Sistema de informação de aprovação de crédito utilizado a lógica Fuzzy

Diego Canabarro Parode

diego.parode@outlook.com

CPM-FURB, Inteligência Artificial

Giovanni

nomealuno2@mail.com

CPM-FURB, Inteligência Artificial

*Resumo –* Um dos problemas clássicos em visão computacional é encontrar padrões em objetos em imagens digitais. Esta tarefa se torna ainda mais difícil quando existe a necessidade em obter invariância a rotação, translação, escala, brilho e contraste (RSTBC - *rotation, scale, translation, brightness and contrast*) nos objetos. Um algoritmo de busca de máscara foi recentemente proposto para atender aos aspecto RSTBC, chamado de Ciratefi. Porém, a execução deste algoritmo em um computador convencional requer diversos segundos. Além disso, o fato de ter muitos parâmetros dificulta a sua implementação em hardware. Este trabalho propõe o projeto de um software que gera automaticamente módulos compiláveis em *Hardware Description Logic* (VHDL) que implementam o filtro circular do algoritmo Ciratefi em dispositivos *Field Programmable Gate Array* (FPGA). Esta solução acelera o tempo de processamento do algoritmo, que em um PC de 3GHz chega a 7s, enquanto que em um dispositivo Stratix III da Altera executa em 1,367ms.

*Palavras-Chaves –* Invariante ao RSTBC, Localização de máscaras, Tempo real, VHDL, FPGA, Visão computacional.

*Abstract –* Abstract - One of the classic problems in computer vision is to find patterns in digital image objects. This task becomes more difficult when there is a need for obtaining invariance to rotation, scale, translation, brightness and constrast (RSTBC) in the objects. A search algorithm, which was named Cifatefi, was recently proposed to fulfil those requirements. However, the running of its implementation in a conventional computer takes several seconds. Moreover, the fact that the algorithm demands that many parameters makes it difficult to be implemented in hardware. This paper proposes the design of a software that automatically generates compilable modules in Hardware Description Logic (VHDL), which implement the Ciratefi's circular filter algorithm in Field Programmable Gate Array (FPGA) devices. This solution speeds up the processing time of the algorithm, that on a 3GHz PC reaches up to 7s, while in an Altera Stratix III device runs in 1.367 ms.

*Keywords –* Invariant to RSTBC, Location masks, Real Time, VHDL, FPGA, Computer vision.

# INTRODUÇÃO

Processamento de imagem e visão computacional tornaram-se populares em muitas áreas com aplicações em medicina, segurança, indústria e robótica. Muitos algoritmos poderosos continuam sendo desenvolvidos. Porém, são algoritmos que por sua vez, exigem cada vez mais processamento, com o uso intenso de cálculos matemáticos, tais como médias, interpolações ou correlações e outros ainda mais complexos. Sistemas que utilizam tecnologia de processamento paralelo podem apresentar um desempenho semelhante ou mesmo superior aos sistemas computacionais convencionais para aplicações específicas, sendo uma opção de menor custo e também mais compactas.

Neste sentido, os dispositivos Field Programmable Gate Array (FPGA) representam uma tecnologia de processamento adequada e promissora. Para muitos algoritmos de cálculos matemáticos repetitivos, devido principalmente à capacidade que os FPGAs têm de realizar muitas tarefas em um único ciclo.

Um novo algoritmo de busca por imagem de máscara (template matching) invariante à rotação, escala, translação, brilho e contraste (RSTBC em inglês), chamado de Ciratefi é um bom exemplo de algoritmo que pode ser implementado em FPGA. Seu desempenho em computadores convencionais é competitivo em relação aos algoritmos para aplicações semelhantes, mas é da ordem de alguns segundos para imagens com resolução de 640x480.

Neste trabalho, foi desenvolvido um software que, dados os parâmetros de entrada, gera automaticamente os módulos compiláveis na linguagem de descrição de hardware VHDL, e que implementam o primeiro dos três filtros do Ciratefi em FPGA (filtro Cifi). Os módulos gerados em VHDL estão otimizados e todos os seus processamentos estão sincronizados e registrados em *pipeline*.

Na seção 2 são discutidas os filtros do algoritmo de Ciratefi. Na seção 3 é descrita a metodologia utilizada no desenvolvimento do sistema apresentado. Na seção 4 são expostos os resultados experimentais obtidos pelo uso do sistema. Finalmente, na seção 5, são apresentadas as conclusões e sugestões para trabalhos futuros.

# algoritmo de ciratefi

O algoritmo de Ciratefi tem como objetivo encontrar uma escala de cinza da imagem de consulta Q em uma imagem maior para analise A, invariante a rotação, escala, translação, contraste e brilho. Este algoritmo possui três tipos de filtros em cascata. Cada filtro sucessivamente exclui pixels que não têm nenhuma chance de combinar com o modelo.

O primeiro filtro, chamado CIFI (*Circular Sampling Filter*), calcula a média de tons de cinza das imagens *A* e *Q* em círculos, conforme a figura 1, e as utiliza para classificar pixels e para determina um fator de escala provável. Para conseguir isso, CIFI faz correlações entre as sucessivas *2-D* da matriz dos valores médios em círculos de *Q* em vários fatores de escala (figura 2) e  *3-D* da matriz que contém, para cada pixel (x, y) em A , um vetor dos valores médios dos círculos centrados em (x, y), de acordo com a seguinte função:



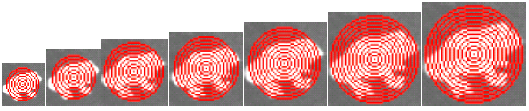


Figura 1: Projeções circulares em diferentes escalas [1]

Fonte: Nobre e Kim (2008, p. 30)

O segundo filtro, chamado Rafi (*Radial Sampling Filter*), calcula para cada pixel candidato de primeiro grau (x, y), as projeções de imagens de *A* e *Q*, em linhas radiais conforme a figura 2, com o raio dado pelo fator de escala calculado pela CIFI. Rafi altera os primeiros pixels candidatos que têm chance de adequação ao modelo de segundo grau.



Figura 2: Projeções em linhas radials

Fonte: Nobre e Kim (2008, p. 31)

O terceiro filtro, chamado TEFI (*Template Matching Filter*), é um algoritmo convencional de casamento de máscara, invariante ao brilho e contraste que é aplicado aos pixels candidatos do segundo grau, usando as escalas e os ângulos determinados respectivamente por Cifi e Rafi.

Porém, Segundo Nobre e Kim (2008, p. 33) o algoritmo de Cirater leva alguns segundos para calcular as posições correspondentes. Para o pior caso testadas (*A* com 465 × 338 pixels, *Q* com 52 × 51 pixels, 6 e 36 escalas de ângulos), o algoritmo Ciratefi completa tomou 22s usando um Pentium 4 de 3GHz, divididos da seguinte forma:

• O primeiro filtro - CIFI, levou 2.5s para calcular a matriz 3D 4.5s para o processo de correlação, com um total de 7s.

• O segundo filtro - Rafi, levou 13s para a saída ao seu resultado.

• O filtro final - Rafi, foi o mais rápido e levou cerca de 1s a saída de seu resultado.

# METODOLOGIA UTILIZADA

Para atender aos requisitos de flexibilidade e alto desempenho optou-se por criar programas na linguagem C, para gerar automaticamente módulos em VHDL para os parâmetros selecionados. Para a utilizada de cálculos, como raiz quadrada e divisão, foram utilizados códigos de terceiros.

Neste trabalho, optou-se por trabalhar com dispositivos Altera e suas ferramentas de síntese, roteamento e análise. Para a simulação, utilizou-se de ModelSim MentorGraphics que permite explorar o paralelismo de comportamento e analisar os resultados do hardware VHDL projetado.

O sistemas utiliza um módulo de hardware *Configurable Window Processor* (CWP), com o objetivo de calcular a soma e a média dos pixels em cada círculo (filtro CIFI) para cada pixel da imagem de *A*. Para evitar acessos a memórias externas, optou-se por um FPGA capaz de armazenar um total de 640 × 480 tons de cinza da imagem *A.* O dispositivo EP3SL340H1152C3, que possui uma memória interna de 16 Mbits.

A Figura 3 representa a arquitetura de hardware de leitura de pixels da imagem *A* da CWP, para que três janelas de processamento possam executar em paralelo. As coordenadas dos pixels para a média são calculadas automaticamente pelo gerador de VHDL.

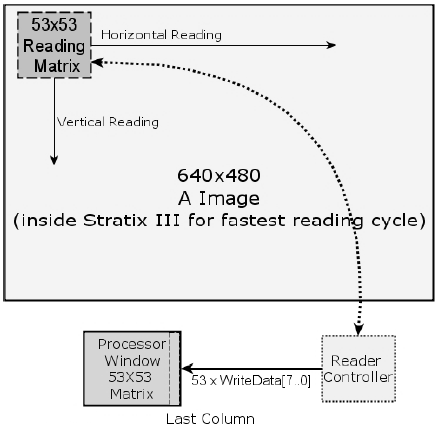


Figura 2: Arquitetura para leitura de dados

Fonte: Nobre e Kim (2008, p. 86)

Na abordagem proposta, a primeira coluna do módulo CWP (filtro CIFI) é passado para o próximo filtro (filtro Rafi) após a primeira correlação. Assim, não é necessário que o filtro Rafi espere o fim do filtro CIFI para iniciar o processamento dos dados. Ambos executam simultaneamente sobre um pixel a cada iteração, com uma latência inicial co, a resolução do maior modelo de escala. Ou seja, a cada iteração, um pixel é classificado como um "candidato" ou "não candidato".

Seguindo o caminho dos dados, após o cálculo da soma dos tons de cinza em círculos CWP, o módulo médio divide cada soma pelo número de pixels. Então, usando o valor calculado pela matriz , as correlações são calculadas e a maior correlação é escolhida. Finalmente, deve-se comparar o resultado da maior correlação com um determinado limite de correspondência entre o modelo apresentado.

# Resultados experimentais

Neste trabalho foi implementado e analisado somente o filtro CIFI, a primeira dos filtros Ciratefi. Comparamos os resultados do software implemento de CIFI (que utiliza variáveis de ponto flutuante), com a aplicação FPGA de ponto fixo a partir de simulações ModelSim, e verificou que os dois são bastante semelhantes.

Na Tabela 1 são apresentados o uso de recursos e desempenho para o pior caso (o maior modelo com resolução 53 × 53, quantidade máxima de 14 círculos, e número máximo de escalas 7, obtidos com a ferramenta Timing Analyzer sobre o software Altera Quartus II.

Após a latência do sistema, a arquitetura proposta irá classificar um pixel em cada relógio como "candidato para correspondência" ou "não candidato", e irá mostrar a escala e o ângulo. O tempo de processamento para CIFI (que procura uma imagem de 640 × 480 para um modelo de 53 × 53) é 1.06ms para o FPGA rodando a 258 MHz, contra 7s em um Pentium de 3GHz.

Tabela 1 – Tamanho e recursos lógicos por módulo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **RESULTADOS DOS TESTE** | | |  |
| **Módulo** | **Freq. Max. (Mhz)** | **Tamanho em lógica** | **Latência** | |
| CWT | 380 | 31.258 (12%) | 9 | |
| Correlações | 258 | 21.538 (1%) | 75 | |

# conclusões e perspectivas FUTURAS

Apresentamos neste projetado um sistema em FPGA que implementa parte de um algoritmo de localização de imagens de máscara, chamado de Ciratefi. Este algoritmo realiza a busca por padrão de imagens de forma invariante em relação à escala, rotação, translação, brilho e contraste. O sistema final pode ser composto por um sistema eletrônico híbrido com processamento paralelo em hardware, para os dois primeiros filtros e processamento convencional para o filtro final. Os demais filtros são objetivos de trabalhos futuros.

Além disso, podem-se realizar melhorias a fim de se alcançar maiores freqüências de operação, com o uso de outras técnicas e ferramentas de análise de roteamento. Além disso, as estratégias utilizadas neste trabalho podem ser utilizadas em outras técnicas de processamento de imagens e algoritmos de visão computacional.

referências bibliográficas

H. P. A. NOBRE, H. Y. KIM. “Geração automática de módulos VHDL para localização de padrões invariante a escala e rotação em FPGA”, *Tese de Mestrado*, USP, São Paulo, 2008.