Programação Automática de Máquinas Morfológicas por Aprendizado Computacional

Aluno: Caetano Jimenez Carezzato
Orientador: Junior Barrera

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

21/Agosto/1998

Bolsa CNPQ

http://www.linux.ime.usp.br/˜caetano/ic/relatorios/1998-1998/

Resumo

A Morfologia Matemática é uma teoria útil para modelar transformações entre imagens e, por isso, vem sendo empregada com bastante sucesso para resolver diversos problemas de processamento de imagens. Entretanto, para se obter bons resultados em aplicações, o usuário deve possuir conhecimentos específicos a cerca da técnica empregada, bem como uma vasta experiência na área. Para simplificar a tarefa do usuário, vem sendo estudadas técnicas de projeto automático de operadores (i.e., transformações) morfológicos. Neste trabalho estudamos a aplicação da técnica de aprendizado Provavelmente Aproximadamente Correto para o projeto de operadores morfológicos (ou, equivalentemente, programas para Máquinas Morfológicas), que resolvem problemas de Análise de Imagens. Em particular, estudamos um método híbrido de projeto de operadores (heurístico-estatístico) que viabiliza a solução de problemas complexos e extensões para melhorar a sua precisão.

Sumário

1 Introdução

A partir de Março deste ano começamos a executar o Cronograma de Atividades. Realizamos a leitura introdutória necessária para a execução do projeto. Devido ao curto tempo decorrido desde o início da bolsa, só a primeira parte da implementação foi executada.

2 Descrição das Atividades

2.1 Leitura

Inicialmente, estudamos o básico do Processamento Morfológico de Imagens, sua representação, relações, operações e operadores, elementos estruturantes, dilatações e erosões, manipulação geométrica, filtros morfológicos, resíduos, gradiente e alguns exemplos de aplicações.

Logo em seguida, passamos a estudar as bases necessárias para a Programação Automática de Máquinas Morfológicas:

- **Elementos da Teoria dos Conjuntos -** Relação de Ordem em Conjuntos, Funções, Elementos da Teoria dos Reticulados, Princípio da Dualidade;
- **Elementos da Teoria de Probabilidades -** Espaço de Probabilidades, Variáveis Aleatórias, noções de Funções Booleanas, Álgebra Booleana, Funções Booleanas, Representação cúbica de uma função Booleana, Minimização Tabular de Quine-McCluskey e Grupo Abeliano;
- **Decomposição de Operadores -** Operadores Elementares, Decomposição Canônica, Decomposição Minimal, Operadores vistos através do Isomorfismo, Relações de Isomorfismo, Relações entre Operadores e Funções Booleanas, Decomposição de Operadores com Propriedades Especiais;
- **Projeto de Operadores -** Conjuntos Aleatórios, Estacionariedade, Transformações Morfológicas de Conjuntos Aleatórios, Projeto de Filtros Morfológicos Ótimos;
- **Modelos de Aprendizado Computacional** - Modelagem do Processo de Aprendizado, Modelo de Aprendizado PAC, Extensão do Modelo PAC;
- **Algoritmos para o Aprendizado de Funções Booleanas** - O algoritmo ISI, Quine X ISI, Variantes do ISI;
- **Geração Automática de Operadores Morfológicos** - Especificação do sistema, Decisão Ótima, Aspectos Práticos;

2.2 Visualização de Exemplos Prontos

Inicialmente instalamos e estudamos a interface de desenvolvimento (Khoros 2.2). Estudamos as Toolboxes já implementadas necessárias para fazer uma pré-filtragem e o Aprendizado Computacional propriamente dito. Estudamos exemplos já prontos, como o reconhecimento de bordas, letras impressas, texturas, eliminação de ruído etc.

3 Exemplo de Aplicação

Após a fase inicial de leituras e estudo de exemplos, resolvemos implementar um reconhecedor de dígitos escritos à mão.

Começamos os experimentos tentando reconhecer o dígito zero.

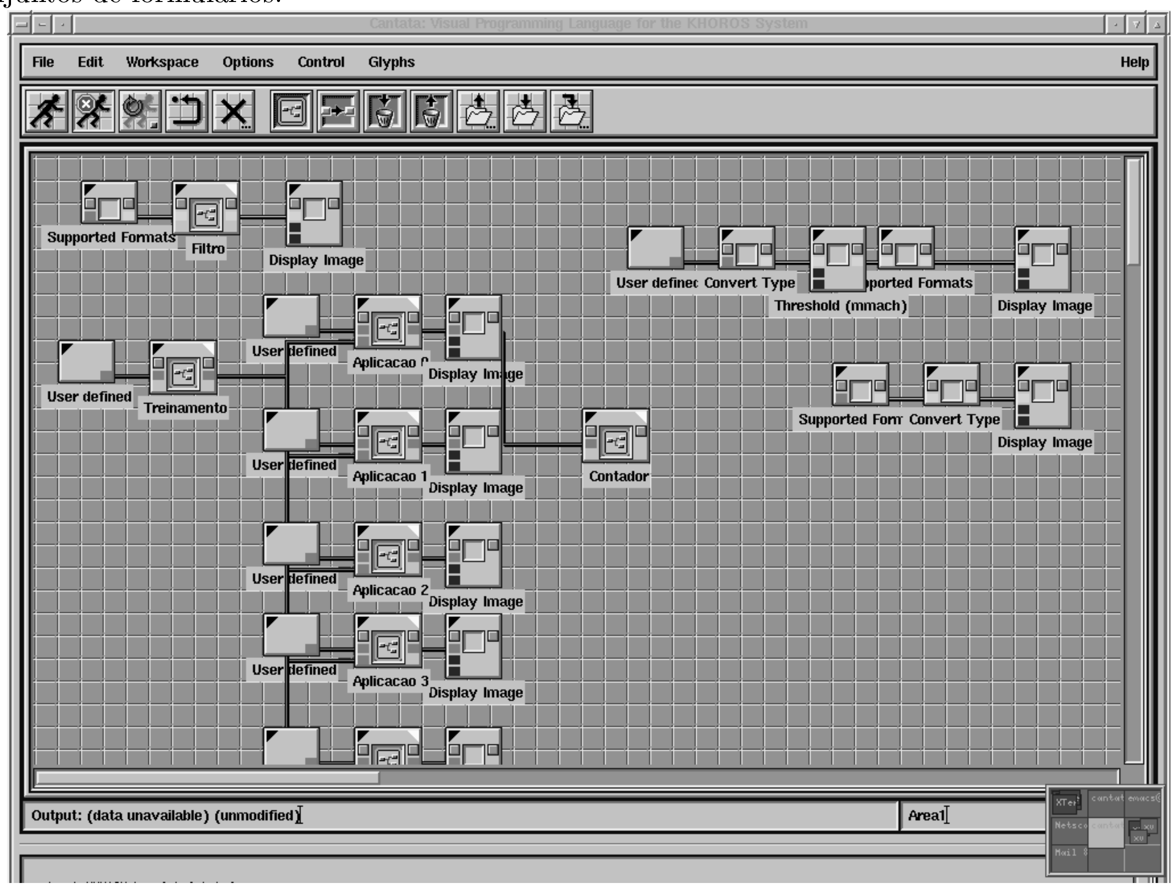
3.1 Implementação de Algoritmos de Aprendizado

3.1.1 Especificações

Utilizamos uma série de formulários já disponíveis (100 ao todo) cada uma com 672 dígitos do mesmo tipo, escrito por diversas pessoas.

Para o nosso aprendizado, escolhemos um subconjunto desses formulários e aplicamos um filtro morfológico. A idéia básica é normalizar as letras das pessoas e reduzir o contexto estatístico.

Após o pré-processamento, realizamos o aprendizado com diversos tamanhos de janela e subconjuntos de formulários.



Exemplo de workspace no khoros2.2.

3.1.2 Aplicações e Testes

Descrição dos resultados obtidos - Para testar os resultados obtidos, utilizamos um subconjunto de folhas. Criamos um procedimento para totalizar os erros. Os resultados foram os seguintes:

Folhas com o dígito	Erros	Erro Percentual
0	264	39.28%
1	1	0,14%
2	44	6,54%
3	38	5,65%
4	35	5,20%
5	38	5,65%
6	57	8,48%
7	29	4,31%
8	73	10,86%
9	46	6,84%

Erro para menos (Não reconheceu o que devia reconhecer): 39,28%
Erro para mais (Reconheceu o que não devia) : 5,96%
Erro total: **9**,29%

Alguns exemplos de aplicação do algoritmo obtido:

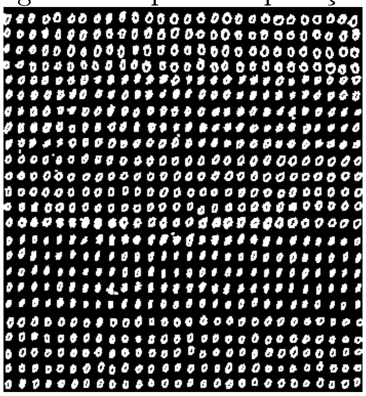
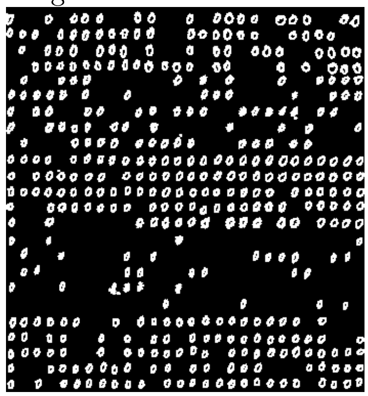
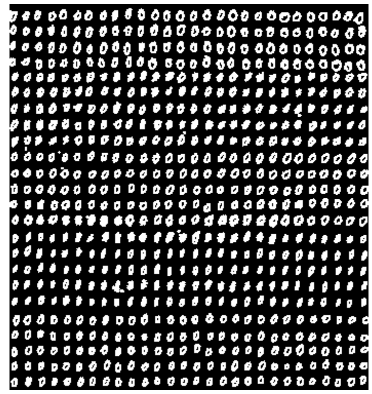
  

Imagem Original | Imagem Obtida | Imagem Ideal

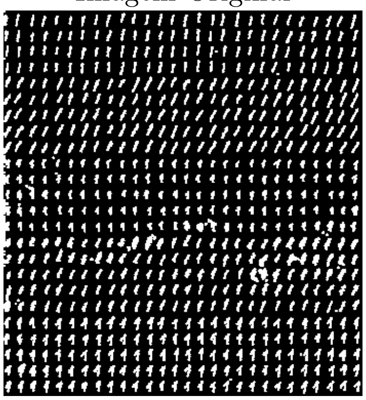
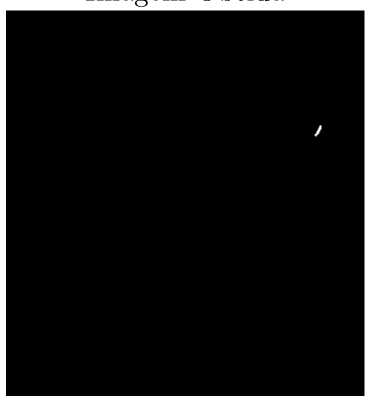
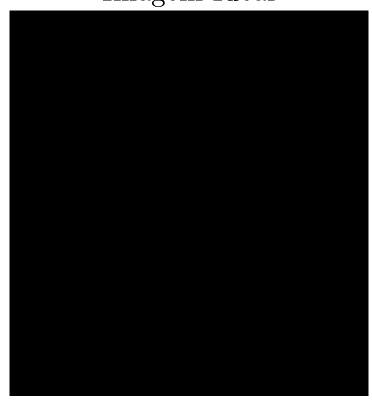
  

Imagem Original | Imagem Obtida | Imagem Ideal

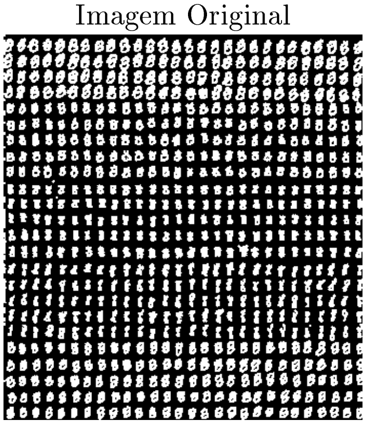
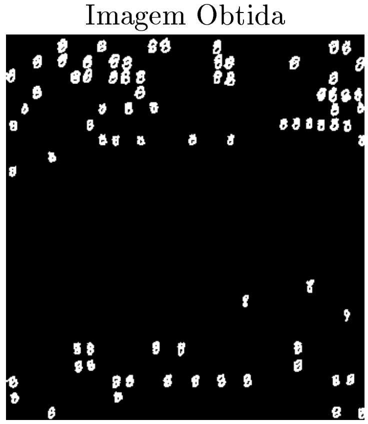
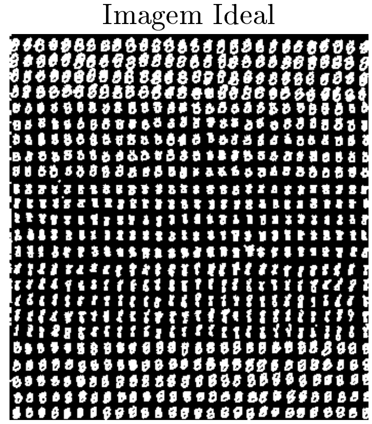
  

Imagem Original | Imagem Obtida | Imagem Ideal

Apesar do erro total ser relativamente pequeno para uma primeira abordagem (9,29%), o erro para menos está muito elevado. O ideal é tentar reduzir esse erro com técnicas de pré-filtragem e técnicas mais sofisticadas de aprendizado.

4 Conclusão

Neste breve período de estudos, executamos todos os objetivos previstos para o período. Estudamos o básico da morfologia matemática e Aprendizado Computacional. Implementamos a primeira parte do reconhecedor de dígitos e aprendemos a usar o ambiente de desenvolvimento. Agora estamos preparados para nos aprofundar mais no assunto.

5 Futuras Considerações

Após essa primeira etapa, vamos continuar o nosso reconhecedor de dígitos. Para tal vamos estudar métodos para otimizar a busca de operadores. Vamos melhorar os filtros e realizar experimentos mais intensivos.

É necessário também generalizar os resultados obtidos para os outros dígitos.