

Técnicas de Alto Desempenho aplicadas à Sistemas de Visão Computacional para navegação autônoma de VANTs

Ronaldo Y. Nagata, Álvaro L. Fazenda

¹Instituto de Ciência e Tecnologia - Universidade Federal de São Paulo (UNIFESP)
São José dos Campos, Brazil, 12231-280

ronaldoyutakanagata@gmail.com, alvaro.fazenda@unifesp.br

Abstract. *The use of UAV (Unmanned Aerial Vehicle) has been the object of study and research since it has numerous applications and advantages in comparison to larger manned equipment. However, there is still a concern regarding of it's autonomous navigation system. Autonomous navigation using GPS-only spatial orientation can cause errors due to noise or attacks on the transmit / receive system. An alternative is the use of the computer vision technique, which includes the acquisition of images in real time by the equipment allowing the control of flight over a predefined trajectory. The image processing used for computer vision presents significant demand for numerical computation, and will be optimized in this work, so that it can execute efficiently on a low consumption GPGPU board, which can be embedded in the UAV.*

Resumo. *O uso de VANT (Veículo Aéreo não Tripulado) tem sido objeto de estudo e pesquisa, já que possui inúmeras aplicações e vantagens em relação a equipamentos tripulados de maior porte. Entretanto, ainda existe uma preocupação em relação ao sistema de navegação autônomo do VANT. A navegação, apenas utilizando-se de orientação espacial por GPS, pode ocasionar erros, devido a ruídos ou ataques no sistema de transmissão/recepção. A navegação alternativa utilizada neste trabalho consiste no uso da técnica de visão computacional, que inclui a aquisição de imagens em tempo real pelo equipamento, permitindo o controle de vôo sobre uma trajetória pré-definida. Uma vez que o processamento de imagens utilizado para visão apresenta significativa demanda por computação numérica, este trabalho demonstra uma melhoria de desempenho do algoritmo utilizando-se de uma placa GPGPU de baixo consumo, e que possa ser embarcada no VANT.*

1. Introdução

De acordo com [Valavanis and Vachtsevanos 2014] um Veículo Aéreo Não Tripulado (VANT), também conhecido popularmente como drone, refere-se a uma aeronave que não possui tripulação ou passageiros a bordo, podendo ser controlado remotamente ou apresentar navegação autônoma. O uso de VANTs obteve uma alta de crescimento nos últimos anos devido ao seu baixo custo operacional e aplicabilidade em diversas áreas, tais como agricultura e sensoriamento remoto [Longhitano 2010], por exemplo.

Embora o uso de VANTs possua inúmeras vantagens, existe ainda a possibilidade que ocorra alguma falha em seu sistema de navegação autônoma, tornando-se uma preocupação na área de segurança, principalmente se o vôo ocorrer em áreas urbanas.

Muitos sistemas de navegação autônomas utilizam o sinal de GPS (*Global Positioning System* ou Sistema de Posicionamento Global) em conjunto com Sistemas de Navegação Inercial (*INS - Inertial Navigation System*) para estimar sua trajetória [Goltz 2011]. Contudo, o sinal de GPS pode sofrer interferências e ataques maliciosos levando a erros no cálculo de sua trajetória. Já no contexto militar a dependência de uma tecnologia estrangeira, em especial o sinal de *GPS*, também tem um fator considerável de relevância.

Uma das propostas para solucionar estes problemas reside em técnicas de visão computacional para que seja possível estimar o posicionamento do VANT através de imagens [Goltz 2011] previamente georreferenciadas. Para efetuar a captura de imagens durante o voo é necessária uma câmera de vídeo equipada no VANT. A câmera deve registrar imagens em tempo real que podem ser comparadas com imagens georreferenciadas previamente embarcadas no equipamento [Goltz 2011]. Com as duas fontes de imagens é possível, através do processo de correlação destas, estimar a posição corrente do VANT.

Este trabalho propõe a aplicação de técnicas de processamento de alto desempenho em busca de melhorar o desempenho do sistema de navegação proposto por [Braga et al. 2016], o qual utiliza a técnica de correlação de imagens. A alternativa proposta consiste em utilizar uma arquitetura de baixo consumo energético, que pode ser embarcada no VANT, a qual conta com uma GPGPU, conhecida por *NVIDIA JETSON TX2*, juntamente com um algoritmo de navegação autônoma implementado na sua versão em CUDA.

2. Navegação Autônoma em VANTs por correspondência de imagens

É possível realizar a navegação autônoma de VANTs através da correspondência de imagens como mostram os trabalhos de [Goltz and Shiguemori 2007], [SANTOS and Shiguemori 2009] e [Goltz 2011]. No trabalho de [Conte and Doherty 2008] faz-se a extração de bordas da imagem capturada em tempo real, e então, através do registro desta com uma imagem georreferenciada, obtém-se a posição geográfica aproximada do VANT. Tal técnica é conhecida por Correlação Digital de Imagens (*DIC - Digital Image Correlation*). A correspondência é executada após uma Rede Neural Artificial (RNA) extrair as bordas das imagens. A técnica apresentou bons resultados quando comparados a outros métodos da literatura [He and Siyal 1998] [Li et al. 2008]. Os trabalhos de [Goltz 2011] e [Braga et al. 2016] utilizam uma RNA do tipo MLP (*MultiLayer Perceptron*) treinado com o algoritmo conhecido por *back-propagation* para a detecção de bordas, sendo esta técnica utilizada neste trabalho.

3. Resultados

O modelo do equipamento a ser utilizado neste trabalho constitui-se de uma placa *NVIDIA Jetson TX2* [NVIDIA 2014], que integra a linha de produtos da arquitetura "Pascal", a qual é configurada com duas CPUs ARM heterogêneas que totalizam 6 núcleos, uma GPU com 256 *Cuda cores* e uma memória principal de 8Gb. As placas GPUs Jetson são desenvolvidas para aplicações embarcadas, destacando-se o baixo consumo de energia (aproximadamente menos que 10 Watts).

Toda a otimização no código se concentrou no algoritmo de extração de bordas, uma vez que este corresponde a mais de 80% do tempo total de execução. A Tabela

	Média Frame (s)	Desvio padrão	SpeedUp
Single Core	0.0084462000	0.000445801909	1
OPENMP - 1	0.0080525328	0.0002017036807	1.048887376
OPENMP - 2	0.006359668133	0.0004045247754	1.272515038
OPENMP - 3	0.004488549467	0.000313083725	1.802981875
OPENMP - 4	0.0044657892	0.0009435951718	1,812170922
CUDA - 160	0,001971213333	0,0001721645138	4,105478183

Tabela 1. Resultados obtidos no Modo *Max-Q*

	Média Frame (s)	Desvio padrão	SpeedUp
Single Core	0.0060879867	0.0004371252352	1
OPENMP - 1	0.006438166267	0,0002541919151	0.9456087983
OPENMP - 2	0.01106070427	0.004805481648	0.5504158252
OPENMP - 3	0.0090835208	0.004570252743	0.6702232318
OPENMP - 4	0.009343199733	0.003221233991	0.6515954748
OPENMP - 5	0.01013126387	0.003350143123	0.6009108781
OPENMP - 6	0.00928976	0.003025165038	0.6553438051
CUDA - 160	0.0008493733333	0.0002549503466	7.167621619

Tabela 2. Resultados obtidos no Modo *Max-N*

1 mostra o resultado para o modo de operação denominado *Max-Q*, o qual representa o modo de menor consumo de energia possível na placa Jetson, onde foram desativados dois núcleos de computação, correspondente a um dos processadores (conhecido por *Denver 2*), enquanto a outra CPU (conhecida por *ARM-A57*) trabalha em uma frequência de *clock* de 1.2 GHz. O *clock* da GPU é definido em 0.85Ghz. Nesta configuração o *speedup* máximo com um código em *OpenMP* foi atingido utilizando-se a quantidade máxima de núcleos (quatro), entretanto, com uma eficiência computacional de apenas 45,3%. A versão em CUDA obteve um resultado expressivo mesmo trabalhando em *clock* reduzido, alcançando um resultado quase quatro vezes mais rápido que a versão de referência, e mais de duas vezes o desempenho em relação a melhor versão paralela com CPU.

A Tabela 2 mostra o resultado da configuração *MAX-N*, onde se espera encontrar o máximo de desempenho, com o dispositivo operando com os 2 núcleos da CPU *Denver 2* e os 4 núcleos do *ARM A57* em 2.0 GHz e a GPU em 1.3 GHz. Os resultados em *OpenMP*, no entanto, tiveram um desempenho abaixo do esperado, alcançando um valor de *speedup* menor que a versão com um processador. Por outro lado, a versão em CUDA teve um resultado expressivo, ficando 7 vezes mais rápido que a versão serial. Análises mais detalhadas para explicar o baixo desempenho das CPUs neste modo de operação deverão ser realizadas no futuro, onde se pretende investigar a alocação, comunicação e o gerenciamento de *threads* entre os dois processadores heterogêneos e paralelos presentes.

4. Conclusão

Neste trabalho foram apresentados resultados da versão em *OpenMP* e *CUDA*, de um simulador de navegação autônoma. A extração de bordas em sua versão *OpenMP* alcançou um resultado satisfatório operando no modo *MAX-Q* para o dispositivo utilizado, conseguindo diminuir o tempo de processamento em relação à versão serial, porém, em outros

modos de operação os resultados da versão OpenMP não foram os esperados. Entretanto, o desempenho da versão CUDA atingiu ganhos de até sete vezes em relação ao tempo de processamento serial, sendo, portanto, o indicado a ser utilizado em condições operacionais, em vista das análises atuais.

Conforme já mencionado anteriormente, o comportamento da versão paralela OpenMP pode ser investigado futuramente. Uma hipótese a ser analisada se deve ao fato de que o gerenciamento do *clusters* de processadores implica em um custo adicional de tempo no gerenciamento de múltiplas *threads*. Cabe citar que o desempenho do código em sua versão CUDA, executando na GPU, sempre alcançou melhor desempenho do que qualquer versão em CPU. O melhor desempenho foi obtido com o modo denominado *MAX-N*, onde a GPU opera em 1.3GHz com speedup de 7 vezes em relação à versão serial de referência. Entretanto, o consumo de energia é superior aos demais modos. O desempenho em GPU no modo de máxima economia de energia (*MAX-Q*), onde, conforme a especificação do dispositivo o consumo é de apenas 7.5W, operando no pico de operação, apresenta *speedup* de 4 vezes em relação à versão serial, o que já representa um ganho significativo.

Referências

- Braga, J. R. G., Velho, H. F. C., Conte, G., Doherty, P., and Shiguemori, É. H. (2016). An image matching system for autonomous uav navigation based on neural network. In *2016 14th ICARCV*, pages 1–6.
- Conte, G. and Doherty, P. (2008). An integrated uav navigation system based on aerial image matching. In *2008 IEEE Aerospace Conference*, pages 1–10.
- Goltz, G. A. (2011). Redes neurais artificiais em imagens para estimação da posição de um vant. Master's thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos.
- Goltz, G. A. M. and Shiguemori, E. H. (2007). Aplicação do algoritmo sift em imagens de navegação autônoma. In *Atividades de Pesq.e Desenv. do IEAV*, pages 5–12.
- He, Z. and Siyal, M. Y. (1998). Edge detection with bp neural networks. 2:1382–1384 vol.2.
- Li, W., Wang, C., Wang, Q., and Chen, G. (2008). An edge detection method based on optimized bp neural network. In *2008 International Symposium on Information Science and Engineering*, volume 2, pages 40–44.
- Longhitano, G. A. (2010). Vants para sensoriamento remoto - aplicabilidade na avaliação e monitoramento de impactos ambientais causados por acidentes com cargas perigosas. Master's thesis, Universidade de São Paulo, São Paulo.
- NVIDIA (2014). *NVIDIA Jetson TX2 - Datasheet*. NVIDIA Corporation.
- SANTOS, B. A. O. and Shiguemori, E. H. (2009). Determinação de pontos de controle em imagens aéreas e de videografia para aplicação na navegação autônoma. In *Seminário Anual de Iniciação Científica e Pós-Graduação do IEAV*, pages 41–42.
- Valavanis, K. and Vachtsevanos, G. (2014). *Handbook of Unmanned Aerial Vehicles*. Springer-Verlag GmbH.