



Hardware Acelerador da Técnica de Reconhecimento de Caracteres

PROPOSTA DE TRABALHO DE GRADUAÇÃO

Aluno: Luiz Antonio de Oliveira Júnior (laoj2@cin.ufpe.br)
Orientador: Edna Natividade da Silva Barros (ensb@cin.ufpe.br)
Área: Sistemas Embarcados

Abril de 2017

Resumo

A maioria dos algoritmos de reconhecimento de caracteres (OCR) de alta precisão propostos na literatura tem alto custo computacional, o que pode inviabilizar sua utilização em CPUs embarcadas, se o tempo de resposta for um requisito crítico. A complexidade de desenvolvimento desses algoritmos também inviabiliza implementá-los como um Processador de Propósito Único, dado o alto custo de desenvolvimento do projeto. Assim, propomos uma arquitetura heterogênea de hardware e software para acelerar uma técnica de OCR baseada em HOG e NN. O sistema será prototipado em uma plataforma embarcada, composta por uma CPU conectada a um componente de hardware reconfigurável (FPGA) via PCI Express, e será comparado com um modelo de referência em software para avaliação da precisão e tempo de processamento.

Introdução

Optical Character Recognition (OCR) é uma tarefa desafiadora nas áreas de Processamento de Imagens e Aprendizagem de Máquina que objetiva desenvolver sistemas computacionais capazes de reconhecer automaticamente caracteres em imagens digitais.

Esses sistemas têm sido amplamente utilizados em aplicações industriais e comerciais. Um exemplo é a digitalização automática de passaportes, extratos bancários e outros documentos impressos por um computador. Nesse contexto, os caracteres geralmente são de um mesmo tipo de fonte e estão dispostos em um background aproximadamente uniforme.

Por outro lado, aplicações como assistentes de navegação móvel e scene understanding [1], reconhecimento automático de placas automotivas e busca de imagens pelo seu conteúdo textual envolvem o reconhecimento de caracteres em cenas naturais (natural scenes ou natural images). Nesse contexto, o reconhecimento de caracteres pode ser mais desafiador e computacionalmente custoso, uma vez que o background geralmente é não-uniforme, os caracteres são de fontes e tamanhos diferentes, existem diferentes condições de iluminação, diferentes disposições do caractere na imagem e diferentes texturas [2].

Algumas aplicações de OCR aplicadas a imagens de cenas naturais podem ser ainda mais críticas. Por exemplo, dispositivos que utilizam OCR para auxiliar deficientes visuais a se locomoverem nas ruas exigem requisitos como baixo tempo de resposta do sistema, baixo custo, baixa consumo de energia e suporte a mobilidade, ao mesmo tempo que requerem boa taxa de reconhecimento.

Adicionalmente, a maioria dos algoritmos de OCR de alta acurácia que têm sido propostos na literatura são computacionalmente caros, o que pode inviabilizar sua implementação em processadores embarcados de baixo custo e consumo de energia, se o tempo de resposta for um requisito crítico do sistema. A complexidade de desenvolvimento desses algoritmos também inviabiliza a implementação de todas suas funcionalidades como um Processador de Propósito Único (PPU), dado o alto custo de desenvolvimento NRE (non-recurring-engineering) do projeto.

Uma solução para esses problemas é implementar o algoritmo em arquiteturas heterogêneas, compostas por processadores embarcados de propósito geral (CPUs embarcadas) e aceleradores em hardware (PPUs, como FPGAs ou DSPs). Nesse tipo de solução, as tarefas mais computacionalmente custosas do algoritmo são implementadas diretamente em hardware, o que faz a solução atingir alto desempenho, enquanto mantém baixo custo e baixo consumo de energia. Como resultado de trabalhos anteriores, uma técnica de reconhecimento de caracteres (OCR) em cenas naturais, baseada em Histograma de Gradientes Orientados (HOG) e Redes Neurais Artificiais (NN), foi desenvolvida e validada pelos autores deste projeto no Centro de Informática da UFPE. Apesar de a técnica ter taxa de acerto melhor que muitos trabalhos propostos na literatura nos benchmarks Chars74K-15 [3] e ICDAR 2003 [4], sua execução em uma CPU embarcada é inviabilizada devido ao seu alto custo computacional.

Motivados por esse contexto, propomos uma arquitetura heterogênea de hardware e

software (C-FPGA) para acelerar a técnica de OCR em imagens de cenas naturais desenvolvida e que é baseada em HOG e NN.. O sistema será prototipado na placa Terasic DE2i-150, uma plataforma embarcada composta por um processador Intel Atom N2600 conectado a um FPGA da família Cyclone IV da Altera via PCI Express. A implementação será comparada com um modelo de referência otimizado escrito na linguagem de programação C para avaliação da acurácia e tempo de processamento.

Objetivos

O objetivo deste trabalho é propor, implementar e avaliar o desempenho de uma arquitetura heterogênea de hardware e software (C-FPGA) para acelerar o tempo de execução da técnica de OCR em cenas naturais, baseada em Histograma de Gradientes Orientados [5] e Redes Neurais Artificiais [6], desenvolvida pelos autores deste projeto em um trabalho anterior. A arquitetura será prototipada na placa Terasic DE2i-150, uma plataforma embarcada híbrida composta por um processador Intel ATOM N2600 conectado a um FPGA da família Cyclone IV da Altera via PCI Express.

A implementação do algoritmo de OCR deverá ser dividida entre hardware e software. Os módulos de hardware serão implementados na linguagem System Verilog HDL como IP Cores, devem acelerar alguma das tarefas mais computacionalmente custosas do algoritmo e serão prototipados em FPGA. Já os módulos de software, serão escritos em C ANSI e deverão ser executados na CPU.

O sistema deve ter tempo de resposta menor que o modelo de referência, manter uma boa taxa de reconhecimento e ser prototipado em uma placa de baixo custo, de modo que seja possível sua utilização em aplicações de OCR com requisitos de tempo, potência, custo e área muito críticos, como dispositivos para auxiliar deficientes visuais a se locomoverem nas ruas e sistemas móveis de reconhecimento de placas de carro.

O desempenho do sistema deve ser avaliado através do uso de benchmarkings e de técnicas de profiling, ser validado através de verificação funcional e através da prototipação em FPGA.

Metodologia

Como a implementação do algoritmo de OCR deverá ser dividida entre hardware e software, objetivando acelerar o tempo de execução do sistema, é necessário ter uma metodologia bem definida para atender aos requisitos temporais. Nesse trabalho, o processo de desenvolvimento foi dividido em duas etapas (1) Definição da arquitetura do sistema, que consiste na escolha de quais módulos serão implementados em hardware e quais serão implementados em software e (2) Desenvolvimento dos módulos de hardware e sua integração com os módulos implementados em software.

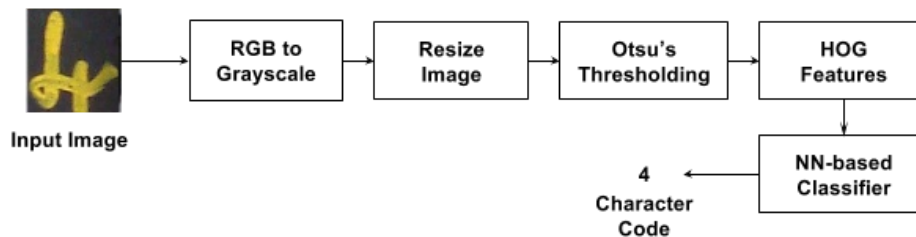


Figura 1: Flowchart do sistema de OCR baseado em HOG e NN. Cada caixa branca representa uma task de alto nível do sistema.

Inicialmente, o tempo de execução de cada módulo (Figura 1) do sistema de OCR baseado em HOG e NN será estimado, utilizando a ferramenta de profiling Gprof [7] e funções da biblioteca de C time.h. Depois, serão analisadas quais os dos módulos mais computacionalmente custosos são mais viáveis para serem implementados em hardware. Essa viabilidade é definida como uma relação entre o custo de implementação e custo de hardware versus a possível melhoria no desempenho geral do sistema, caso o módulo fosse implementado em hardware.

Portanto, a análise de cada módulo candidato a ser implementado em hardware deverá levar em consideração: (1) tempo de processamento estimado em FPGA; (2) tempo de envio dos dados da CPU para FPGA; (3) tempo de envio dos dados da FPGA para CPU; (4) os quão adequados são as estruturas de dados, tipos de dados e algoritmo utilizados no módulo para serem implementados em hardware; (5) limitações de recursos da placa de desenvolvimento DE2i-150.

Escolhido o módulo que será acelerado em hardware, o desenvolvimento do módulo em hardware seguirá a metodologia ip-PROCESS [8]. O fluxo geral de desenvolvimento de um IP-core de acordo com essa metodologia é ilustrado na figura 2.

Inicialmente, será feita uma análise bibliográfica sobre a implementação do algoritmo utilizado para o módulo escolhido. Depois que os requisitos de projeto forem capturados, o IP-core será desenvolvido no nível de abstração RTL, utilizando a linguagem System Verilog HDL, no ambiente de desenvolvimento Quartus II da Altera.

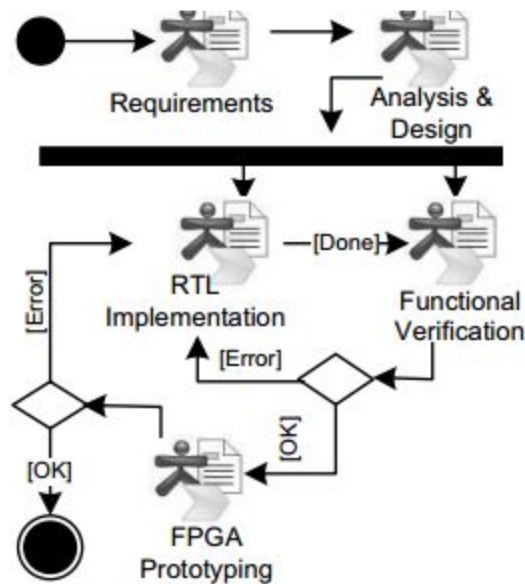


Figura 2. Fluxo geral de desenvolvimento com ip-PROCESS [8]

A verificação funcional será feita paralelamente através de testes de caixa-branca e testes de caixa-preta, no ambiente ModelSim, com o objetivo de validar de modo incremental cada passo do desenvolvimento do hardware (testes de caixa-branca) e sua interface com o sistema (teste de caixa-preta).

Para exemplificar essa metodologia, seja T a task escolhida para ser acelerada por hardware, $T = \{t^1, t^2, \dots, t_n\}$, no qual t_i é uma subtask de T , e seja M o módulo de hardware que implementa T , $M = \{m^1, m^2, \dots, m_n\}$, no qual m_i é o submódulo que implementa a subtask t_i . Deverá, então, existir um testbench b_i para validar cada submódulo m_i (Testes de caixa-branca). Quando todos os submódulos forem finalizados, validados e integrados, deverá também existir um testbench B para validar as interfaces do módulo M com o resto do sistema (Teste de caixa-preta).

O modelo de testbench que será utilizado nos testes está ilustrado na figura 3. As imagens de entrada são retiradas de dois benchmarks mais usados na literatura para avaliar o desempenho de técnicas de OCR, o Chars 74k e o ICDAR2003 e são enviadas em paralelo ao módulo (ou submódulo) em hardware e em software. Por fim, as saídas dos módulos em software (referência) e hardware são comparadas para avaliar a corretude do módulo desenvolvido. Se não houve erro em nenhuma saída, o módulo é considerado validado por simulação. Finalmente, o sistema proposto será prototipado na placa de desenvolvimento da Terasic DE2i-150 e sua corretude e tempo de execução analisados.

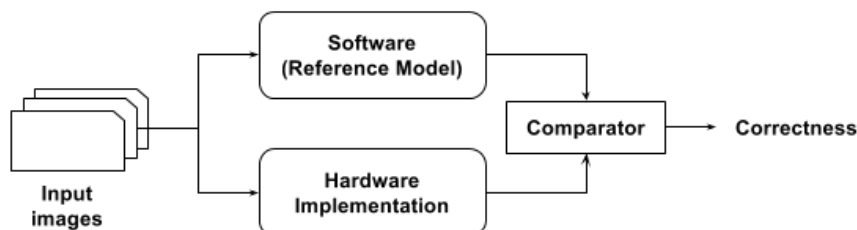


Figura 3: Modelo de testbench para validar os módulos de hardware implementados

Cronograma

| Atividade | Período | | | | | | | | | | | | | | | | | |
|-------------------------------------|---------|---|-------|---|---|------|---|---|-------|---|---|-------|---|---|---|---|--|--|
| | Março | | Abril | | | Maio | | | Junho | | | Julho | | | | | | |
| Definição da arquitetura do sistema | X | X | | | | | | | | | | | | | | | | |
| Revisão bibliográfica | X | X | X | X | | | | | | | | | | | | | | |
| Implementação dos Testbenches | | X | X | X | | | | | | | | | | | | | | |
| Desenvolvimento do hardware | | | | X | X | X | X | | | | | | | | | | | |
| Validação | | | | | | | X | X | X | | | | | | | | | |
| Integração com o Riffa | | | | | | | | X | X | X | | | | | | | | |
| Prototipação na DE2i-150 | | | | | | | | X | X | X | | | | | | | | |
| Integração dos módulos | | | | | | | | | | | X | X | | | | | | |
| Validação do sistema | | | | | | | | | | | | X | X | X | | | | |
| Escrita do TG | | | | | | | | | | | | X | X | X | X | X | | |
| Preparação da apresentação | | | | | | | | | | | | | X | X | X | | | |

Referências

- [1] C. Yi and Y. Tian, "Scene Text Recognition in Mobile Applications by Character Descriptor and Structure Configuration," in *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2972-2982, July 2014.
- [2] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J. Wu, and Andrew Y. Ng. 2011. Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning. In *Proceedings of the 2011 International Conference on Document Analysis and Recognition (ICDAR '11)*. IEEE Computer Society, Washington, DC, USA, 440-445.
- [3] T. E. de Campos, B. R. Babu and M. Varma. Character recognition in natural images. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, Lisbon, Portugal, February 2009.
- [4] Lucas, S. M., Panaretos, A., Sosa, L., Tang, A., Wong, S., & Young, R. (2003, August). ICDAR 2003 Robust Reading Competitions. In *ICDAR (Vol. 2003, p. 682)*.
- [5] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886-893)*. IEEE.
- [6] Hopfield, J. J. (1988). Artificial neural networks. *IEEE Circuits and Devices Magazine*, 4(5), 3-10.
- [7] Graham, S. L., Kessler, P. B., & McKusick, M. K. (2004). Gprof: A call graph execution profiler. *ACM SIGPLAN Notices*, 39(4), 49-57.
- [8] Lima, M., Aziz, A., Alves, D., Lira, P., Schwambach, V., & Barros, E. (2005, June). iprocess: Using a process to teach ip-core development. In *Microelectronic Systems Education, 2005.(MSE'05). Proceedings. 2005 IEEE International Conference on (pp. 27-28)*. IEEE.

Possíveis Avaliadores

Prof. Abel Guilhermino da Silva Filho

Assinaturas

Recife, ___ de _____ de _____

Luiz Antonio de Oliveira Júnior
(Aluno)

Edna Natividade da Silva Barros
(Orientador)