmide 12.

Para cada pirâmide, realizamos a estimação da distribuição conjunta usando o algoritmo descrito no Capítulo 4. Os conjuntos de treinamento variam de acordo com a base de dados utilizada, e serão descritos nas próximas seções. Com o objetivo de testar o critério da entropia (Capítulo 5), calculamos a entropia condicional para cada distribuição obtida. O cálculo da entropia foi feito utilizando o logaritmo na base 10, para que o resultado seja um número no intervalo $[0, 1]$. Em seguida, a partir de cada distribuição projetamos dois W -operadores utilizando os seguintes critérios:

- Maximizar $P(Y|\mathbf{x})$: ao decidirmos o valor de saída para um padrão $\mathbf{x} \in D_0$, tomamos o valor $y \in \mathcal{A}_Y$ tal que $P(Y = y|\mathbf{x})$ é máximo (critério da moda);
- Máxima verossimilhança [21]: obtemos $p(\mathbf{X}|Y = y)$, para $y \in \mathcal{A}_Y$, através da igualdade

$$P(\mathbf{x}|y) = \frac{P(y|\mathbf{x}) \cdot P(\mathbf{x})}{P(y)}, \quad (7.1)$$

e escolhemos o valor de saída y que maximiza $P(\mathbf{x}|y)$. A estimação de $P(y)$ é realizada simplesmente tomando a razão entre o número de exemplos com valor de saída y e o número total de exemplos. Na igualdade acima o valor de $P(\mathbf{x})$ atua como uma constante e não interfere na maximização, logo o critério se reduz a maximizar $\frac{P(y|\mathbf{x})}{P(y)}$.

Os classificadores de dígitos baseados nos operadores projetados foram aplicados a imagens em um conjunto de teste (que também será descrito nas próximas seções). Então um valor de erro para cada classificador é obtido calculando-se a razão entre o número de dígitos incorretamente classificados e o número total de dígitos no conjunto de teste. Padrões para os quais $P(\mathbf{x}) = 0$, e portanto não têm uma distribuição condicional associada recebem o valor de saída 11 e não são contabilizados no momento de definir um rótulo para cada objeto (dígito).

O procedimento foi repetido para valores de α (Seção 4.1) no conjunto $\{2, 3, 4, 8, 11\}$. Além disso, projetamos operadores multirresolução usando a metodologia descrita em [1] usando α (Seção 3.2) em $\{1, 2, 3, 4, 8, 11\}$ e comparamos os resultados. Caso haja empate durante o procedimento de decisão, a mediana entre os valores de y que maximizam $P(y|\mathbf{x})$ é escolhida.

7.2 Experimentos com a Nossa Base de Dados

No período de 1994 a 1997, o grupo de processamento de imagens do IME-USP participou do projeto VERASS (verificação de assinaturas) [34], apoiado pela Olivetti do Brasil. Durante a realização do projeto, foram coletadas algumas amostras de dígitos manuscritos. Para cada um dos dez dígitos, 10 imagens binárias contendo aproximadamente 672 dígitos cada foram obtidas através da digitalização (200 dpi) de formulários de papel com amostras escritas por vários indivíduos. Alguns exemplos de dígitos podem ser vistos na Figura 7.6, e o banco de dados está disponível na World Wide Web [35]. Durante o processo de coleta, o objetivo era simular a escrita de dígitos em envelopes de correio. Os indivíduos escreveram diversos dígitos no interior de regiões quadradas.

Como cada imagem contém diversas realizações da escrita de um mesmo dígito, precisamos segmentá-los antes de classificá-los, isto é, é necessário determinar as regiões que correspondem a cada objeto (dígito) na imagem. Isto é feito da seguinte forma: primeiro, dilata-se [16] a imagem por um disco euclidiano de raio 5; em seguida, são eliminados da imagem os componentes conexos cuja largura ou altura sejam

Figura 7.6: Exemplos de dígitos manuscritos (resolução reduzida).

menores ou iguais a 17 ou maiores ou iguais a 73 pixels. Finalmente, para cada componente restante, sua intersecção com a imagem original (antes do pré-processamento) é definida como sendo um único objeto (dígito). Portanto, o resultado é uma imagem que contém os dígitos da imagem original, com exceção de estruturas muito grandes (note que isto pode eliminar alguns dígitos muito próximos uns dos outros) ou muito pequenas, e onde a região correspondente a cada dígito foi determinada. A dilatação morfológica produz o efeito de agrupar em um mesmo objeto pixels separados por pequenas descontinuidades. A Figura 7.7 ilustra este fato.

De posse dos dígitos segmentados, criamos novas imagens aumentando a distância entre os dígitos. Isto é feito para que, ao observar uma certa configuração cujo centro é parte de um dígito, pixels de dígitos vizinhos não sejam incluídos como parte do W -padrão. Então, partimos para a etapa de projeto dos classificadores, como descrito na seção anterior. Neste experimento inicial, utilizamos a abordagem que leva em conta apenas os padrões cujo pixel central tem valor 1. Para definir o conjunto de exemplos de treinamento, 70% das imagens de cada dígito foram selecionadas. As 30% restantes foram reservadas para avaliar o desempenho dos classificadores projetados, o que resultou em um conjunto de teste de 18.874 dígitos.

A Tabela 7.1 mostra os rótulos das pirâmides ordenados pelos valores da entropia condicional de suas respectivas distribuições conjuntas, para diferentes valores de α . Na Tabela 7.2 mostramos as taxas de erro obtidas pelos classificadores projetados usando o critério de máxima verossimilhança. Em seguida, na Tabela 7.3, as taxas de erro obtidas na aplicação, sobre o conjunto de teste, dos classificadores projetados maximizando-se $P(Y | \mathbf{x})$ são listadas. Finalmente, para efeito de comparação incluímos na Tabela 7.4 as taxas de erro obtidas por classificadores projetados de acordo com a metodologia descrita em [1].

Figura 7.7: Segmentação dos dígitos: a) dígitos originais; b) dilatação por um disco de raio 5; c) componentes conexos da imagem dilatada; d) dígitos segmentados.

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$
14	0,276	14	0,348	14	0,385	14	0,452	14	0,477
11	0,317	11	0,387	11	0,422	11	0,487	11	0,511
10	0,441	10	0,496	10	0,523	10	0,572	10	0,591
13	0,477	13	0,554	13	0,596	13	0,668	13	0,692
5	0,480	5	0,593	12	0,636	12	0,697	12	0,716
12	0,527	12	0,599	5	0,651	6	0,723	6	0,744
6	0,543	6	0,623	6	0,661	8	0,739	8	0,754
1	0,572	9	0,651	9	0,684	9	0,742	9	0,763
9	0,580	8	0,663	8	0,692	5	0,748	7	0,772
8	0,600	1	0,676	1	0,726	7	0,764	5	0,779
2	0,611	2	0,696	2	0,738	3	0,799	3	0,819
3	0,652	3	0,710	7	0,740	1	0,804	1	0,827
7	0,680	7	0,724	3	0,742	2	0,808	2	0,830
4	0,807	4	0,826	4	0,837	4	0,859	4	0,868

Tabela 7.1: Entropias condicionais - Nosso BD.

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
14	10,559%	14	11,079%	14	11,651%	14	13,076%	14	13,807%
11	11,407%	11	12,266%	11	12,848%	11	14,401%	11	15,370%
10	14,443%	10	15,042%	10	15,746%	10	17,198%	10	18,067%
13	25,331%	13	24,579%	13	24,722%	13	26,163%	13	26,989%
12	26,200%	12	26,174%	12	26,640%	12	28,346%	12	29,114%
8	28,484%	8	29,029%	8	29,787%	8	31,959%	8	33,008%
7	29,618%	7	30,174%	7	30,836%	7	32,426%	7	33,427%
9	31,228%	6	33,051%	6	34,301%	6	37,946%	6	39,509%
6	31,562%	9	33,077%	9	34,481%	9	38,042%	9	39,626%
3	45,142%	5	44,092%	5	43,880%	5	46,403%	5	48,225%
5	45,237%	3	44,855%	3	45,512%	3	47,971%	3	49,375%
2	48,289%	1	48,527%	1	49,422%	4	51,314%	4	52,389%
1	48,310%	2	48,644%	2	49,634%	1	52,776%	2	54,329%
4	49,470%	4	49,555%	4	49,963%	2	52,829%	1	54,387%

Tabela 7.2: Erros de classificação - Nosso BD - Máxima Verossimilhança.

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
14	11,884%	14	11,937%	14	12,334%	14	13,505%	14	13,855%
11	12,340%	11	12,827%	11	13,362%	11	14,660%	11	15,068%
10	14,295%	10	14,745%	10	15,037%	10	16,064%	10	16,494%
7	29,453%	7	29,808%	13	29,898%	13	29,665%	13	29,522%
8	29,946%	8	29,999%	12	29,999%	12	30,280%	12	30,391%
12	30,878%	12	30,190%	8	30,306%	8	31,710%	8	32,410%
13	31,085%	13	30,296%	7	30,322%	7	31,753%	7	32,452%
9	31,477%	9	33,326%	9	34,884%	6	37,358%	6	38,730%
6	34,094%	6	34,370%	6	35,069%	9	38,190%	9	39,721%
1	50,005%	1	49,979%	1	49,852%	3	50,472%	3	50,461%
2	50,016%	2	50,106%	2	50,117%	1	50,493%	1	51,219%
3	50,996%	3	51,076%	3	50,572%	2	50,726%	2	51,277%
4	51,473%	4	51,621%	4	51,494%	4	51,319%	4	51,452%
5	52,395%	5	52,612%	5	52,326%	5	52,236%	5	51,701%

Tabela 7.3: Erros de classificação - Nosso BD - Moda.

$\alpha = 1$		$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
14	10,141%	14	11,821%	14	11,963%	14	12,329%	14	13,500%	14	13,855%
11	10,787%	11	12,324%	11	12,822%	11	13,362%	11	14,660%	11	15,068%
10	13,087%	10	14,295%	10	14,745%	10	15,037%	10	16,064%	10	16,494%
9	27,450%	7	29,453%	7	29,808%	13	29,893%	13	29,644%	13	29,517%
7	28,213%	8	29,951%	8	29,999%	12	29,999%	12	30,280%	12	30,391%
8	28,505%	12	30,990%	12	30,190%	8	30,306%	8	31,710%	8	32,410%
13	28,870%	13	31,339%	13	30,248%	7	30,322%	7	31,753%	7	32,452%
12	29,300%	9	31,477%	9	33,326%	9	34,884%	6	37,358%	6	38,730%
6	31,117%	6	34,052%	6	34,370%	6	35,069%	9	38,190%	9	39,721%
5	46,551%	2	50,048%	1	49,952%	1	49,862%	3	50,472%	3	50,461%
1	46,556%	1	50,090%	2	50,095%	2	50,133%	1	50,493%	1	51,219%
2	47,181%	3	51,065%	3	51,070%	3	50,583%	2	50,726%	2	51,277%
3	47,944%	4	51,446%	4	51,616%	4	51,494%	4	51,319%	4	51,452%
4	50,705%	5	52,813%	5	52,792%	5	52,294%	5	52,262%	5	51,690%

Tabela 7.4: Erros de classificação - Nosso BD - Abordagem Anterior.

A partir deste experimento inicial, realizamos novos experimentos aplicando algumas normalizações às imagens de forma a simplificar o conjunto de formas a ser analisado. Primeiro, normalizamos o tamanho dos dígitos tendo como referência o tamanho do maior dígito nas imagens, e preservando a relação de aspecto. Em seguida, foi feita a redução de resolução das imagens: para cada dígito na imagem, se a razão entre o número de pixels em sua abertura [16] por um disco de raio 3 e o número de pixels do dígito original for menor do que 0,8, então dilatamos o dígito original por um disco de raio 3 e realizamos subamostragem por um fator de 4. Caso contrário, apenas realizamos a subamostragem. A idéia desta heurística é aumentar a espessura dos dígitos mais finos de forma que na resolução menor os dígitos tenham espessuras similares (Figura 7.8). Além disso, passamos a utilizar a abordagem que considera todos os padrões na aproximação do fecho convexo de cada dígito. Após estas mudanças, obtivemos os resultados apresentados nas Tabelas 7.5, 7.6, 7.7 e 7.8.

Figura 7.8: Normalização de espessura: a) dígitos normalizados pelo tamanho; b) abertura por um disco de raio 3 (em cinza) sobreposta aos dígitos originais; c) dilatação dos 3 dígitos superiores por um disco de raio 3; d) subamostragem da imagem (a) por um fator de 4; e) subamostragem da imagem (c) por um fator de 4. As imagens (a), (b) e (c) estão representadas em escala reduzida.

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$
14	0,187	14	0,248	14	0,282	14	0,352	14	0,381
11	0,217	11	0,274	11	0,306	11	0,372	11	0,399
13	0,263	13	0,325	13	0,358	13	0,420	13	0,443
12	0,300	12	0,362	12	0,395	12	0,457	12	0,481
5	0,350	8	0,438	8	0,466	8	0,522	8	0,545
8	0,386	5	0,445	7	0,484	7	0,530	7	0,551
6	0,393	7	0,463	5	0,497	10	0,541	10	0,552
7	0,427	6	0,475	10	0,509	5	0,590	5	0,624
3	0,431	10	0,490	6	0,519	6	0,602	6	0,633
10	0,444	3	0,511	3	0,554	3	0,632	3	0,661
1	0,467	1	0,559	1	0,605	4	0,667	4	0,688
2	0,475	2	0,563	2	0,608	1	0,683	1	0,709
9	0,493	9	0,577	4	0,614	2	0,684	2	0,710
4	0,543	4	0,588	9	0,620	9	0,693	9	0,718

Tabela 7.5: Entropias condicionais - Nosso BD (normalizado).

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
13	7,529%	13	7,947%	13	8,207%	13	8,885%	13	9,219%
12	7,990%	12	8,440%	12	8,758%	12	9,627%	12	10,120%
8	9,007%	8	9,606%	8	10,067%	8	11,407%	8	12,218%
7	9,314%	7	9,685%	7	10,189%	7	11,508%	7	12,250%
14	9,468%	14	10,077%	14	10,491%	14	11,932%	14	12,456%
11	9,956%	11	10,480%	11	11,021%	11	12,345%	11	12,875%
6	15,402%	6	16,615%	6	17,532%	10	18,396%	10	18,676%
3	15,890%	3	16,833%	10	17,569%	5	19,376%	5	20,239%
5	16,276%	5	16,986%	3	17,585%	3	19,641%	3	20,653%
9	16,329%	10	17,246%	5	17,596%	4	19,985%	4	20,907%
10	16,605%	4	17,495%	4	18,178%	6	20,806%	6	22,322%
4	16,621%	9	18,300%	9	19,741%	9	23,153%	9	24,796%
2	19,212%	2	20,547%	2	21,877%	2	24,695%	2	26,004%
1	19,371%	1	20,690%	1	22,030%	1	24,849%	1	26,020%

Tabela 7.6: Erros de classificação - Nosso BD (normalizado) - Máxima Verossimilhança.

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
13	8,726%	13	8,721%	13	8,912%	13	9,532%	13	9,956%
12	9,028%	12	9,235%	12	9,452%	12	10,353%	12	10,687%
8	9,569%	14	9,982%	8	10,342%	8	11,476%	8	12,191%
14	9,632%	8	9,987%	14	10,427%	7	11,688%	14	12,250%
7	9,722%	7	10,146%	7	10,501%	14	11,757%	7	12,281%
11	9,998%	11	10,507%	11	10,787%	11	12,117%	11	12,578%
9	15,158%	10	16,356%	10	16,652%	10	17,410%	10	17,633%
10	15,773%	9	16,478%	9	17,654%	9	20,462%	9	21,734%
6	16,107%	6	17,458%	6	18,459%	6	22,263%	6	24,192%
3	20,950%	5	21,522%	5	22,306%	5	24,176%	5	25,125%
5	21,045%	3	21,681%	3	22,597%	3	24,992%	3	26,253%
4	22,523%	4	23,360%	4	24,123%	4	26,343%	4	27,535%
1	23,561%	2	24,616%	2	25,728%	2	28,722%	2	30,168%
2	23,604%	1	24,722%	1	25,808%	1	28,786%	1	30,184%

Tabela 7.7: Erros de classificação - Nosso BD (normalizado) - Moda.

$\alpha = 1$		$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
13	7,577%	13	8,774%	13	8,694%	13	8,906%	13	9,532%	13	9,956%
12	8,022%	12	9,039%	12	9,230%	12	9,458%	12	10,353%	12	10,687%
14	8,143%	14	9,516%	14	9,987%	8	10,342%	8	11,476%	8	12,191%
8	8,700%	8	9,569%	8	9,987%	14	10,432%	7	11,688%	14	12,250%
11	8,838%	7	9,722%	7	10,146%	7	10,501%	14	11,757%	7	12,281%
7	8,853%	11	9,987%	11	10,507%	11	10,787%	11	12,117%	11	12,578%
9	12,048%	9	15,158%	10	16,356%	10	16,652%	10	17,410%	10	17,633%
6	13,633%	10	15,773%	9	16,478%	9	17,654%	9	20,462%	9	21,734%
10	14,295%	6	16,070%	6	17,458%	6	18,459%	6	22,263%	6	24,192%
5	17,506%	3	20,838%	5	21,442%	5	22,300%	5	24,197%	5	25,109%
3	17,649%	5	20,950%	3	21,665%	3	22,576%	3	24,992%	3	26,253%
2	20,319%	4	22,481%	4	23,350%	4	24,128%	4	26,338%	4	27,535%
1	20,377%	1	23,493%	2	24,600%	2	25,723%	2	28,717%	2	30,168%
4	20,388%	2	23,546%	1	24,695%	1	25,797%	1	28,780%	1	30,184%

Tabela 7.8: Erros de classificação - Nosso BD (normalizado) - Abordagem Anterior.

Podemos notar uma redução considerável no erro dos classificadores projetados. Isto condiz com o fato de que, ao fazermos a normalização, reduzimos o conjunto de formas a ser analisado, e, portanto, simplificamos o processo estocástico observado, resultando em uma melhor estimativa da distribuição de probabilidade [36].

7.3 Experimentos com a Base de Dados MNIST

A base de dados MNIST [37] consiste em um banco de imagens de dígitos manuscritos separados em dois conjuntos: um de imagens de teste, contendo 10.000 dígitos, e um de treinamento, contendo 60.000 dígitos. Os dígitos são normalizados quanto ao tamanho e centralizados em imagens de tamanho fixo. Eles também são fornecidos já segmentados, isto é, cada imagem contém apenas um dígito. Este é um teste bastante difundido para algoritmos de análise de formas e para comparação entre algoritmos de aprendizado.

Como as imagens disponíveis contêm níveis de cinza (devido ao algoritmo de normalização usado pelos autores), estas foram transformadas em imagens binárias através de uma simples limiarização. Aplicamos o método sobre este banco de dados, e obtivemos o seguinte resultado. Neste caso, não foi feita nenhuma normalização, e também consideramos todos os padrões na região do fecho convexo de cada dígito:

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$
14	0,182	14	0,231	14	0,257	14	0,308	14	0,328
11	0,204	11	0,253	11	0,279	11	0,331	11	0,353
10	0,327	10	0,364	10	0,382	10	0,420	10	0,437
13	0,367	13	0,429	13	0,462	13	0,520	13	0,541
12	0,405	12	0,463	12	0,493	12	0,545	12	0,564
5	0,405	5	0,498	6	0,541	8	0,593	8	0,609
6	0,431	6	0,504	5	0,547	6	0,607	7	0,625
8	0,476	8	0,526	8	0,550	7	0,614	6	0,632
1	0,493	7	0,569	7	0,585	5	0,631	5	0,660
9	0,499	9	0,571	9	0,606	9	0,665	9	0,686
2	0,522	1	0,581	3	0,618	3	0,675	3	0,696
3	0,531	3	0,588	1	0,623	1	0,695	1	0,720
7	0,535	2	0,597	2	0,634	2	0,700	2	0,723
4	0,693	4	0,712	4	0,723	4	0,747	4	0,757

Tabela 7.9: Entropias condicionais - MNIST.

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
14	3,29%	14	3,56%	14	3,74%	14	4,21%	14	4,56%
11	3,49%	11	3,73%	11	3,99%	11	4,47%	11	4,83%
10	4,79%	10	4,97%	10	5,11%	10	5,58%	10	5,67%
13	7,40%	13	7,61%	13	7,76%	13	8,55%	13	8,91%
12	7,82%	12	8,08%	12	8,31%	12	9,15%	12	9,41%
8	8,21%	8	8,53%	8	8,86%	7	9,63%	8	10,12%
7	8,53%	7	8,74%	7	9,03%	8	9,72%	7	10,15%
6	9,25%	6	10,19%	6	11,06%	6	13,02%	6	14,17%
9	9,33%	9	10,48%	9	11,20%	9	13,86%	9	15,09%
5	15,16%	5	16,11%	5	16,95%	5	19,89%	5	21,20%
3	15,81%	3	17,05%	3	17,91%	3	20,44%	3	22,03%
1	17,95%	1	19,78%	1	20,99%	1	25,41%	1	27,53%
2	18,24%	2	20,04%	2	21,22%	2	25,44%	4	27,69%
4	24,06%	4	24,73%	4	25,26%	4	26,83%	2	27,73%

Tabela 7.10: Erros de classificação - MNIST - Máxima Verossimilhança.

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
14	3,84%	14	3,95%	14	4,07%	14	4,51%	14	4,71%
11	3,87%	11	4,13%	11	4,39%	11	4,76%	11	5,08%
10	4,99%	10	5,15%	10	5,24%	10	5,61%	10	5,82%
13	10,33%	13	9,97%	13	9,96%	13	10,66%	13	10,86%
12	10,55%	12	10,38%	12	10,49%	12	11,15%	12	11,48%
8	10,83%	8	11,05%	7	11,34%	7	12,05%	8	12,43%
7	10,90%	7	11,19%	8	11,36%	8	12,08%	7	12,44%
9	11,14%	9	12,44%	9	13,34%	9	16,39%	9	17,44%
6	13,37%	6	14,01%	6	14,94%	6	17,30%	6	18,67%
3	22,39%	3	22,76%	3	23,40%	3	25,27%	3	26,31%
5	22,87%	5	22,95%	5	23,61%	5	25,89%	5	26,57%
2	25,34%	1	26,55%	2	27,51%	2	30,67%	4	31,74%
1	25,37%	2	26,56%	1	27,61%	1	30,79%	2	32,14%
4	29,09%	4	29,30%	4	29,65%	4	31,00%	1	32,26%

Tabela 7.11: Erros de classificação - MNIST - Moda.

$\alpha = 1$		$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
14	3,17%	14	3,78%	14	3,94%	14	4,07%	14	4,51%	14	4,71%
11	3,40%	11	3,83%	11	4,13%	11	4,39%	11	4,76%	11	5,08%
10	4,51%	10	4,99%	10	5,15%	10	5,24%	10	5,61%	10	5,82%
9	8,91%	13	10,37%	13	9,97%	13	9,97%	13	10,64%	13	10,86%
13	9,37%	12	10,58%	12	10,39%	12	10,50%	12	11,15%	12	11,48%
12	9,67%	8	10,83%	8	11,05%	7	11,34%	7	12,05%	8	12,43%
8	10,08%	7	10,90%	7	11,19%	8	11,36%	8	12,08%	7	12,44%
7	10,35%	9	11,14%	9	12,44%	9	13,34%	9	16,39%	9	17,44%
6	10,93%	6	13,36%	6	14,00%	6	14,94%	6	17,30%	6	18,67%
5	18,00%	3	22,42%	5	22,78%	3	23,42%	3	25,27%	3	26,31%
3	19,65%	5	22,76%	3	22,80%	5	23,62%	5	25,93%	5	26,58%
1	20,83%	2	25,28%	2	26,55%	2	27,48%	2	30,67%	4	31,74%
2	21,12%	1	25,32%	1	26,58%	1	27,62%	1	30,79%	2	32,14%
4	27,86%	4	29,09%	4	29,29%	4	29,65%	4	31,00%	1	32,26%

Tabela 7.12: Erros de classificação - MNIST - Abordagem Anterior.

Como podemos ver, temos bons resultados. Apresentamos na Tabela 7.13 as taxas de erro obtidas por alguns algoritmos listados em [37] para efeito de comparação. Mais detalhes sobre tais experimentos podem ser vistos em [37, 38]. Na comparação incluímos apenas experimentos que usaram apenas os dados brutos no treinamento, mas em [37] também são citados experimentos que realizaram algum tipo de pré-processamento nos dados, ou então experimentos em que o conjunto de treinamento foi aumentado usando variações das amostras originais, obtidas através de distorções aleatórias, como translação, rotação, mudança de escala e compressão. Entre os resultados com tais modificações existem tanto algoritmos com taxas de erro maiores do que as que obtivemos em nossos experimentos, quanto algoritmos com taxas de erro menores. O melhor algoritmo listado apresenta erro igual a 0,4%, e baseia-se em redes neurais convolucionais usando um conjunto de treinamento expandido [39]. Isto sugere um possível tópico para pesquisa futura: investigar o efeito, sobre o projeto multirresolução de W -operadores, de aumentar o conjunto de treinamento com amostras distorcidas.

Algoritmo	Erro
Classificador linear	12,0%
k -vizinhos mais próximos, L2 (LeCun)	5,0%
Rede neural de 2 camadas, 300 unidades ocultas, MSE	4,7%
Rede neural de 2 camadas, 1000 unidades ocultas	4,5%
1000 RBF + classificador linear	3,6%
40 PCA + classificador quadrático	3,3%
W -operadores multirresolução	3,17%
k -vizinhos mais próximos, L2 (Wilder)	3,09%
Rede neural de 3 camadas, 300+100 unidades ocultas	3,05%
Rede neural de 3 camadas, 500+150 unidades ocultas	2,95%
k -vizinhos mais próximos, L3 (Wilder)	2,83%
SVM, kernel Gaussiano	1,4%
Rede neural convolucional LeNet-4	1,1%
Rede neural convolucional LeNet-5	0,95%

Tabela 7.13: Comparação de algoritmos sobre a base de dados MNIST.

Outra informação interessante de ser verificada é o erro de classificação considerando cada dígito separadamente. A Tabela 7.14 mostra as taxas de erro por dígito do melhor classificador obtido para a base de dados do MNIST, isto é, do classificador projetado a partir da pirâmide 14 segundo a metodologia de [1], cujo erro aparece na Tabela 7.12. Também são listados os números (N_d) de dígitos de cada classe presentes no conjunto de teste. As taxas de erro para cada dígito d , $d \in \{0, 1, \dots, 9\}$ foram obtidas dividindo o número de dígitos da classe d no conjunto de teste que foram classificados incorretamente pelo número total N_d de dígitos da classe d no conjunto de teste.

Dígito	0	1	2	3	4	5	6	7	8	9
N_d	980	1135	1032	1010	982	892	958	1028	974	1009
Erro	0,612%	1,322%	2,035%	2,970%	3,462%	5,045%	2,610%	5,447%	2,669%	5,847%

Tabela 7.14: Taxas de erro por dígito, para o classificador projetado a partir da pirâmide 14 no experimento da Tabela 7.12.

7.4 Experimentos com Mapeamentos da Teoria de Pirâmides de Imagens

Um dos objetivos deste trabalho é a experimentação com mapeamentos de resolução utilizados na teoria de pirâmides de imagens. Realizamos experimentos com alguns dos mapeamentos descritos no Apêndice A. Através destes experimentos, descobrimos a utilidade da amostragem *quincunx* [40] no projeto de operadores para o problema de reconhecimento de dígitos manuscritos. As pirâmides 11 e 12 dos experimentos mostrados nas seções anteriores foram construídas usando simples subamostragem de acordo com o esquema *quincunx*, e as pirâmides 13 e 14 usam uma ligeira variação do mesmo.

Já os mapeamentos compostos por uma filtragem antes da etapa de amostragem não levaram a bons resultados. Intuitivamente, ao reduzirmos a resolução dos padrões observados gostaríamos que dígitos de uma mesma classe se tornassem cada vez mais parecidos; ao mesmo tempo, gostaríamos que dígitos de classes diferentes não se misturassem. No problema de reconhecimento de dígitos manuscritos isto não acontecia com os mapeamentos utilizados, pois havia uma degradação muito rápida dos dígitos. Isto fazia com que se as classes se misturassem muito rapidamente.

Porém, temos um exemplo de experimento que mostra que mapeamentos baseados em filtros podem ser melhores do que a simples subamostragem em alguns problemas. Em vez de utilizarmos as imagens brutas da base de dados do MNIST, utilizamos o esqueleto dos dígitos tanto no projeto quanto na aplicação dos classificadores. Todos os padrões no fecho convexo do esqueleto de cada dígito foram considerados. Neste experimento, uma pirâmide baseada em dilatação [16] seguida de amostragem saiu-se melhor do que as melhores pirâmides obtidas até o momento para as imagens brutas. Isto é esperado, visto que a subamostragem degrada rapidamente o esqueleto da imagem. As Tabelas 7.15, 7.16 e 7.17 mostram os resultados obtidos, onde *Dil* representa a pirâmide definida pela dilatação por um quadrado de lado 2 seguida pela subamostragem diádica.

Este experimento mostra claramente que a melhor pirâmide a ser escolhida depende do problema a ser resolvido. Outra linha de pesquisa interessante seria a experimentação com os mapeamentos de resolução da teoria de pirâmides de imagens em outras aplicações, incluindo problemas para imagens em níveis de cinza ou coloridas.

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$	Δ	$H(Y \mathbf{X})$
<i>Dil</i>	0,333	<i>Dil</i>	0,379	<i>Dil</i>	0,400	<i>Dil</i>	0,436	<i>Dil</i>	0,449
14	0,345	14	0,434	14	0,481	13	0,536	13	0,557
13	0,413	13	0,457	13	0,484	12	0,556	12	0,577
12	0,431	12	0,477	12	0,504	14	0,568	14	0,601
11	0,432	11	0,509	11	0,548	11	0,619	11	0,645
10	0,626	10	0,689	10	0,716	10	0,758	10	0,773

Tabela 7.15: Entropias condicionais - MNIST (Esqueletos).

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
<i>Dil</i>	8,17%	<i>Dil</i>	8,46%	<i>Dil</i>	8,67%	<i>Dil</i>	9,26%	<i>Dil</i>	9,34%
14	10,67%	14	11,35%	14	12,06%	13	13,11%	13	13,51%
13	11,35%	13	11,98%	13	12,23%	14	13,66%	12	14,38%
11	11,56%	12	12,20%	12	12,53%	12	13,77%	14	14,65%
12	11,68%	11	12,43%	11	12,99%	11	14,57%	11	15,52%
10	22,31%	10	24,45%	10	26,02%	10	29,84%	10	31,48%

Tabela 7.16: Erros de classificação - MNIST (Esqueletos) - Máxima Verossimilhança.

$\alpha = 2$		$\alpha = 3$		$\alpha = 4$		$\alpha = 8$		$\alpha = 11$	
Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro	Δ	Erro
<i>Dil</i>	8,74%	<i>Dil</i>	9,08%	<i>Dil</i>	9,28%	<i>Dil</i>	9,62%	<i>Dil</i>	9,83%
14	13,37%	14	13,99%	14	14,69%	14	17,05%	14	18,38%
11	14,72%	11	15,35%	11	16,33%	13	17,94%	13	18,49%
13	16,34%	13	16,75%	13	16,95%	12	18,65%	12	19,42%
12	16,78%	12	16,97%	12	17,45%	11	18,67%	11	20,00%
10	27,32%	10	29,36%	10	31,04%	10	34,84%	10	36,73%

Tabela 7.17: Erros de classificação - MNIST (Esqueletos) - Moda.

7.5 Experimentos com Tamanhos Variados de Amostras

Uma propriedade dos bons algoritmos de aprendizado é a diminuição do erro conforme o número de amostras de treinamento aumenta. Quando o número de amostras tende a infinito, o erro do classificador projetado tende a zero. Com o objetivo de analisarmos o comportamento da nossa técnica conforme o número de amostras aumenta, realizamos experimentos com a base de dados MNIST variando o número de dígitos usados para treinamento, enquanto o conjunto de teste foi mantido fixo em 10000 dígitos. Usamos $\alpha = 2$. As tabelas seguintes mostram os resultados obtidos, onde, para cada pirâmide, apresentamos as taxas de erro (em %) para números variáveis de dígitos usados no treinamento. Podemos verificar

também que para as melhores pirâmides o erro aumenta bem pouco quando diminuimos a quantidade de exemplos.

Δ	6000	12000	18000	24000	30000	36000	42000	48000	54000	60000
1	32,09	26,41	24,58	22,29	20,83	20,08	19,33	18,90	18,53	17,95
2	32,15	26,51	24,62	22,48	21,04	20,48	19,67	19,12	18,95	18,24
3	25,94	21,94	20,42	19,11	18,10	17,01	16,81	16,62	16,34	15,81
4	31,43	28,24	27,52	26,42	25,56	25,04	24,69	24,66	24,42	24,06
5	24,90	20,89	20,03	17,99	17,30	16,63	16,10	15,71	15,34	15,16
6	17,62	13,78	12,48	11,22	10,54	9,89	9,74	9,46	9,34	9,25
7	11,46	10,18	9,62	9,02	8,73	8,67	8,63	8,64	8,57	8,53
8	11,10	9,61	9,39	8,83	8,47	8,46	8,25	8,24	8,04	8,21
9	17,91	14,68	12,72	11,92	11,04	10,37	9,88	9,80	9,36	9,33
10	6,68	6,01	5,75	5,51	5,16	4,93	4,85	4,88	4,79	4,79
11	5,75	5,11	4,63	4,18	4,00	3,64	3,50	3,56	3,55	3,49
12	9,59	8,99	8,70	8,33	7,99	7,81	7,70	7,89	7,73	7,82
13	9,18	8,35	8,18	7,97	7,57	7,48	7,27	7,34	7,20	7,40
14	5,53	4,72	4,31	3,99	3,79	3,57	3,41	3,43	3,42	3,29

Tabela 7.18: Erros (em %) para números diferentes de amostras de treinamento - MNIST - Máxima Verossimilhança.

Δ	6000	12000	18000	24000	30000	36000	42000	48000	54000	60000
1	44,12	36,59	32,70	30,78	28,39	27,46	26,82	25,86	25,12	25,37
2	43,96	36,47	32,40	30,83	28,33	27,20	26,66	25,82	25,12	25,34
3	38,20	31,73	27,99	25,88	24,89	23,75	22,99	22,43	22,12	22,39
4	42,65	37,19	33,70	32,47	31,61	30,02	29,14	29,22	28,51	29,09
5	39,36	33,26	29,70	27,38	26,07	24,90	23,69	23,17	22,62	22,87
6	23,51	19,59	17,33	16,26	15,44	14,76	14,06	13,53	13,38	13,37
7	15,06	13,38	11,98	11,59	11,44	11,19	10,98	11,02	10,90	10,90
8	15,18	13,62	12,16	11,77	11,52	11,26	11,32	11,13	10,82	10,83
9	21,06	17,66	15,29	14,79	13,58	13,18	12,31	11,82	11,78	11,14
10	6,96	6,31	5,84	5,59	5,30	5,13	5,07	5,02	4,88	4,99
11	6,07	5,44	4,89	4,59	4,31	4,05	3,86	3,93	3,86	3,87
12	14,38	13,02	12,14	11,62	11,41	11,19	10,86	10,76	10,60	10,55
13	14,27	12,89	11,95	11,55	11,17	11,00	10,75	10,62	10,55	10,33
14	5,69	5,17	4,80	4,49	4,25	3,93	3,76	3,76	3,74	3,84

Tabela 7.19: Erros (em %) para números diferentes de amostras de treinamento - MNIST - Moda.

7.6 Análise

Nos experimentos realizados, pudemos observar os valores da entropia condicional da distribuição induzida por cada pirâmide, assim como os valores de erro (obtidos através da aplicação sobre um conjunto de teste) dos classificadores projetados a partir da distribuição estimada usando os critérios da moda e máxima verossimilhança. Em cada experimento, mostramos também os valores de erro do classificador projetado usando a abordagem de [1], para efeito de comparação.

Dividiremos esta análise em duas partes: a primeira discutirá a utilização da entropia condicional como critério de escolha da pirâmide, enquanto a segunda comentará algumas características interessantes do método, como a adequação ao projeto de operadores de janelas grandes e o desempenho dos diferentes critérios de decisão utilizados na escolha do operador.

7.6.1 Uso da Entropia Condicional na Escolha da Pirâmide

Vamos iniciar recordando a motivação da experimentação com o critério da entropia condicional. Trabalhamos com a hipótese de que as distribuições condicionais reais possuem massa de probabilidade concentrada em um dos possíveis valores de saída. Este tipo de distribuição possui baixa entropia; então, a idéia é procurar, entre as pirâmides de um conjunto escolhido pelo projetista de operadores, a pirâmide que induz a distribuição de menor entropia condicional. O objetivo dos experimentos foi verificar se o classificador projetado a partir da pirâmide de menor entropia coincide com o “melhor” classificador dentre os projetados a partir das pirâmides do conjunto, isto é, com o classificador de menor erro ao ser aplicado em dados reais. Este erro foi estimado a partir da aplicação do classificador a um conjunto de teste (uma amostra dos dados reais).

No primeiro experimento com a nossa base de dados (Seção 7.2), no experimento com a base de dados MNIST (Seção 7.3) e no experimento com os esqueletos (Seção 7.4) o critério da entropia condicional foi eficiente na escolha da pirâmide que induz o classificador de menor erro entre as pirâmides consideradas. Já no experimento com o nosso banco de dados normalizado (final da Seção 7.2), a melhor pirâmide quanto ao erro não coincide com a pirâmide que induz a distribuição de menor entropia. Tais resultados sugerem que a entropia condicional pode ser útil na formulação de um critério a ser usado na escolha automática da pirâmide, mas usá-la isoladamente não resolve o problema em todos os casos possíveis. Discutiremos agora algumas limitações do critério da entropia.

Uma observação importante é o fato de que a entropia reflete a concentração de massa nas distribuições condicionais, mas não informa nada sobre os valores de saída nos quais a massa se concentra. Por exemplo, suponha que para um determinado padrão $\mathbf{x}$ não observado nos exemplos a pirâmide Δ_A induziu a distribuição $P(Y = 0 | \mathbf{x}) = 0,2$ e $P(Y = 1 | \mathbf{x}) = 0,8$, enquanto a pirâmide Δ_B induziu $P(Y = 0 | \mathbf{x}) = 0,8$ e $P(Y = 1 | \mathbf{x}) = 0,2$. A entropia será a mesma nos dois casos, mas, considerando a classificação do padrão $\mathbf{x}$ e os operadores projetados usando o critério da moda a partir de cada distribuição, claramente um dos

operadores terá erro maior do que o outro se supormos que a distribuição real possui massa concentrada em um dos dois valores de saída, assumindo que o conjunto de teste tenha distribuição similar à real.

Como trabalhamos sob a hipótese de que nas distribuições condicionais reais a massa de probabilidade se concentra em uma das possíveis classes, se uma pirâmide que induz uma distribuição com estas características estiver no conjunto de pirâmides candidatas o critério da entropia será eficiente para encontrá-la. No entanto, como utiliza-se um conjunto de pirâmides limitado, pode não existir uma pirâmide com estas características entre as candidatas. Neste caso, a entropia pode não ser eficiente para diferenciar as pirâmides.

É interessante observar que no experimento com o nosso banco de dados normalizado (Tabelas 7.5, 7.6, 7.7 e 7.8), entre as pirâmides construídas com base na amostragem *quincunx* o critério da entropia favorece as pirâmides 14 e 11, de base 17×17 , em relação às pirâmides 13 e 12, de base 9×9 , mas o erro das pirâmides 13 e 12 é menor do que o das pirâmides 14 e 11. Isto sugere que um critério de escolha da pirâmide deveria levar em conta o tamanho da janela base, já que os erros de estimação da distribuição nos dois casos são diferentes (pois o número de amostras é o mesmo). Outro fatores que influenciam no erro de estimação e poderiam ser considerados são o número de janelas em cada pirâmide e o número de pontos em cada uma delas. O critério da entropia usado isoladamente não diferencia entre tais casos.

Uma análise mais apurada também deveria considerar a semelhança entre a distribuição dos dados usados no treinamento e no teste com a distribuição dos dados reais a serem classificados, erros na estimação da distribuição conjunta e a validade da hipótese de concentração de massa no problema em questão. Além disso, não foi feito o cálculo de medidas médias a partir da repetição dos experimentos para diversos conjuntos de treinamento e teste sorteados ao acaso, o que seria mais preciso do ponto de vista estatístico.

Em resumo, o critério da entropia condicional foi eficiente em encontrar a melhor pirâmide do conjunto de candidatas na maioria dos experimentos realizados. Porém, o critério ainda poderia ser aperfeiçoado para contornar algumas limitações, e uma análise mais detalhada do ponto de vista estatístico seria necessária para chegar a uma conclusão mais significativa.

7.6.2 Outras Considerações

A pirâmide rotulada com o número 14, que saiu-se melhor na maioria dos experimentos, tem uma janela 17×17 em sua base, e é construída utilizando uma variação do esquema de subamostragem conhecido como *quincunx* para determinar a sequência de janelas. O número de W -padrões observados foi igual a 10.473.292 no primeiro experimento com o nosso banco de dados, 5.629.143 com o nosso banco de dados normalizado e 12.335.931 no experimento com a base de dados do MNIST. Ou seja, a quantidade de exemplos observados é da ordem de 2^{23} , o que é muito menor do que 2^{289} , a quantidade de configurações possíveis de serem observadas por uma janela 17×17 . Isto é uma evidência de que a abordagem multirresolução pode proporcionar bons resultados com janelas grandes, quando o número de amostras de

treinamento é muito pequeno comparado com o número de diferentes padrões possíveis de serem observados pela janela.

Quanto ao critério de decisão utilizado, verificamos resultados bastante similares nos dois casos. Apesar do critério da moda ser ótimo ao se trabalhar com distribuições reais, no caso de distribuições estimadas isto pode não ser verdade devido aos erros de estimação da distribuição. Nos nossos experimentos, a máxima verossimilhança resultou em operadores ligeiramente melhores do que os projetados usando o critério da moda. Na comparação do nosso estimador das distribuições condicionais com o estimador de [1], ambos também proporcionaram resultados similares. Como discutimos na Seção 4.3, o nível da pirâmide em que a distribuição condicional é estimada pode variar de uma metodologia para a outra. Isto influencia no resultado final, e também pode ser assunto de pesquisas futuras sobre a melhor resolução a ser usada na estimação da distribuição condicional para um dado padrão, ou uma técnica multiescala que combine informação das diversas resoluções. Sobre o parâmetro α , usar o menor α possível levou aos melhores resultados com os conjuntos de dados que utilizamos. Porém, isto pode ser diferente em outros problemas.

A aplicação do projeto multirresolução de operadores morfológicos ao problema de reconhecimento de dígitos manuscritos também é uma contribuição importante deste trabalho. O método apresentou taxas de acerto muito boas, considerando que a abordagem é geral e não foi realizado nenhum tipo de ajuste específico para a aplicação particular de reconhecimento de dígitos.

Capítulo 8

Conclusão

Neste trabalho, concentramos nossos esforços no sentido de criar uma técnica de escolha automática da pirâmide de janelas a ser utilizada no projeto de um W -operador multirresolução. Investigamos o uso da entropia condicional como um critério de escolha da pirâmide; para isto foi necessário desenvolver um algoritmo de estimação da distribuição conjunta que, além de estimar as distribuições condicionais $p(Y | \mathbf{x})$, estima também a distribuição $p(\mathbf{X})$. Usamos o problema de reconhecimento de dígitos manuscritos para testar a nossa metodologia.

Abordamos o problema de projeto estatístico de operadores como um problema composto por duas etapas sequenciais: a estimação das distribuições de probabilidade condicional e a escolha do operador ótimo, dentre uma família de operadores candidatos, segundo uma medida de erro. O problema de escolha da pirâmide foi formulado como um problema na primeira etapa, considerando o uso de uma pirâmide como um modelo que induz as distribuições condicionais a partir de dados de treinamento. A questão então passa a ser encontrar uma “boa” distribuição induzida por uma pirâmide, ou seja, uma distribuição que ao ser usada no projeto do operador resulte em um operador de baixo erro quando aplicado sobre dados reais. Trabalhamos com a hipótese de que as distribuições condicionais reais apresentam massa concentrada em um dos valores de saída. Desta forma, uma “boa” distribuição passa a ser uma que possua massa de probabilidade concentrada, e a entropia surge como um critério natural para procurar por uma distribuição com esta característica. A utilização de uma estrutura piramidal para representar a distribuição permite que esta seja armazenada de forma eficiente.

Conduzimos diversos experimentos com o critério da entropia, para analisar a sua capacidade de, a partir de um conjunto de pirâmides fornecido pelo projetista de operadores e um conjunto de amostras de treinamento, escolher automaticamente uma boa pirâmide. Na maioria dos experimentos realizados, o critério da entropia foi eficiente na identificação da melhor pirâmide no conjunto de pirâmides candidatas. Porém, em um dos experimentos isto não ocorreu.

Como discutimos na Seção 7.6, a medida de entropia apresenta algumas limitações que devem ser

levadas em conta ao utilizá-la como um critério para escolher a distribuição utilizada. Entre elas, podemos citar o fato dela não levar em consideração os valores de saída em que as distribuições condicionais se concentram, nem as diferenças nos erros de estimação presentes quando compara-se pirâmides com janelas de tamanhos diferentes em sua base, assim como pirâmides com números distintos de níveis e diferentes números de pontos em cada janela. Uma possível continuidade deste trabalho poderia ser o aprimoramento do critério de escolha para, além de tirar proveito das propriedades da entropia condicional, utilizar outras medidas com propriedades complementares. Por exemplo, outras medidas sobre a distribuição estimada, como o erro MAE do operador projetado pelo critério da moda calculado com base na distribuição estimada, poderiam ser investigadas.

Outros tópicos importantes do ponto de vista estatístico que não foram incluídos em nossa análise são a verificação da semelhança entre a distribuição dos dados usados no treinamento e no teste com a distribuição dos dados reais a serem classificados, erros na estimação da distribuição conjunta e a validade da hipótese de concentração de massa no problema em questão. Além disso, fica também para trabalhos futuros a obtenção de medidas médias a partir da repetição dos experimentos para diversos conjuntos de treinamento e teste sorteados ao acaso.

Outra contribuição deste trabalho foi a experimentação com mapeamentos de resolução da teoria de pirâmides de imagens no contexto do projeto multirresolução de W -operadores. Através destes experimentos, identificamos uma boa pirâmide para o projeto de classificadores de dígitos manuscritos. Tal pirâmide consiste em simples subamostragem de acordo com o esquema *quincunx*. Mesmo que os mapeamentos baseados em filtros não tenham levado a bons resultados nesta aplicação em particular, apresentamos um exemplo de problema em que um mapeamento baseado em dilatação morfológica poderia ser útil. Isto comprova que a escolha dos mapeamentos de resolução depende diretamente do problema considerado.

Em caráter prático, uma contribuição original deste trabalho foi a aplicação do projeto multirresolução de operadores morfológicos ao problema de classificação de dígitos manuscritos. O método apresentou bons resultados, com potencial para melhorias.

Em suma, neste trabalho estudamos diversos aspectos do projeto multirresolução de W -operadores, desde os seus fundamentos teóricos até a sua aplicação a um problema real, e propusemos respostas a algumas questões neste campo de pesquisa. Como é inerente ao processo científico, muitas outras questões surgiram durante o desenvolvimento do nosso trabalho; estas podem ser assunto de trabalhos futuros. A seguir, listamos algumas delas:

- Restrições sobre a distribuição de probabilidade combinadas com restrições algébricas sobre a família de operadores candidatos: neste trabalho, utilizamos as pirâmides para restringir o conjunto de distribuições de probabilidade consideradas durante o processo de estimação. Porém, de posse da distribuição estimada, não foi feita restrição sobre o espaço de busca do operador ao escolher o operador ótimo segundo uma medida de erro. Uma linha de pesquisa interessante seria a utilização de diferentes restrições sobre as distribuições consideradas na estimação e sobre a família de ope-

radores candidatos. Por exemplo, poderíamos usar a abordagem multirresolução para estimar as distribuições condicionais e restringir o espaço de busca do operador aos operadores crescentes;

- Critério de escolha da pirâmide: como foi apresentado anteriormente, o critério de escolha pode ser aperfeiçoado com medidas que levem em conta outros fatores como erros de estimação, tamanho das janelas, e número de níveis da pirâmide;
- Escolha da resolução em que a distribuição condicional é estimada para um dado padrão: no trabalho de Dougherty *et al.* [1], a distribuição condicional é estimada na maior resolução em que existem pelo menos α amostras nos exemplos. No estimador proposto no Capítulo 4, a resolução varia de acordo com os dados de treinamento. Isto levou a pequenas diferenças nos resultados experimentais. Um trabalho futuro poderia propor uma técnica que determinasse a melhor resolução a ser usada na estimação da distribuição condicional para um dado padrão, ou uma abordagem multiescala que combine informações das diversas resoluções na estimação da distribuição condicional;
- Parâmetro α : poderiam ser investigados métodos de determinação automática do parâmetro α usado no algoritmo de estimação. Outra possível modificação seria o uso de múltiplos valores de α , um em cada resolução;
- Mapeamentos de resolução: neste trabalho fizemos experimentos com mapeamentos da teoria de pirâmides de imagens. Porém, na literatura existem outros tipos de decomposições multirresolução, como as estudadas pela teoria de *wavelets* [41, 42], que também poderiam ser úteis no projeto de operadores. Outra possível direção seria o uso de mapeamentos de resolução baseados em diferentes filtros em cada nível da pirâmide – neste trabalho consideramos apenas pirâmides em que o mesmo filtro é usado em cada resolução;
- Relação com o problema de redução de dimensionalidade: o problema de seleção de características busca encontrar, para um determinado conjunto de características que descrevem um padrão, um subconjunto que melhor o represente. Podemos fazer um paralelo com as pirâmides baseadas em simples subamostragem, onde em cada redução de resolução escolhemos um subconjunto dos pixels da janela anterior. Um trabalho futuro poderia utilizar métodos de seleção de características para determinar progressivamente as janelas da pirâmide de acordo com os pixels que melhor representem os padrões considerados;
- Aplicações: o método poderia ser aplicado para resolver outros problemas de processamento de imagens, incluindo problemas com imagens em níveis de cinza e coloridas. Nestes casos, além da restrição de resolução espacial pode-se explorar restrições no intervalo de níveis de cinza ou de cores das imagens [6, 7, 43];
- Reconhecimento de dígitos manuscritos: ainda no campo das aplicações, outra possível direção seria o refinamento do classificador de dígitos manuscritos através do aumento do conjunto de treinamento pela inclusão de dígitos distorcidos [39].

Apêndice A

Pirâmides de Imagens

Um dos tópicos que investigamos neste trabalho foi a utilização de mapeamentos de resolução da teoria de pirâmides de imagens (Seção 3.3) no contexto do projeto multirresolução de W -operadores. Para isto, fizemos um levantamento de alguns mapeamentos encontrados na literatura, e implementamos vários deles. Neste apêndice faremos uma revisão dos principais mapeamentos estudados. Inicialmente, introduziremos alguns conceitos básicos da teoria de Morfologia Matemática [15, 16, 17], e definiremos a notação que será utilizada. Tais conceitos serão importantes para a compreensão dos exemplos, visto que grande parte das pirâmides estudadas são construídas usando operadores morfológicos. Em seguida, os exemplos de mapeamentos são apresentados. Concluímos com alguns exemplos de aplicações encontrados na literatura e uma seção com comentários finais.

A.1 Fundamentos e Notação

Um conjunto $\mathcal{L}$ munido de uma ordem parcial $\leq$ é um *reticulado completo* se todo subconjunto $\mathcal{K}$ de $\mathcal{L}$ tem um *supremo* (menor limitante superior) $\bigvee \mathcal{K}$ e um *ínfimo* (maior limitante inferior) $\bigwedge \mathcal{K}$. Dizemos que $\mathcal{L}$ é uma *cadeia completa* se $\mathcal{L}$ for um reticulado completo em que $x \leq y$ ou $y \leq x$ para todo par $x, y \in \mathcal{L}$. Um exemplo simples de cadeia completa é o conjunto $\mathbb{R} \cup \{-\infty, \infty\}$ com a ordenação usual dos números reais. Se $\mathcal{T}$ é um reticulado completo e E é um conjunto não vazio, o conjunto $Fun(E, \mathcal{T}) = \mathcal{T}^E$ que inclui todas as funções de E em $\mathcal{T}$ é um reticulado completo de acordo com a ordenação pontual

$$x \leq y \text{ se } x(p) \leq y(p), \forall p \in E, x, y \in Fun(E, \mathcal{T}). \quad (\text{A.1})$$

Utilizaremos a notação $Fun(E, \mathcal{T})$ para representar os sinais cujo domínio é E e cujos valores estão em $\mathcal{T}$. O menor elemento de $\mathcal{T}$ ($\bigwedge \mathcal{T}$) é denotado por $\perp$, e o maior elemento de $\mathcal{T}$ ($\bigvee \mathcal{T}$) é denotado por $\top$.

Ao considerarmos sinais d-dimensionais, estamos interessados no caso em que E é o espaço discreto $\mathbb{Z}^d$ de dimensão d . Dois operadores morfológicos básicos em $Fun(\mathbb{Z}^d, \mathcal{T})$ são a *dilatação* δ_A e a *erosão* ε_A :

$$\delta_A(x)(n) = (x \oplus A)(n) = \bigvee_{k \in A} x(n - k) \quad (\text{A.2})$$

$$\varepsilon_A(x)(n) = (x \ominus A)(n) = \bigwedge_{k \in A} x(n + k), \quad (\text{A.3})$$

onde $A \subseteq \mathbb{Z}^d$ é um conjunto denominado *elemento estruturante*. Ao considerarmos sinais binários, a representação destes por funções de E em $\{0, 1\}$ é equivalente à representação por subconjuntos de E [44]. Neste caso, a dilatação e a erosão são equivalentes à adição [45] e à subtração de Minkowski [46], de onde vem o uso da notação $\oplus$ e $\ominus$. Se b é uma função de domínio em A e imagem em $\mathcal{T}$, podemos estender a definição de δ_A e ε_A para

$$\delta_A(x)(n) = (x \oplus A)(n) = \bigvee_{k \in A} x(n - k) \dot{+} b(k) \quad (\text{A.4})$$

$$\varepsilon_A(x)(n) = (x \ominus A)(n) = \bigwedge_{k \in A} x(n + k) \dot{-} b(k), \quad (\text{A.5})$$

onde

$$t \dot{+} v = \begin{cases} \perp & \text{se } t = \perp, \\ \perp & \text{se } t > \perp \text{ e } t + v \leq \perp, \\ t + v & \text{se } t > \perp \text{ e } \perp \leq t + v \leq \top, \\ \top & \text{se } t > \perp \text{ e } t + v \geq \top. \end{cases} \quad (\text{A.6})$$

$$t \dot{-} v = \begin{cases} \perp & \text{se } t < \top \text{ e } t - v \leq \perp, \\ t - v & \text{se } t < \top \text{ e } \perp \leq t - v \leq \top, \\ \top & \text{se } t < \top \text{ e } t - v > \top, \\ \top & \text{se } t = \top. \end{cases} \quad (\text{A.7})$$

Neste caso, dizemos que A é um elemento estruturante *não-flat*.

O operador identidade em $\mathcal{L}$ é denotado por id . Se ψ é um operador definido de $\mathcal{L}$ em $\mathcal{L}$, então adotaremos a convenção $\psi^0 = \text{id}$ e denotaremos por ψ^j , para $j > 0$ a composição de operadores $\psi \cdots \psi$ (j vezes). Um operador ψ que satisfaz $\psi \leq \text{id}$ é chamado de *anti-extensivo*; um operador ψ que satisfaz $\psi \geq \text{id}$ é chamado de *extensivo*. A partir da dilatação δ_A e da erosão ε_A , definimos os operadores de *abertura* e *fechamento* pelo elemento estruturante A :

$$\gamma_A(x) = x \circ A = \delta_A \varepsilon_A \quad (\text{A.8})$$

$$\phi_A(x) = x \bullet A = \varepsilon_A \delta_A. \quad (\text{A.9})$$

A abertura é um operador anti-extensivo, enquanto o fechamento é um operador extensivo.

Na prática, ao trabalharmos com imagens digitais consideramos o reticulado $Fun(E, \mathcal{T})$ munido da ordenação pontual, onde o conjunto E é um subconjunto finito de $\mathbb{Z}^2$; usualmente, E é uma grade retangular. O conjunto $\mathcal{T}$ representa o intervalo de valores que os pixels de uma imagem podem assumir. Por exemplo, em imagens binárias $\mathcal{T} = \{0, 1\}$, e em imagens em níveis de cinza $\mathcal{T} = \{0, \dots, N-1\}$, onde N é o número de diferentes níveis de cinza que a imagem pode ter (em geral, $N = 2^k$).

A.2 Exemplos

Nesta seção, veremos alguns exemplos de decomposições piramidais encontradas na literatura. A maior parte dos exemplos pode ser encontrada no trabalho de Goutsias e Heijmans [23, 24, 25, 26], que inclui diversos exemplos de pirâmides de imagens.

A.2.1 Pirâmide Usando Amostragem Diádica Simples

Suponha que os espaços de sinais são tais que $V_j = Fun(E, \mathcal{T})$, onde $E \subseteq \mathbb{Z}^d$ e $\mathcal{T}$ é um conjunto arbitrário, e seja t um elemento fixo de $\mathcal{T}$. A pirâmide é dada pelos operadores $\sigma^\dagger : V_j \rightarrow V_{j+1}$ e $\sigma_t^\dagger : V_{j+1} \rightarrow V_j$:

$$\sigma^\dagger(x)(n) = x(2n) \tag{A.10}$$

$$\sigma_t^\dagger(x)(2n) = x(n) \text{ e } \sigma_t^\dagger(x)(m) = t, \text{ se } m \notin 2\mathbb{Z}^d, \tag{A.11}$$

onde $2\mathbb{Z}^d$ denota os vetores de $\mathbb{Z}^d$ cujas coordenadas são pares. Os operadores são os mesmos em todos os níveis da pirâmide.

A.2.2 Pirâmide de Burt-Adelson

Em [47], Burt e Adelson propuseram uma estrutura piramidal em que os operadores de análise são compostos por duas etapas:

- filtragem passa-baixas usando uma máscara de tamanho 5×5 cujo formato é similar ao da distribuição Gaussiana de probabilidade, com o objetivo de eliminar altas frequências;
- amostragem diádica.

Ou seja,

$$\psi^\dagger(x)(m, n) = \sum_{i=-2}^2 \sum_{j=-2}^2 w(i, j) x(2m + i, 2n + j), \quad (\text{A.12})$$

onde w é o *kernel* da convolução. Por simplicidade, podemos supor que w é separável, isto é, $w(i, j) = \hat{w}(i)\hat{w}(j)$, onde $\hat{w}$ é uma máscara unidimensional de tamanho 5. $\hat{w}$ deve respeitar as seguintes restrições:

- $\hat{w}$ é normalizada, ou seja, $\sum_{i=-2}^2 \hat{w}(i) = 1$;
- $\hat{w}$ é simétrica, isto é, $\hat{w}(i) = \hat{w}(-i)$ para $i = 0, 1, 2$;
- Sejam $\hat{w}(0) = a$, $\hat{w}(-1) = \hat{w}(1) = b$ e $\hat{w}(-2) = \hat{w}(2) = c$. A condição de *igual contribuição* requer que $a + 2c = 2b$.

As três restrições são satisfeitas quando $\hat{w}(0) = a$, $\hat{w}(-1) = \hat{w}(1) = \frac{1}{4}$ e $\hat{w}(-2) = \hat{w}(2) = \frac{1}{4} - \frac{a}{2}$. A Figura A.1, do artigo [47], mostra as funções equivalentes às máscaras obtidas para alguns valores de a .

Figura A.1: Funções equivalentes às máscaras.

A operação de síntese é feita por interpolação e filtragem. O operador é dado por:

$$\psi^\dagger(x)(m, n) = 4 \sum_{i=-2}^2 \sum_{j=-2}^2 w(i, j) x\left(\frac{m-i}{2}, \frac{n-j}{2}\right), \quad (\text{A.13})$$

onde apenas os termos em que $\frac{m-i}{2}$ e $\frac{n-j}{2}$ são inteiros são incluídos na soma.

Os autores deram o nome de *pirâmide Gaussiana* à seqüência $\{x_j\}$, $j \geq 0$ de sinais de aproximação, e o nome de *pirâmide do Laplaciano* à seqüência $\{y_j\}$, $j \geq 1$ de sinais de detalhe. A Figura A.2 ilustra as pirâmides Gaussiana e do Laplaciano geradas a partir de uma imagem em níveis de cinza de 256×256 pixels, com $a = 0.4$. Podemos ver que as imagens geradas apresentam o borramento característico da filtragem passa-baixas. No artigo [47] é sugerido que esta pirâmide possui propriedades interessantes em aplicações de compressão de imagens e transmissão progressiva, que serão discutidas na Seção A.3.

Figura A.2: Pirâmide de Burt-Adelson. A escala foi reduzida para facilitar a visualização, e as imagens de detalhe foram mapeadas para o intervalo $[0, 255]$.

A.2.3 Pirâmide de Toet

Seja $\mathcal{T}$ uma cadeia completa, e suponha que os espaços de sinais são dados por $V_j = \text{Fun}(\mathbb{Z}^d, \mathcal{T})$. Considere os operadores $\sigma^\dagger$ e $\sigma_t^\dagger$ dados pelas equações (A.10) e (A.11). Defina

$$\psi_j^\dagger = \phi_A \gamma_A \sigma^\dagger \text{ e } \psi_j^\dagger = \phi_A \gamma_A \sigma_\top^\dagger, \quad j \geq 0, \quad (\text{A.14})$$

onde γ_A e ϕ_A são a abertura e o fechamento pelo elemento estruturante $A = \{0, 1\}^d$, e $\top$ é o maior elemento de $\mathcal{T}$. A decomposição piramidal resultante é conhecida como *pirâmide de Toet* [48].

A.2.4 Pirâmide da Mediana

Seja $\mathcal{T}$ uma cadeia completa, e suponha que $V_j = \text{Fun}(\mathbb{Z}, \mathcal{T})$, para todo j . Defina os seguintes operadores de análise e síntese para todo nível j :

$$\psi^\dagger(x)(n) = \text{mediana}\{x(2n-1), x(2n), x(2n+1)\} \quad (\text{A.15})$$

$$\psi^\dagger(x)(2n) = \psi^\dagger(x)(2n+1) = x(n). \quad (\text{A.16})$$

A pirâmide de sinais unidimensionais gerada pelos operadores acima é chamada de *pirâmide da mediana*. Uma alternativa para a pirâmide da mediana é:

$$\psi^\dagger(x)(n) = \begin{cases} x(2n), & \text{se } x(2n-1) \wedge x(2n) \wedge x(2n+1) = x(2n) \\ \text{mediana}\{x(2n-1), x(2n), x(2n+1)\}, & \text{caso contrário} \end{cases} \quad (\text{A.17})$$

$$\psi^\dagger(x)(2n) = x(n), \quad \psi^\dagger(x)(2n+1) = x(n) \vee x(n+1). \quad (\text{A.18})$$

Esta pirâmide gera melhores aproximações $\hat{x} = \psi^\dagger \psi^\dagger(x)$ de x em relação às geradas pela pirâmide anterior, já que mais informação é usada para obter $\psi^\dagger(x)(2n+1)$.

No caso bidimensional, se A é o quadrado 3×3 com centro na origem, a pirâmide da mediana é definida por:

$$\psi^\dagger(x)(m, n) = \text{mediana}\{x(2m+k, 2n+l) \mid (k, l) \in A\} \quad (\text{A.19})$$

$$\psi^\dagger(x)(2m, 2n) = x(m, n) \quad (\text{A.20})$$

$$\psi^\dagger(x)(2m, 2n+1) = x(m, n) \wedge x(m, n+1) \quad (\text{A.21})$$

$$\psi^\dagger(x)(2m+1, 2n) = x(m, n) \wedge x(m+1, n) \quad (\text{A.22})$$

$$\psi^\dagger(x)(2m+1, 2n+1) = x(m, n) \vee x(m, n+1) \vee x(m+1, n+1) \vee x(m+1, n). \quad (\text{A.23})$$

Podemos citar propriedades interessantes desta pirâmide: a preservação de detalhes e a geração de

decomposições que podem ser comprimidas de forma eficiente [49]. A Figura A.3 ilustra a decomposição de uma imagem em níveis de cinza usando a pirâmide da mediana.

Figura A.3: Pirâmide da mediana. A escala foi reduzida para facilitar a visualização, e as imagens de detalhe foram mapeadas para o intervalo $[0, 255]$.

A.2.5 Pirâmide de Haar

Suponha que, para todo $j \geq 0$, $V_j = \ell^2(\mathbb{Z})$, o espaço das seqüências de valores reais $(\dots, x(-1), x(0), x(1), \dots)$ com $\sum_{n=-\infty}^{\infty} |x(n)|^2 < \infty$. Os operadores

$$\psi^\uparrow(x)(n) = \frac{1}{2}(x(2n) + x(2n+1)) \quad (\text{A.24})$$

$$\psi^\downarrow(x)(2n) = \psi^\downarrow(x)(2n+1) = x(n) \quad (\text{A.25})$$

geram um esquema de decomposição de sinais unidimensionais chamado de *Pirâmide de Haar*. Os operadores de análise e síntese coincidem com os filtros passa-baixas associados à *wavelet* de Haar [41, 42].

A versão equivalente da pirâmide de Haar para o caso de sinais bidimensionais é dada por

$$\psi^\dagger(x)(m, n) = \frac{1}{4}(x(2m, 2n) + x(2m, 2n + 1) + x(2m + 1, 2n) + x(2m + 1, 2n + 1)) \quad (\text{A.26})$$

$$\begin{aligned} \psi^\dagger(x)(2m, 2n) &= \psi^\dagger(x)(2m + 1, 2n) = \\ \psi^\dagger(x)(2m, 2n + 1) &= \psi^\dagger(x)(2m + 1, 2n + 1) = x(m, n). \end{aligned} \quad (\text{A.27})$$

A.2.6 Pirâmide de Haar Morfológica

Seja $\mathcal{T}$ uma cadeia completa, e considere uma pirâmide para a qual $V_j = \text{Fun}(\mathbb{Z}, \mathcal{T})$, para todo j , e os mesmos operadores de análise e síntese são usados em cada nível j :

$$\psi^\dagger(x)(n) = x(2n) \wedge x(2n + 1) \quad (\text{A.28})$$

$$\psi^\dagger(x)(2n) = \psi^\dagger(x)(2n + 1) = x(n). \quad (\text{A.29})$$

A versão em duas dimensões da pirâmide de Haar morfológica é dada por

$$\psi^\dagger(x)(m, n) = x(2m, 2n) \wedge x(2m, 2n + 1) \wedge x(2m + 1, 2n + 1) \wedge x(2m + 1, 2n) \quad (\text{A.30})$$

$$\begin{aligned} \psi^\dagger(x)(2m, 2n) &= \psi^\dagger(x)(2m, 2n + 1) = \\ \psi^\dagger(x)(2m + 1, 2n + 1) &= \psi^\dagger(x)(2m + 1, 2n) = x(m, n). \end{aligned} \quad (\text{A.31})$$

Esta pirâmide é “similar” à pirâmide de Haar linear apresentada no exemplo anterior, mas utiliza operadores morfológicos (portanto, é não-linear). Ela possui uma propriedade interessante para aplicações de compressão e codificação de imagens: $\psi_j^\dagger \psi_j^\dagger \leq \text{id}$, isto é, $\psi_j^\dagger \psi_j^\dagger$ é anti-extensivo. Isto faz com que os sinais de detalhe sejam sempre não-negativos, permitindo que um bit de informação seja economizado.

A.2.7 Pirâmide de Heijmans-Toet

No caso unidimensional, considere a pirâmide dada por:

$$\psi^\dagger(x)(n) = x(2n - 1) \wedge x(2n) \wedge x(2n + 1) \quad (\text{A.32})$$

$$\psi^\dagger(x)(2n) = x(n) \text{ e } \psi^\dagger(x)(2n + 1) = x(n) \vee x(n + 1). \quad (\text{A.33})$$

Esta decomposição, sugerida por Heijmans e Toet [50], gera uma versão simétrica da pirâmide de Haar

morfológica. A versão bidimensional é dada por:

$$\psi^\dagger(x)(m, n) = \bigwedge_{-1 \leq k, l \leq 1} x(2m + k, 2n + l) \quad (\text{A.34})$$

$$\psi^\downarrow(x)(2m, 2n) = x(m, n) \quad (\text{A.35})$$

$$\psi^\downarrow(x)(2m, 2n + 1) = x(m, n) \vee x(m, n + 1) \quad (\text{A.36})$$

$$\psi^\downarrow(x)(2m + 1, 2n) = x(m, n) \vee x(m + 1, n) \quad (\text{A.37})$$

$$\psi^\downarrow(x)(2m + 1, 2n + 1) = x(m, n) \vee x(m, n + 1) \vee x(m + 1, n) \vee x(m + 1, n + 1). \quad (\text{A.38})$$

A Figura A.4 ilustra esta decomposição em uma imagem em níveis de cinza. Assim como a pirâmide de Haar morfológica, esta pirâmide também tem a propriedade de gerar sinais de detalhe não-negativos.

A.2.8 Pirâmide de Sun-Maragos

Suponha que os sinais são unidimensionais, isto é, $V_j = \text{Fun}(\mathbb{Z}, \mathcal{T})$ para todo j , e os mesmos operadores de análise e síntese são usados em todos os níveis j . Os operadores

$$\psi^\dagger(x)(n) = (x \circ A)(2n) \quad (\text{A.39})$$

$$\psi^\downarrow(x)(2n) = x(n) \text{ e } \psi^\downarrow(x)(2n + 1) = x(n) \vee x(n + 1), \quad (\text{A.40})$$

com $A = \{-1, 0, 1\}$ e $x \circ A = \delta_{A \in A}(x)$ formam um esquema de decomposição piramidal conhecido como *pirâmide morfológica de Sun-Maragos* [51].

É fácil estender a definição dos operadores de análise e síntese para sinais de dimensões maiores. A Figura A.5 ilustra a aplicação da pirâmide de Sun-Maragos em uma imagem binária, onde o elemento estruturante A é o quadrado 3×3 com centro na origem e a operação de subtração em $\mathcal{T} = \{0, 1\}$ utilizada é o “ou exclusivo”.

A.2.9 Pirâmide de Aberturas ou Fechamentos

No artigo [52], pirâmides construídas utilizando aberturas ou fechamentos morfológicos em combinação com amostragem são usadas para fazer compressão de imagens (ver Seção A.3.1). Se os espaços de imagens V_j são iguais a $\text{Fun}(E, \mathcal{T})$, onde $E \subseteq \mathbb{Z}^2$ e o conjunto de níveis de cinza $\mathcal{T}$ é igual a $\{0, \dots, 2^k - 1\}$, a

Figura A.4: Pirâmide de Heijmans-Toet. A escala foi reduzida para facilitar a visualização.

pirâmide de aberturas é dada por:

$$\psi^\dagger = \sigma^\dagger \gamma_B \quad (\text{A.41})$$

$$\psi^\downarrow = \phi_B \sigma_0^\downarrow, \quad (\text{A.42})$$

onde $\sigma^\dagger$ e $\sigma_0^\downarrow$ são os operadores descritos pelas equações (A.10) e (A.11), e γ_B e ϕ_B denotam a abertura e o fechamento morfológico pelo elemento estruturante convexo B . Desta forma, o operador de análise consiste em aplicar uma abertura morfológica e em seguida fazer a amostragem, enquanto o operador de síntese realiza uma interpolação (inclui zeros entre os pixels) seguida por um fechamento. Os autores do

Figura A.5: Pirâmide de Sun-Maragos. A escala foi reduzida para facilitar a visualização.

artigo não descrevem explicitamente a formulação da pirâmide de fechamentos, mas sugerem que seja:

$$\psi^\dagger = \sigma^\dagger \phi_B \quad (\text{A.43})$$

$$\psi^\perp = \phi_B \sigma_0^\perp. \quad (\text{A.44})$$

A utilização do seguinte elemento estruturante não-flat é sugerida no artigo:

$$B = \begin{bmatrix} -a & -b & -a \\ -b & 0 & -b \\ -a & -b & -a \end{bmatrix}, \quad (\text{A.45})$$

onde a e b são inteiros positivos, $a \leq b$ e a origem é o ponto de valor zero. Como o valor central é zero, ao aplicar a abertura ou o fechamento em regiões planas da imagem os valores dos níveis de cinza não serão alterados.

A.2.10 Pirâmide Baseada em Amostragem “Quincunx”

Em [40], Heijmans e Goutsias apresentam uma pirâmide para sinais bidimensionais baseada em um esquema de amostragem denominado amostragem *quincunx*. Seja S o conjunto dos pontos inteiros no plano, isto é, $S = \{(s_1, s_2) \mid s_1, s_2 \in \mathbb{Z}\}$, e seja Q o subconjunto de S resultante após a aplicação da amostragem *quincunx* sobre S , ou seja, $Q = \{(q_1, q_2) \mid q_1, q_2 \in \mathbb{Z} \text{ e } q_1 + q_2 \text{ é par}\}$. Além disso, seja $S' \subset Q$ o conjunto resultante da aplicação da amostragem *quincunx* sobre Q : $S' = \{(s'_1, s'_2) \mid s'_1, s'_2 \in 2\mathbb{Z}\}$.

Definimos as seguintes normas em S :

$$\|s\|_1 = |s_1| + |s_2| \text{ e } \|s\|_\infty = \max\{|s_1|, |s_2|\}, \quad (\text{A.46})$$

onde $s = (s_1, s_2) \in S$. Considere as relações binárias

$$s \rightarrow_0 q \text{ sse } \|s - q\|_1 \leq 1 \text{ e } q \rightarrow_1 s' \text{ sse } \|q - s'\|_\infty \leq 1 \quad (\text{A.47})$$

em $S \times Q$ e $Q \times S'$, respectivamente. Tais relações são ilustradas na Figura A.6, do artigo [40].

Figura A.6: Relações de equivalência. Os discos maiores representam os pontos selecionados pela amostragem *quincunx*.

Defina os espaços de sinais $V_0 = \text{Fun}(S, \mathcal{T})$ e $V_1 = \text{Fun}(Q, \mathcal{T})$. Os operadores de análise e síntese são

dados por

$$\psi_0^\uparrow(x)(q) = \bigwedge_{s:s \rightarrow_0 q} x(s) \quad (\text{A.48})$$

$$\psi_0^\downarrow(x)(s) = \bigvee_{q:s \rightarrow_0 q} x(q). \quad (\text{A.49})$$

De forma similar, defina $V_2 = Fun(S', \mathcal{T})$ e os operadores de análise e síntese

$$\psi_1^\uparrow(x)(s') = \bigwedge_{q:q \rightarrow_1 s'} x(q) \quad (\text{A.50})$$

$$\psi_1^\downarrow(x)(q) = \bigvee_{s':q \rightarrow_1 s'} x(s'). \quad (\text{A.51})$$

Como $S' = 2S'$, para obtermos níveis subseqüentes da pirâmide basta repetir o mesmo procedimento para $Q' = 2Q$ e $S'' = 2S'$. A Figura A.7 mostra alguns níveis da transformação pirâmide correspondente, onde os níveis ímpares são exibidos após uma rotação de 45 graus no sentido anti-horário, obtida através da transformação $(r_1, r_2) \rightarrow \frac{1}{2}(r_1 - r_2, r_1 + r_2)$. Esta pirâmide também possui a propriedade de que os sinais de detalhe gerados são não-negativos.

A.2.11 Pirâmide de Filtros Alternados Seqüenciais

Os filtros alternados seqüenciais (*alternating sequential filters*, ou *ASF*) [16] constituem uma classe muito importante de filtros em Morfologia Matemática. Definimos um mapeamento *close-open* como a abertura seguida por um fechamento pelo elemento estruturante B :

$$M_B(X) = (X \circ B) \bullet B. \quad (\text{A.52})$$

Um filtro alternado seqüencial consiste na aplicação iterativa de M_B

$$ASF = M_{B_N} M_{B_{N-1}} \cdots M_{B_1}, \quad (\text{A.53})$$

onde N é um inteiro, $B_N, B_{N-1}, \dots, B_1$ são elementos estruturantes de tamanhos diferentes, e $B_N \supseteq B_{N-1} \supseteq \cdots \supseteq B_1$. Variações podem ser obtidas substituindo o operador *close-open* pelo *open-close* (fechamento seguido de abertura), *close-open-close* (fechamento seguido de abertura, seguida de fechamento) ou *open-close-open* (abertura seguida de fechamento, seguido de abertura).

Em [53, 54], é apresentado um esquema de decomposição piramidal inspirado nos filtros alternados seqüenciais. A etapa de análise consiste em aplicar um filtro *close-open* seguido de uma amostragem

Figura A.7: Pirâmide usando amostragem *quincunx*. Foi feita mudança de escala em algumas imagens para facilitar a visualização.

diádica, enquanto na etapa de síntese é feita uma interpolação seguida por uma dilatação ou um fechamento. O elemento estruturante usado pelo filtro *close-open* é mantido fixo em todos os níveis da pirâmide; como existe amostragem de nível para nível, o resultado obtido é “similar” ao de um filtro

alternado seqüencial. Formalmente, a decomposição é dada por:

$$\psi^\uparrow = \sigma^\uparrow M_B \quad (\text{A.54})$$

$$\psi^\downarrow = \delta_K \sigma_0^\downarrow \text{ ou } \psi^\downarrow = \phi_K \sigma_0^\downarrow, \quad (\text{A.55})$$

onde $\sigma^\uparrow$ e $\sigma_0^\downarrow$ são os operadores descritos pelas equações (A.10) e (A.11), δ_K e ϕ_K denotam a dilatação e o fechamento pelo elemento estruturante K , B é o elemento estruturante usado pelo operador *close-open* e K é o elemento estruturante usado na reconstrução. Para esta pirâmide, K pode ser o quadrado 3×3 com centro na origem. Para uma formulação mais precisa das condições que devem ser satisfeitas por K , o leitor pode consultar [54].

A.2.12 Pirâmide de Difusão Anisotrópica

Uma limitação da pirâmide de Burt-Adelson é o borramento que surge nas imagens em virtude da aplicação do filtro passa-baixas. Tal borramento pode trazer alguns efeitos indesejáveis em algumas aplicações, pois torna difícil a detecção precisa de bordas e pode levar à fusão de objetos. Visando contornar estas limitações, os autores de [55] propõem a utilização de pirâmides não-lineares para identificação de objetos. Uma das alternativas sugeridas é uma pirâmide que usa operadores *open-close* (fechamento seguido de abertura), discutida no exemplo anterior. A outra alternativa consiste na utilização de filtros de difusão anisotrópica [56].

A abordagem de identificação de objetos apresentada no artigo envolve a criação de uma pirâmide, mas sem a necessidade de reconstruir o sinal original x_0 . Desta forma, apenas a pirâmide de imagens $\{x_0, x_1, \dots, x_k\}$ é construída. O operador de análise sugerido é o filtro de difusão anisotrópica seguido por uma amostragem. Ao contrário da filtragem passa-baixas usando uma Gaussiana, tal filtro tem a propriedade de suavizar regiões internas dos objetos enquanto evita interações entre objetos distintos (a suavização é adaptativa, sendo mais intensa no interior dos objetos do que nas bordas) [56].

A.2.13 Pirâmide da Diferença

Em [57], Kresch e Heijmans apresentam um exemplo de pirâmide morfológica interessante para decompor *imagens de diferença*, isto é, imagens obtidas a partir da diferença entre duas funções reais. Por exemplo, se I_A é uma imagem e I_B é uma imagem obtida a partir de um operador de predição da imagem I_A , isto é, um operador que tem como objetivo estimar o valor de I_A , a diferença $I_A - I_B$ pode ser interpretada como um erro de predição (os sinais de detalhe de pirâmides de imagens são exemplos de imagens de diferença). Outro exemplo interessante aparece em processamento de vídeo digital, onde a análise da diferença entre dois quadros sucessivos em uma seqüência permite extrair informações sobre mudanças na localização dos objetos.

A ordenação pontual de imagens (equação (A.1)) não é apropriada para imagens de diferença, pois esta ordenação não trata igualmente valores positivos e negativos. Então, foi proposta a seguinte ordenação, que não apresenta este defeito: $f \leq g \Leftrightarrow f(x, y) = \text{mediana}\{f(x, y), g(x, y), 0\}, \forall x, y \in \mathbb{Z}$. Suponha então que os espaços de sinais $V_j = \text{Fun}(\mathbb{Z}^2, \mathcal{T})$ sejam munidos desta ordenação, e defina o seguinte operador de V_j em V_{j+1} :

$$\varepsilon(x)(m, n) = (0 \vee MIN) \wedge MAX, \quad (\text{A.56})$$

onde

$$MIN = x(2m, 2n) \wedge x(2m + 1, 2n) \wedge x(2m, 2n + 1) \wedge x(2m + 1, 2n + 1), \quad (\text{A.57})$$

$$MAX = x(2m, 2n) \vee x(2m + 1, 2n) \vee x(2m, 2n + 1) \vee x(2m + 1, 2n + 1). \quad (\text{A.58})$$

Defina também o operador δ por

$$\begin{aligned} \delta(x)(2m, 2n) &= \delta(x)(2m + 1, 2n) = \\ \delta(x)(2m, 2n + 1) &= \delta(x)(2m + 1, 2n + 1) = x(m, n). \end{aligned} \quad (\text{A.59})$$

Se considerarmos os operadores de análise e síntese $\psi_j^\dagger = \varepsilon$ e $\psi_j^\downarrow = \varepsilon^j \delta^{j+1} = \delta$ para todo j , obtemos uma decomposição piramidal denominada *pirâmide da diferença*. Em [57] a pirâmide da diferença é comparada a uma pirâmide linear, e observa-se que a primeira preserva características como bordas, ao contrário da segunda. A pirâmide linear também apresenta o efeito indesejado de criar diversos artefatos nas imagens.

A.2.14 Pirâmides com Quantização

Suponha que os níveis de cinza na base de uma pirâmide possam ser representados por no máximo N bits, isto é, o conjunto $\mathcal{T}$ de níveis de cinza é igual a $\mathcal{T}_N = \{0, 1, \dots, 2^N - 1\}$. Suponha também que os espaços de sinais são dados por $V_j = \text{Fun}(E, \mathcal{T}_{N-j})$, e defina [23]

$$\psi_j^\dagger = q_{N-j} \text{ e } \psi_j^\downarrow = d_{N-j}, \quad (\text{A.60})$$

onde $q_{N-j} : V_j \rightarrow V_{j+1}$ e $d_{N-j} : V_{j+1} \rightarrow V_j$ são definidos como

$$q_{N-j}(x)(n) = \left\lfloor \frac{x(n)}{2} \right\rfloor \quad (\text{A.61})$$

$$d_{N-j}(x)(n) = 2 \cdot x(n). \quad (\text{A.62})$$

Ou seja, conforme subimos da base para o topo da pirâmide o tamanho do intervalo de níveis de cinza é dividido por 2. Uma decomposição piramidal formada apenas pelos operadores de análise e síntese de (A.60) pode ser considerada multirresolução no sentido de que a resolução em profundidade das imagens varia de nível para nível da pirâmide.

A.3 Aplicações

Nesta seção, mostraremos algumas aplicações em processamento de imagens e visão computacional em que são usadas pirâmides de imagens.

A.3.1 Compressão e Codificação de Imagens

Burt e Adelson propuseram em [47] um mecanismo de codificação de imagens baseado na decomposição piramidal pelas pirâmides Gaussiana e do Laplaciano. O método consiste em aplicar a transformação pirâmide sobre a imagem $x = x_0$ que se deseja codificar, obtendo imagens $\{x_k, y_k, y_{k-1}, \dots, y_1\}$. Em seguida, tais imagens são codificadas visando economizar espaço (pode-se usar técnicas de codificação sem perda de informação, como códigos de Huffman, por exemplo). Para recuperar a imagem original novamente, é feita a decodificação das imagens da pirâmide e em seguida a imagem original é reconstruída utilizando a transformação pirâmide inversa.

Se assumirmos que os valores dos pixels de uma imagem são estatisticamente independentes, então o número mínimo de bits por pixel necessários para codificar a imagem é dado pela entropia da distribuição de valores dos pixels. Por exemplo, se estamos trabalhando com imagens cujos níveis de cinza estão em $\{0, \dots, 255\}$, a entropia de uma imagem [47, 52] é dada por

$$H = - \sum_{i=0}^{255} f(i) \log_2 f(i), \quad (\text{A.63})$$

onde $f(i)$ denota a frequência de ocorrência do nível de cinza i na imagem. A codificação das imagens de detalhe (neste caso, das imagens da pirâmide do Laplaciano) ao invés da imagem original é justificada pelo fato de tais imagens serem a diferença entre uma imagem e o resultado de um processo de predição desta. Com isso, grande parte da correlação entre os pixels é removida, os valores dos pixels das imagens de detalhe concentram-se ao redor do zero, e a entropia dessas imagens é menor [47].

Burt e Adelson também sugerem o uso de quantização nas imagens de detalhe a fim de reduzir ainda mais a entropia, caso seja satisfatório que a imagem reconstruída seja apenas uma aproximação da original (dependendo do processo de quantização utilizado, este pode introduzir erros irrecuperáveis). Outra otimização que pode ser feita em termos de espaço, mas que também invalida a propriedade de reconstrução perfeita é a remoção do primeiro sinal de detalhe y_1 da pirâmide.

A discussão acima mostra que, para obter uma boa taxa de compressão, é interessante gerar uma pirâmide cujas imagens de detalhe possuem baixa entropia. Assim, existem na literatura alguns trabalhos que exploram esta idéia para obter melhorias na taxa de compressão. Entre eles, podemos citar [52], que sugere que a utilização de pirâmides de aberturas ou fechamentos morfológicos (Seção A.2.9) gera imagens de detalhe com menor entropia.

Já em [58] os autores observam o processo de geração de uma pirâmide segundo a formulação clássica: a análise consiste em uma filtragem linear seguida de amostragem diádica, enquanto a síntese é constituída por interpolação (zeros são incluídos entre as amostras) seguida por filtragem linear. Então, é sugerido que, dentro desta classe de pirâmides, se os filtros lineares dos operadores de análise e síntese forem as convoluções pelas máscaras g e h abaixo, respectivamente, então a pirâmide de imagens de detalhe tem entropia mínima:

$$g = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.64})$$

$$h = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 \\ 0 & 1/2 & 1 & 1/2 & 0 \\ 0 & 1/4 & 1/2 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.65})$$

A decomposição obtida é chamada pelos autores de pirâmide de entropia minimal (*minimal entropy pyramid*, ou *MEP*).

A.3.2 Transmissão Progressiva

As pirâmides de imagens podem ser usadas de forma natural em aplicações de transmissão progressiva de uma imagem [47]. Neste tipo de transmissão, primeiro é enviada uma representação grosseira da imagem, a fim de que o receptor tenha uma idéia sobre o seu conteúdo. Em seguida, transmissões subsequentes proporcionam detalhes da imagem em resolução cada vez mais fina. Se desejar, o receptor pode interromper a transmissão assim que a imagem alcançar um nível satisfatório de detalhe.

Suponha que a imagem x_0 será transmitida desta forma. No nó de origem da transmissão, x_0 é decomposta em imagens $\{x_k, y_k, y_{k-1}, \dots, y_1\}$ pela aplicação da transformação pirâmide. Então, primeiramente é transmitida a imagem x_k de menor resolução, e em seguida as imagens de detalhe $y_k, y_{k-1}, \dots, y_1$ são en-

viadas progressivamente. No nó receptor, a reconstrução da imagem x_0 também é feita progressivamente, conforme as imagens y_{j+1} são recebidas. A cada recepção de uma imagem y_{j+1} , é feita a aplicação do operador de síntese $\psi_j^l(x_{j+1})$ (onde x_{j+1} é a imagem que já foi reconstruída até o momento) e o resultado é adicionado a y_{j+1} para obter x_j . A imagem x_j passa por um processo de interpolação antes de ser exibida para o usuário com a resolução de x_0 . Com isso, a imagem exibida para o usuário no nó receptor é atualizada progressivamente, até que todas as imagens de detalhe sejam transmitidas, resultando na imagem x_0 . Ou seja, a imagem vai “entrando em foco” gradativamente. A Figura A.8, do artigo [47], ilustra este fato.

Figura A.8: Transmissão progressiva. A imagem “entra em foco” gradativamente.

Também é interessante notar que técnicas de codificação de imagens podem ser usadas sobre as imagens transmitidas, para reduzir o tempo de transmissão.

A.3.3 Localização de Objetos

Na literatura encontramos alguns trabalhos que utilizam pirâmides de imagens em aplicações de localização de objetos [28, 55, 59]. Neste tipo de aplicação o objetivo é determinar a posição de um ou mais objetos de interesse em uma dada imagem. O uso de pirâmides proporciona ganhos em eficiência computacional através de estratégias de busca “*coarse-to-fine*” (da resolução mais grosseira em direção à resolução mais fina). Em linhas gerais, este tipo de estratégia consiste em, de posse da representação piramidal de uma imagem (não é necessária a geração das imagens de detalhe, basta a seqüência de imagens $\{x_0, \dots, x_k\}$, para algum k em que ainda seja possível reconhecer as características da imagem), iniciar a busca pelo objeto de interesse na imagem de menor resolução, obtendo assim uma estimativa da posição do objeto. Tal estimativa é usada para guiar a busca no nível seguinte (de maior resolução) na pirâmide: o objeto é procurado na região correspondente à localização obtida na resolução mais grosseira (eventualmente incluindo uma pequena vizinhança ao redor desta região). Repete-se o procedimento até chegarmos à base da pirâmide, que contém a imagem de maior resolução. Desta forma, conforme pros seguimos do topo para a base da pirâmide a estimativa da localização do objeto é refinada, e a procura apenas nas regiões correspondentes aos pontos obtidos faz com que computacionalmente este método seja mais eficiente do que fazer uma única busca na imagem de maior resolução.

O artigo [55] explora pirâmides não-lineares em conjunto com uma estratégia de *template matching* baseado em bordas para identificar objetos. Os autores mostram resultados experimentais comparando

pirâmides não-lineares com a pirâmide linear de Burt e Adelson e concluem que neste tipo de aplicação as decomposições que utilizam operadores não-lineares levam a resultados melhores com custo computacional equivalente, devido à preservação de estruturas como bordas. Já em [59], é construída uma pirâmide a partir da imagem do objeto de interesse e uma pirâmide a partir da imagem onde se quer fazer a busca, e um procedimento de *template matching* entre imagens das duas pirâmides é realizado. A decomposição piramidal utilizada é a pirâmide de Haar, e uma variação da estratégia de busca *coarse-to-fine* baseada em um procedimento de descida do gradiente é sugerida. Outro trabalho que usa estratégias de busca *coarse-to-fine* é [28], onde é apresentado um algoritmo para rastreamento (*tracking*) de objetos em vídeo que faz decomposições piramidais de todos os quadros da seqüência e, para cada dois quadros sucessivos, trabalha com a informação dada pelas duas pirâmides sucessivas para determinar a mudança de localização de um objeto de forma multirresolução.

A.4 Algumas Considerações

As pirâmides de imagens constituem uma técnica bastante importante na criação de decomposições multirresolução em visão computacional e processamento de imagens. Neste apêndice, vimos diversos exemplos de decomposições piramidais. As diferentes propriedades apresentadas pelas decomposições fazem com que para uma dada aplicação alguns tipos de pirâmides possam ser mais adequados do que outros. Cada pirâmide define uma maneira de fazer mapeamentos entre imagens de diferentes resoluções, e a escolha da melhor pirâmide para uma determinada aplicação pode ser uma tarefa bastante complicada.

As primeiras pirâmides propostas eram lineares. Tais pirâmides são importantes em diversas aplicações, mas quando é necessário preservar informação geométrica (por exemplo, bordas) entre as diversas resoluções da pirâmide os resultados obtidos podem ser insatisfatórios devido ao borramento gerado pelos filtros passa-baixas. Isto serviu como motivação para o estudo de esquemas de decomposição não-lineares, entre eles as pirâmides morfológicas. Além das propriedades citadas nos exemplos apresentados, as pirâmides morfológicas são interessantes pelo fato de sempre mapearem valores de pixels inteiros em valores inteiros, e nunca gerarem imagens de aproximação $\{x_j\}$, $j \geq 1$ com valores fora do intervalo de níveis de cinza da imagem original x_0 [23].

A representação de uma imagem ou sinal por meio de uma estrutura piramidal é considerada redundante, pois sem a utilização de nenhum método de compressão a quantidade de memória necessária para armazenar os sinais x_k e $y_1, y_2, \dots, y_k$ é maior do que a necessária para armazenar o sinal original x_0 . Isto é uma consequência do fato de que, para cada j , o domínio dos sinais em W_{j+1} é igual ao domínio dos sinais em V_j .

Apêndice B

Publicação Relacionada ao Tema da Dissertação

Alguns resultados preliminares deste trabalho foram publicados no XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2005). No artigo intitulado “A Maximum-likelihood Approach for Multiresolution W-operator Design” [60], propusemos a metodologia de estimação da distribuição conjunta descrita no Capítulo 4, o uso da entropia condicional para auxiliar na escolha da pirâmide e mostramos alguns resultados experimentais com a nossa base de dados de dígitos manuscritos, usando operadores projetados usando o critério da máxima verossimilhança.

Apêndice C

Lista de Símbolos

W	Janela
E	Retângulo finito (região de uma imagem)
L, L'	Intervalo de níveis de cinza de uma imagem
$\mathcal{P}(E)$	Coleção dos subconjuntos de E
$[a, b]$	Intervalo entre a e b
L^E	Conjunto das funções de E em L
Ψ	Operador (filtro) de imagens
$ W $	Se W é um conjunto, é o número de elementos em W ; Se W é um escalar, denota o valor absoluto
$\mathbf{X}$	Vetor aleatório sobre o espaço de possíveis padrões observados por uma janela
$\mathbf{x}$	Realização de $\mathbf{X}$
$\mathcal{A}_X$	Conjunto de possíveis valores que a variável aleatória X pode assumir
Y	Variável aleatória sobre os valores de saída de um operador
y	Realização de Y
ψ	Função característica de um W -operador
f, g, h	Imagens ou sinais

Φ, Φ'	Espaço de operadores, subconjunto de Φ dado por uma restrição
$l(y, \psi(\mathbf{x}))$	Função de perda
ψ_{opt}	Função característica ótima
ψ'_{opt}	Função característica ótima no espaço restrito
$p(\cdot)$	Função distribuição de probabilidade
$P(\cdot)$	Probabilidade
$E[\cdot]$	Esperança
$\hat{p}$	Estimativa de p
$\Leftrightarrow$	Se e somente se (sse)
$W_0, W_1, \dots, W_r$	Seqüência de janelas de uma pirâmide
$D_0, D_1, \dots, D_r$	Espaços de configurações
$\mathbf{z}, \mathbf{z}_k$	Configuração, em geral usada para resoluções menores
ρ, ρ_{ij}	Mapeamentos de resolução
$\mathcal{X}^0, \dots, \mathcal{X}^r$	Partições aninhadas do espaço D_0
$n_0, n_1, \dots, n_r$	Números de conjuntos nas partições $\mathcal{X}^0, \dots, \mathcal{X}^r$
$\mathcal{X}_j^i$	j -ésimo conjunto da partição $\mathcal{X}^i$
$\sim, \sim_i$	Relações de equivalência
ρ^{-1}	Conjunto inverso de ρ
$C[\mathbf{z}]$	Classe de equivalência
$V_0, V_1, \dots$	Espaços de sinais
$W_1, W_2, \dots$	Espaços de sinais (Seção 3.3 e Apêndice A)
$\psi_j^\dagger, \omega_j^\dagger$	Operadores de análise (Seção 3.3 e Apêndice A)
$\Psi_j^\dagger, \psi_j^\dagger$	Operadores de síntese (Seção 3.3 e Apêndice A)
$x_j, \hat{x}_j, y_j$	Sinal, sinal de aproximação, sinal de detalhe (Seção 3.3 e Apêndice A)
$\mathcal{T}$	Cadeia completa, conjunto de níveis de cinza (Seção 3.3 e Apêndice A)
σ	Operador de amostragem (Seção 3.3 e Apêndice A)
$\mathcal{X}$	Partição de D_0 , resultado do algoritmo de estimação (Seção 4.1)
$\mathcal{X}_i$	Conjunto em $\mathcal{X}$

$\gamma(\mathcal{X}), \Gamma(\mathcal{X}_i)$	Distribuição de probabilidade sobre $\mathcal{X}$, probabilidade de um conjunto $\mathcal{X}_i \in \mathcal{X}$
$P(Y \mathcal{X}_i)$	Probabilidade de Y dado que $\mathbf{X} = \mathbf{x}$, para algum $\mathbf{x} \in \mathcal{X}_i$
$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$	Amostras de treinamento (Seção 4.1)
Δ	Pirâmide de janelas
$\mathcal{S}$	Subconjunto não vazio de D_0
$N_{\mathcal{S}}$	Número de vezes em que as configurações em $\mathcal{S}$ aparecem nos exemplos
$T_{\alpha}(n)$	Função limiar (<i>threshold</i>)
α	Parâmetro para o algoritmo de estimação (Seção 4.1); Número mínimo de exemplos para estimar a distribuição condicional (Seção 3.2)
$\mathcal{R}$	Representação da distribuição conjunta (Seção 4.1)
$M_{\mathcal{X}_j^i}$	Massa do conjunto $\mathcal{X}_j^i$ na resolução i (Seção 4.1)
$\mathcal{T}_0, \dots, \mathcal{T}_r$	Tabelas usadas na representação da distribuição e do operador (Seção 4.4)
$\log_b(\cdot)$	Logaritmo na base b
$H(X)$	Entropia da variável aleatória discreta X
$H(Y X = x)$	Entropia condicional de Y dado que $X = x$
$H(Y X)$	Entropia condicional de Y dado X
$a \vee b$	Supremo de a e b (Apêndice A)
$a \wedge b$	Ínfimo de a e b (Apêndice A)
$Fun(E, \mathcal{T})$	Conjunto das funções de E em $\mathcal{T}$ (Apêndice A)
$\perp$	Menor elemento de $\mathcal{T}$ (Apêndice A)
$\top$	Maior elemento de $\mathcal{T}$ (Apêndice A)
$\delta_A(x), x \oplus A$	Dilatação de x pelo elemento estruturante A (Apêndice A)
$\varepsilon_A(x), x \ominus A$	Erosão de x pelo elemento estruturante A (Apêndice A)
$\gamma_A(x), x \circ A$	Abertura de x pelo elemento estruturante A (Apêndice A)
$\phi_A(x), x \bullet A$	Fechamento de x pelo elemento estruturante A (Apêndice A)
id, ψ^0	Operador identidade (Apêndice A)
ψ^j	Composição $\psi \cdots \psi$ do operador ψ (j vezes) (Apêndice A)

Referências Bibliográficas

- [1] E. R. Dougherty, J. Barrera, G. Mozelle, S. Kim, and M. Brun. Multiresolution Analysis for Optimal Binary Filters. *Journal of Mathematical Imaging and Vision*, (14):53–72, 2001.
- [2] J. Barrera, E. R. Dougherty, and N. S. Tomita. Automatic Programming of Binary Morphological Machines by Design of Statistically Optimal Operators in the Context of Computational Learning Theory. *Electronic Imaging*, 6(1):54–67, January 1997.
- [3] J. Barrera, R. Terada, F. S. C. da Silva, and N. S. Tomita. Automatic Programming of Morphological Machines for OCR. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 385–392, Atlanta, GA, May 1996. International Symposium on Mathematical Morphology, Kluwer Academic Publishers.
- [4] R. Hirata Jr., E. R. Dougherty, and J. Barrera. Aperture Filters. *Signal Processing*, 80(4):697–721, April 2000.
- [5] N. S. T. Hirata, E. R. Dougherty, and J. Barrera. A Switching Algorithm for Design of Optimal Increasing Binary Filters Over Large Windows. *Pattern Recognition*, 33(6):1059–1081, June 2000.
- [6] R. Hirata Jr., M. Brun, J. Barrera, and E. R. Dougherty. Multiresolution Design of Aperture Operators. *Journal of Mathematical Imaging and Vision*, 16(3):199–222, 2002.
- [7] R. Hirata Jr. *Projeto de Operadores Morfológicos para Imagens e Sinais - Abordagem de reticulados finitos discretos*. PhD thesis, Instituto de Matemática e Estatística - Universidade de São Paulo, São Paulo, SP - Brasil, Novembro de 2001.
- [8] M. Brun. *Projeto de Operadores Morfológicos Multi-Escala por Otimização Estatística*. PhD thesis, Instituto de Matemática e Estatística - Universidade de São Paulo, São Paulo, SP - Brasil, Julho de 2002.
- [9] J. M. Jolion and A. Rosenfeld. *A Pyramid Framework for Early Vision*. Kluwer Academic Publishers, 1994.

- [10] D. C. Martins-Jr. Redução de Dimensionalidade Utilizando Entropia Condicional Média Aplicada a Problemas de Bioinformática e Processamento de Imagens. Master's thesis, Instituto de Matemática e Estatística - Universidade de São Paulo, 2004.
- [11] D. C. Martins-Jr, R. M. Cesar-Jr, and J. Barrera. W-operator Window Design by Maximization of Training Data Information. In *Proceedings of XVII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 162–169. IEEE Computer Society Press, October 2004.
- [12] D. C. Martins-Jr, R. M. Cesar-Jr, and J. Barrera. Automatic Window Design for Gray-Scale Image Processing Based on Entropy Minimization. In *X Iberoamerican Congress on Pattern Recognition*, pages 813–824, 2005.
- [13] J. Barrera, R. M. Cesar-Jr, D. C. Martins-Jr, R. Z. N. Vêncio, E. F. Merino, M. M. Yamamoto, F. G. Leonardi, C. A. B. Pereira, and H. A. del Portillo. Constructing Probabilistic Genetic Networks of *Plasmodium falciparum* From Dynamical Expression Signals of the Intraerythrocytic Development Cycle. In J. S. Shoemaker and S. M. Lin, editors, *Methods of Microarray Data Analysis V*. Simon, Springer, 2005.
- [14] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, second edition, 2002.
- [15] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
- [16] H. J. A. M. Heijmans. *Morphological Image Operators*. Academic Press, Boston, Massachusetts, 1994.
- [17] J. Serra. *Image Analysis and Mathematical Morphology II: Theoretical Advances*. Academic Press, 1988.
- [18] J. Barrera, E. R. Dougherty, and M. Brun. Hybrid Human-Machine Binary Morphological Operator Design. An Independent Constraint Approach. *Signal Processing*, 80(8):1469–1487, August 2000.
- [19] J. Barrera and G. P. Salas. Set Operations on Closed Intervals and Their Applications to the Automatic Programming of Morphological Machines. *Electronic Imaging*, 5(3):335–352, July 1996.
- [20] E. R. Dougherty. *Random Processes for Image and Signal Processing*. SPIE and IEEE Presses, Bellingham, 1999.
- [21] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2001.
- [22] G. Gamow. *One, two, three ... infinity: facts and speculations of science*. Penguin, 1961.
- [23] J. Goutsias and H. J. A. M. Heijmans. Multiresolution Signal Decomposition Schemes – Part 1: Linear and Morphological Pyramids. Technical report, Centrum voor Wiskunde en Informatica (CWI), 1998.

- [24] J. Goutsias and H. J. A. M. Heijmans. Nonlinear Multiresolution Signal Decomposition Schemes – Part I: Morphological Pyramids. *IEEE Transactions on Image Processing*, 9(11):1862–1876, November 2000.
- [25] J. Goutsias and H. J. A. M. Heijmans. An Axiomatic Approach to Multiresolution Signal Decomposition. In *Proceedings of the IEEE International Conference on Image Processing*, Chicago, Illinois, October 1998.
- [26] H. J. A. M. Heijmans and J. Goutsias. Some Thoughts on Morphological Pyramids and Wavelets. In *Proceedings of the IX European Signal Processing Conference*, Island of Rhodes, Greece, September 1998.
- [27] H. J. A. M. Heijmans and J. Goutsias. *Space, Structure and Randomness – Contributions in Honor of Georges Matheron in the Fields of Geostatistics, Random Sets, and Mathematical Morphology*, volume 183 of *Lecture Notes in Statistics*, chapter Morphological Decomposition Systems with Perfect Reconstruction: From Pyramids to Wavelets, pages 279–314. Springer, 2005.
- [28] J. Z. Zhang and Q. M. J. Wu. A Pyramid Approach to Motion Tracking. *Real-Time Imaging*, 7(6):529–544, December 2001.
- [29] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.
- [30] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [31] N. S. Tomita. Programação Automática de Máquinas Morfológicas Binárias baseada em Aprendizado PAC. Master’s thesis, Instituto de Matemática e Estatística - Universidade de São Paulo, São Paulo, SP - Brasil, março de 1996.
- [32] N. S. Tomita. Programação Automática de Máquinas Morfológicas Binárias baseada em Aprendizado PAC. In *Anais do XXIII Seminário Integrado de Software e Hardware - IX Concurso de Teses e Dissertações - XV Concurso de Trabalhos de Iniciação Científica*, pages 517–525, Recife, PE, Agosto de 1996. XVI Congresso da SBC, Sociedade Brasileira de Computação.
- [33] K. Konstantinides and J. Rasure. The KHOROS Software Development Environment for Image and Signal Processing. *IEEE Transactions on Image Processing*, 3(3):243–252, 1994.
- [34] Projeto VERASS - Verificação de Assinaturas. URL: <<http://www.vision.ime.usp.br/nonlinear/verass/>>. [Acessado em 02/03/2006].
- [35] Base de Dados de Dígitos Manuscritos - Grupo de Visão - IME-USP. URL: <<http://www.vision.ime.usp.br/~daniel/sibgrapi2005/>>. [Acessado em 30/11/2005].

- [36] J. Barrera, M. Brun, R. Terada, and E. R. Dougherty. Boosting OCR Classifier by Optimal Edge Noise Filtering. *ISMM 2000, International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing V*, June 2000.
- [37] Y. LeCun and C. Cortes. The MNIST Database of Handwritten Digits. URL: <http://yann.lecun.com/exdb/mnist/>. [Acessado em 30/11/2005].
- [38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.
- [39] P. Y. Simard, D. Steinkraus, and J. Platt. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 958–962. IEEE Computer Society, 2003.
- [40] H. J. A. M. Heijmans and J. Goutsias. Morphological Pyramids and Wavelets Based on the Quincunx Lattice. *ISMM 2000, International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing V*, June 2000.
- [41] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.
- [42] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, California, 1998.
- [43] F. C. Flores, R. Hirata Jr., J. Barrera, R. A. Lotufo, and F. Meyer. Morphological Operators for Segmentation of Color Sequences. In *Proceedings of SIBGRAPI'2000*, pages 300–307, Gramado, Brazil, October 2000.
- [44] G. J. F. Banon and J. Barrera. *Bases da Morfologia Matemática para a Análise de Imagens Binárias*. IX Escola de Computação, Recife, 1994.
- [45] H. Minkowski. Volumen und Oberfläche. *Math. Ann.*, 57:447–495, 1903.
- [46] H. Hadwiger. Minkowskische Addition und Subtraktion beliebiger Punktmengen und die Theoreme von Erhard Schmidt. *Math. Zeitschrift*, 53:210–218, 1950.
- [47] P. J. Burt and E. H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, COM-31(4):532–540, April 1983.
- [48] A. Toet. A Morphological Pyramidal Image Decomposition. *Pattern Recognition Letters*, 9:255–261, 1989.
- [49] X. Song and Y. Neuvo. Image Compression using Nonlinear Pyramid Vector Quantization. *Multidimensional Systems and Signal Processing*, 5:133–149, 1994.
- [50] H. J. A. M. Heijmans and A. Toet. Morphological Sampling. *Computer Vision, Graphics and Image Processing: Image Understanding*, 54:384–400, 1991.

-
- [51] F.-K. Sun and P. Maragos. Experiments on Image Compression using Morphological Pyramids. In *Visual Communications and Image Processing '92*, volume 1199 of *SPIE Proceedings*, Philadelphia, Pennsylvania, 1989.
- [52] H. Ching-Han and C.-C. J. Kuo. Multiresolution Image Decomposition and Compression Using Mathematical Morphology. In *1993 Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 21–25, Pacific Grove, CA, USA, November 1993.
- [53] A. Morales and R. Acharya. An Image Pyramid with Morphological Operators. In *Proceedings of CVPR '91*, pages 526–531, 1991.
- [54] A. Morales, R. Acharya, and S.-J. Ko. Morphological Pyramids with Alternating Sequential Filters. *IEEE Transactions on Image Processing*, 4(7):965–977, July 1995.
- [55] C. A. Segall, W. Chen, and S. T. Acton. Nonlinear Pyramids for Object Identification. In *Thirtieth Asilomar Conference on Systems, Signals and Computers*, Pacific Grove, California, November 1996.
- [56] P. Perona and J. Malik. Scale-space and Edge Detection using Anisotropic Diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.
- [57] R. Kresch and H. J. A. M. Heijmans. Adjunctions in Pyramids, Curve Evolution and Scale-Spaces. *International Journal on Computer Vision*, pages 139–151, 2003.
- [58] D. Houlding and J. Vaisey. Low Entropy Image Pyramids for Efficient Lossless Coding. *IEEE Transactions on Image Processing*, 4(8):1150–1153, August 1995.
- [59] J. MacLean and J. Tsotsos. Fast Pattern Recognition Using Gradient-Descent Search in an Image Pyramid. In *Proceedings of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 2000.
- [60] D. A. Vaquero, J. Barrera, and R. Hirata Jr. A Maximum-likelihood Approach for Multiresolution W-operator Design. In *Proceedings of XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 71–78. IEEE Computer Society, October 2005.