

MORFOL: UMA FERRAMENTA PARA ANÁLISE LÓGICA DE CENAS

Aluno: Marco Antônio Barbosa Teixeira

Orientador(es): Edward Hermann Haeusler e Geiza Maria Hamazaki da Silva

Introdução

O processo de Reconhecimento de Cenas em Processamento Digital de Imagens (PDI) pode ser feito a partir de algoritmos de reconhecimento categórico/estatístico e uso de funções de verossimilhança. Pode-se também fazer uso de princípios de morfologia matemática e através de uma gramática ou descrição linguística da mesma para reconhecer padrões estruturais na imagem que estejam formalmente relacionados com a descrição do protótipo ou modelo. O Juiz Virtual é um programa de reconhecimento de padrões aplicado em jogos de futebol, o qual faz a calibração automática da imagem do campo, determinando assim a posição das linhas do campo e o posicionamento da câmera em relação ao campo. Este programa utiliza-se principalmente de processos de reconhecimento de padrão estatístico. Não se tem conhecimento na literatura da aplicação do reconhecimento de padrões sintático ou baseado em morfologia aplicado ao domínio específico do Juiz Virtual (JV). Parece ser interessante o uso de reconhecimento estrutural em função de uma maior estruturação e conseqüente possibilidade de otimização no reconhecimento.

Objetivos

O objetivo deste projeto é a construção de uma ferramenta (MORFOL) que possibilita a análise morfológica de uma linguagem, que descreve um modelo, com base em uma descrição linguística em linguagem lógica gerando código na linguagem C. Este código tenta otimizar a tarefa de reconhecimento de segmentos encontrados em uma imagem (Figura 1 e Figura 2), como por exemplo, encontrar dois segmentos de modo que estes sejam paralelos. Utilizando MORFOL espera-se dar subsídios para uma análise mais conclusiva do uso de princípios morfológicos em detrimento de reconhecimento estatístico de cenas no domínio do JV.



Figura 1- Imagem a ser analisada



Figura 2- Imagem com linhas interpretadas

Metodologia

Utilizando a ferramenta MORFOL, dado um modelo descrito na Linguagem de Descrição (LD) é gerado código em Linguagem C (LC) para o mesmo modelo, o qual utiliza primitivas de segmentação do Juiz Virtual. A linguagem de descrição tem como base uma descrição em linguagem lógica semelhante ao Prolog[3]. Inicialmente é definida a base (Base) composta das primitivas de segmentação, seguida das definições dos predicados utilizados nas modelagem (Def) e finalmente o código que descreve o modelo (Cod).

Base: <Sequência de Primitivas >	<Primitiva> ::= <identificador> “=” <identificador>
Def: < Sequência de Predicados >	<Predicado> ::= <Funcao> <Funcao> <operador> <Predicado>
Cod: < Funcao Principal >	<Operador> ::= “ , ” “ ; ” “ ! ”
	<Funcao> ::= <identificador> (<lista de parametros>)
	Descrição dos operadores: ' , ' - operador de conjunção , ' ; ' - operador de disjunção e ' ! ' - operador de corte

Figura 3- Esquema do arquivo que descreve o modelo e a gramática da Linguagem de descrição

Para a geração do código em LC, foram analisadas as opções de implementação via transformação de código (XML/XSLT[4] e TXL[1]). Foi utilizado o TXL, onde dada a gramática de LD e de LC foi realizado um mapeamento, com o apoio de tabelas para o armazenamento de informações relevantes (associação de primitivas e funções) e locais (variáveis de funções, recursão e corte), para a transformação do código entre as linguagens.

<p><i>Exemplo de Especificação em MORFOL</i></p> <p>Base: Par=paralelo Ort=ortogonal Ent=entre</p> <p>Def: Rec(x1,x2,x3,x4)=Par(x1,x3),Par(x2,x4),Ort(x1,x2) Campo(x1,x2,x3,x4,x5,x6,x7,x8,x9)=Area(x1,x2,x3,x4,x5,x6,x7),Fundo(x8,x1,x9),(Ent(x8,x5,x9);Ent(x8,x7,x9)) Fundo(x1,x2,x3)=Par(x1,x3),Ort(x1,x2) Area(x1,x2,x3,x4,x5,x6,x7)=Rec(x1,x2,x3,x4),Rec(x1,x5,x6,x7),Ent(x5,x2,x7),Ent(x2,x4,x7),Ent(x1,x3,x6)</p> <p>Cod: Area(x1,x2,x3,x4,x5,x6,x7)</p>
<p><i>Saída obtida pela transformação</i></p> <pre>int Area (x1, x2, x3, x4, x5, x6, x7) { for (x1 = 0; x1 < MAXLINHAS; x1 ++) { for (x2 = 0; x2 < MAXLINHAS; x2 ++) { for (x3 = 0; x3 < MAXLINHAS; x3 ++) { for (x4 = 0; x4 < MAXLINHAS; x4 ++) { for (x5 = 0; x5 < MAXLINHAS; x5 ++) { for (x6 = 0; x6 < MAXLINHAS; x6 ++) { for (x7 = 0; x7 < MAXLINHAS; x7 ++) { Rec (x1, x2, x3, x4) && Rec (x1, x5, x6, x7) && Ent (x5, x2, x7) && Ent (x2, x4, x7) && Ent (x1, x3, x6) } } } } } } } } int Rec (x1, x2, x3, x4) { return (Par (x1, x3) && Par (x2, x4)&& Ort (x1, x2));} int Campo (x1, x2, x3, x4, x5, x6, x7, x8, x9) { return (Area (x1, x2, x3, x4, x5, x6, x7) && Fundo (x8, x1, x9) && (Ent (x8, x5, x9) Ent (x8, x7, x9)));} int Fundo (x1, x2, x3) { return (Par (x1, x3) && Ort (x1, x2));}</pre>

Figura 4- Exemplo dos padrões de entrada e saída

Conclusões

A utilização de uma linguagem de descrição para reconhecer padrões estruturais da imagem que estejam formalmente relacionados apresentou resultados interessantes como a facilidade na descrição do protótipo ou modelo (não se limitando somente à área de PDI), a rápida implementação do mesmo além da geração de um código legível. Como trabalhos futuros podemos citar a comparação de eficiência entre o método de reconhecimento apresentado e os baseados em padrões estatísticos, a verificação da possibilidade de otimização do código gerado, além da aplicação deste processo para sistemas de tempo real.

Referências

- [1] FELIX, M. F. **LET: Uma Linguagem para Especificar Traduções e seu Compilador**. Dissertação de Mestrado - Departamento de Informática - PUC-Rio. 1998.
- [2] Szenberg, F. **Acompanhamento de Cenas com Calibração Automática de Câmeras**. Tese de Doutorado - Departamento de Informática - PUC-Rio. 2001.
- [3] Palazzo, L. A. M. **Introdução à Programação Prolog**. EDUCAT, Pelotas, 1997.
- [4] <http://www.w3.org/Style/XSL/>