Arcabouço para Reconhecimento de Escrita

Sistema Titanium de Reconhecimento de Escrita Online - SisTREO

ORIENTADORA
Prof. Dra. Nina S. T. Hirata
DCC – IME/USP

ALUNOS

Ricky Ye Lun Chow	4894603
Pedro Henrique Simões de Oliveira	4894502
Eduardo Gusmão Cáceres Pires	4895271

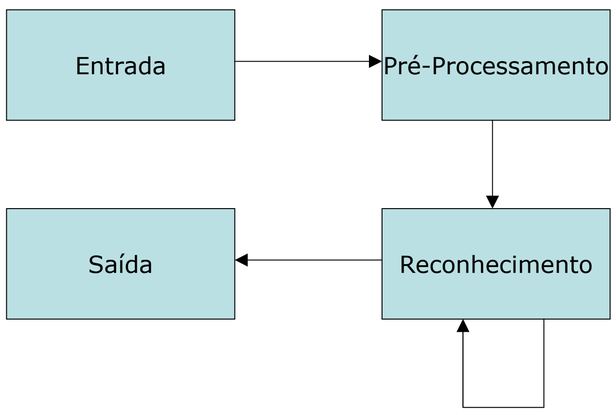
Motivação

O reconhecimento da escrita humana pelo computador é um problema extremamente complexo, envolvendo inúmeras variáveis que dependem não só da capacidade do algoritmo de **prever** o que o usuário realmente quer dizer, mas também do próprio usuário, que pode, por diversos motivos, escrever de diversas maneiras.

Existem muitas formas de se resolver, em partes, este problema. A nossa proposta é desenvolver um arcabouço capaz de padronizar o tratamento dos sinais e o comportamento dos algoritmos com relação a uma **estrutura de dados**, para que a interação entre eles seja possível, possibilitando uma combinação de técnicas que possa melhorar a capacidade de reconhecimento.

Antes, porém, é necessário conhecer como o reconhecimento é feito. Para isso, recorremos a diversos artigos focados em diferentes problemas de reconhecimento:

- Texto (letra de forma e à mão[1])
- Ideogramas orientais [2]
- Fórmulas matemáticas [3] [5]
- Diagramas de blocos [4]



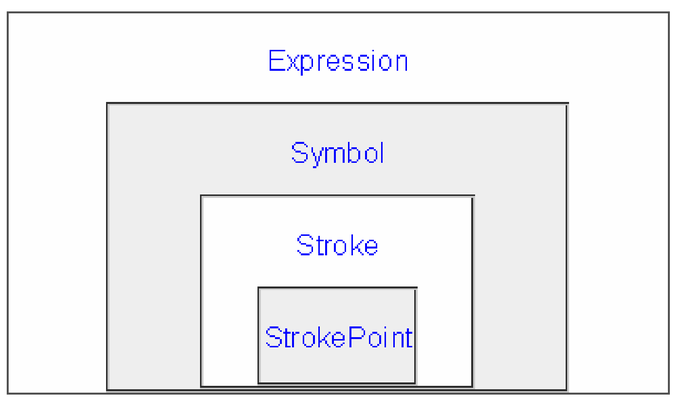
Fluxo de Dados do Arcabouço

Metodologia

Utilizamos a seguinte metodologia para o desenvolvimento do arcabouço:

1-) Análise de Requisitos: identificamos quais são os subproblemas envolvidos, quais as necessidades quanto à manipulação e armazenamento dos dados.
2-) Desenvolvimento da Arquitetura: desenvolvemos um modelo conceitual da estrutura de dados e como é feito o armazenamento interno das informações.
3-) Implementação de Algoritmos de Exemplo: resolvemos alguns dos subproblemas envolvidos a fim de testar a utilização do arcabouço e sua arquitetura (SisTREO).
4-) Testes: teste do SisTREO.

No SisTREO, focamo-nos no problema de reconhecimento de **fórmulas matemáticas**, pois inserir uma fórmula matemática num computador é muito mais complexo do que digitar um texto. A partir de uma fórmula matemática, a idéia é gerar um código em **LaTeX** que corresponda à fórmula escrita pelo usuário.



Estrutura Conceitual do Arcabouço

Definições

Traço (Stroke) – Seqüência de pontos (**StrokePoint**) que começa a partir do momento em que o dispositivo de entrada (mouse ou tablet) é ativado (segurar o botão do mouse, por exemplo) até ele ser desativado (soltar o botão).
Caractere – Um elemento do alfabeto reconhecível.
Símbolo (Symbol) – Conjunto de traços que pode representar um caractere.
Expressão (Expression) – Conjunto de todos os Símbolos escritos.
Bounding Box – Caixa que envolve todos os pontos de um conjunto (Traço, Símbolo, Expressão) de forma *justa*, "sem borda".
Escrita Online – Tipo de escrita onde há a indicação de tempo nos pontos (*x*, *y*, *t*).
Escrita Offline – Tipo de escrita onde não há indicação temporal (uma página escaneada, por exemplo).

Subproblemas Envolvidos

Pré-Processamento

Interpolação de Pontos – Como um traço é um conjunto de pontos finitos, se ele for traçado muito rapidamente dois pontos consecutivos podem ficar distantes um do outro. Isso compromete não só a visualização do traço na tela, mas também a verificação de intersecção entre dois traços. Para isso a interpolação de dados pode ser útil para estimar pontos implícitos.

Agrupar Traços em Símbolos – Quando escrevemos, a única informação delimitadora que temos é a indicação de início e fim dos traços. Como existe a intenção de reconhecermos símbolos, devemos agrupar estes traços em conjuntos que possamos reconhecer. Para isso, a forma mais simples e lógica é verificar a intersecção entre traços, o que nem sempre resolve este problema, como é o caso do símbolo '=', onde os traços não se interceptam, mas formam um único símbolo.

Normalização do Tamanho dos Símbolos – Dependendo do algoritmo, a comparação entre um símbolo e um caractere pode ser inviável se um símbolo estiver maior ou menor que o caractere, em *pixels*. Diversas técnicas de normalização envolvem "sampling" ou o uso de um *baseline* [1].

Outros Subproblemas – Menos utilizados, mas que podem ser importantes dependendo da abordagem do problema, existem os subproblemas de normalização da rotação (para letras inclinadas), detecção de traços "atrasados" (como o corte no t, ou o pingo no i), suavização (para remover movimentos bruscos durante a escrita), entre outros.

Reconhecimento

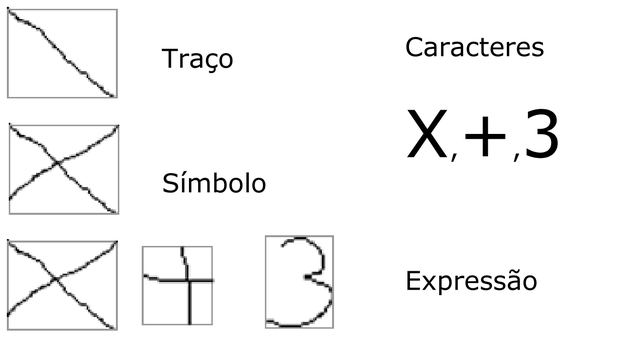
Reconhecimento de Símbolos – Existem inúmeras técnicas para resolver este problema. Algumas delas abordam isoladamente cada símbolo, outros levam em consideração os símbolos ao redor, e outros ainda requerem um treinamento inicial com a escrita do usuário para tomar como base.

Como exemplo, optamos pela independência dos símbolos, e pela utilização de redes neurais para efetuar o reconhecimento dos símbolos. A entrada das redes neurais foi obtida da seguinte forma: pegamos a caixa envoltória de cada símbolo, quebramos em um "*grid*" de tamanho 8x6 e em cada célula que contém um ponto desenhado definimos a entrada associada como 1. Caso contrário, -1.

A técnica utilizada para o treinamento da rede foi retro-propagação, mais precisamente o método do gradiente com um termo de "momentum" somado. Ainda poderiam ter sido utilizadas outras técnicas, como algoritmos genéticos ou outros algoritmos, como o método dos gradientes conjugados.

Reconhecimento de Expressões – Com os símbolos propriamente reconhecidos, queremos verificar relações entre eles. Com esses relacionamentos entre símbolos podemos definir, por exemplo, que o símbolo reconhecido como 2 é o expoente do símbolo x em x^2. Só então poderemos gerar algo como `x^2` e não somente x2. Para este problema mesclamos algumas técnicas, tais como:
- Dado o grafo completo formado pela ligação dos centros de todos os símbolos (centro do bounding box), encontrar a sua **árvore geradora mínima** (símbolos próximos têm uma maior chance de estarem relacionados) [5].
- Dado o conjunto de pontos que formam os símbolos da expressão, calcular a reta de **regressão linear** (reta que melhor aproxima o conjunto de pontos).

Com estas informações geramos um grafo onde cada símbolo é um nó e a eles estão associadas referências aos símbolos: superior, superior direito, frontal, inferior direita e inferior. Para a identificação de símbolos compostos (Ex.: "=") e símbolos dependentes (Ex.: somatório, integral) fazemos buscas no grafo.



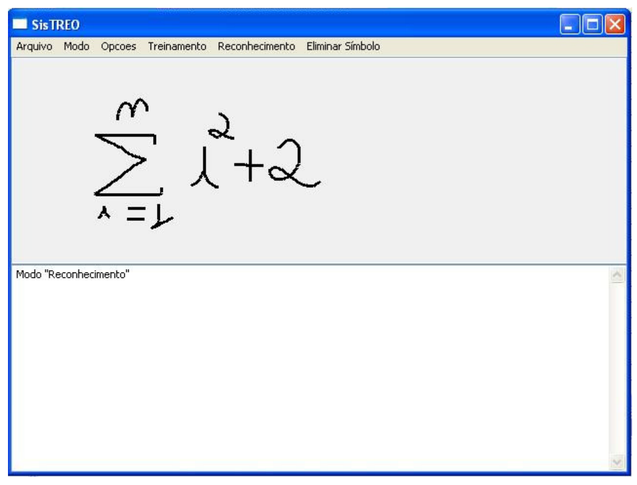
Traços e Símbolos com os seus Bounding Boxes, exemplos de Caracteres e a Expressão completa

Utilizando o SisTREO

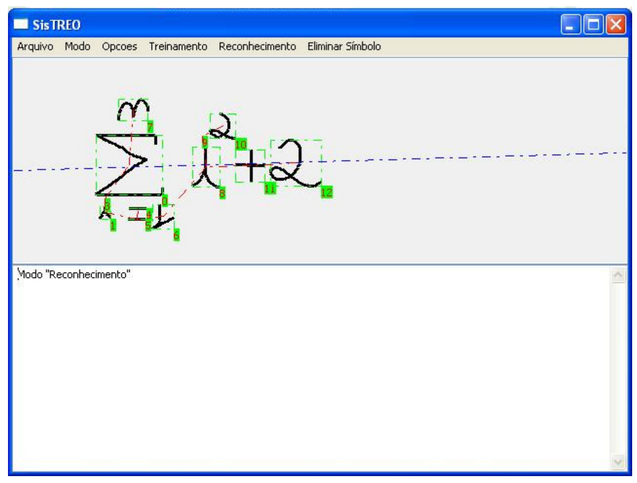
O Sistema possui 3 modos:
- **Reconhecimento**: Neste modo o usuário irá escrever a sua expressão e, ao selecionar Exibir Símbolos, o programa irá exibir os símbolos isolados por seus "bounding boxes", realizando também o reconhecimento dos símbolos e da expressão.

Ativando a opção Eliminar símbolo, pode-se "riscar" um símbolo para que aquele símbolo seja eliminado. Desativando o modo eliminar símbolo, o usuário poderá clicar novamente em "Exibir Símbolos" no menu e a expressão será reconhecida novamente, mas sem o símbolo eliminado.

Outra opção importante é carregar/gravar o perfil de cada usuário, isto é, a configuração da rede neural obtida através do treinamento, ajustando o reconhecimento a cada indivíduo.
- **Escrever LaTeX**: Ativa o modo para que o usuário escreva o texto em LaTeX e insira a expressão reconhecida.
- **Treinamento**: Modo de configuração das Redes Neurais. Nele, o usuário seleciona a opção definir símbolos, onde o programa pedirá exemplos que servirão de modelo para treinar a rede. Este conjunto de exemplos pode ser gravado em arquivo para ser carregado mais tarde, inclusive para continuar o treinamento.



Tela do SisTREO em Modo Reconhecimento



Tela do SisTREO com a divisão em Símbolos, a árvore geradora mínima e a reta de regressão linear

Conclusão

O arcabouço desenvolvido é muito abrangente para tratar os mais diversos tipos de problemas relacionados ao reconhecimento de escrita online. Dessa maneira podemos integrar, por exemplo, uma solução para reconhecer textos com outra que reconheça ideogramas e criar uma nova solução que reconheça dinamicamente qualquer tipo de texto, sem se preocupar em dizer de antemão que tipo de problema está sendo tratado.

Como teste, os algoritmos implementados são modestos e não tratam casos especiais, mas a abrangência do arcabouço permite melhorias até mesmo sobre uma implementação simples como esta. Pretendemos disponibilizar o arcabouço livremente para que outros projetos futuros usem o arcabouço como base para novas pesquisas.

Referências Bibliográficas

[1] S. Jaeger, S. Manke, J. Reichert, A. Waibel. *Online handwriting recognition: the NPen++ recognizer*. IJDAR, 2001.
[2] K. Ishigaki, H. Tanaka, N. Iwayama. *Interactive Character Recognition Technology for Pen-based Computers*. 1999.
[3] K. Chan, D. Yeung. *Mathematical expression recognition: a survey*. IJDAR, 2000.
[4] Kara, Levent. *Automatic Parsing and Recognition of Hand-drawn Sketches for Pen-Based Computer Interfaces*. 2004
[5] Matsakis, Nicholas. *Recognition of Handwritten Mathematical Expressions*. 1999.