

Identificação de Estradas em Imagens para uma Aeronave em Ambiente Simulado

Armando Alves Neto
Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Laboratório de Robótica e Visão Computacional
Belo Horizonte, Minas Gerais, Brasil - 2007
aaneto@dcc.ufmg.br

Resumo

Este artigo apresenta a implementação de um algoritmo robusto para a detecção de estradas em um ambiente de simulação de vôo, capaz de definir comandos para o sistema de controle lateral de uma aeronave virtual, utilizando para isso, princípios de Visão Computacional. Para tal, foi utilizado um ambiente simulado, do qual foi possível ler dados referentes aos estados de vôo da aeronave e escrever dados para as superfícies de controle da mesma. Além disso, foram capturadas imagens de uma camera virtual posicionada abaixo do veículo, apontada exatamente para baixo.

O sistema proposto, utiliza técnicas de visão computacional para identificar a posição e a orientação de uma estrada na imagem transmitida pelo simulador. Estas informações irão servir de parâmetros de referência para o controle latero-direcional da aeronave, que utiliza os dados de sensores da aeronave para calcular a ação de controle necessário para que o veículo siga a estrada identificada.

1. Introdução

A cada dia, cresce a aplicação das teorias de Visão Computacional aos sistemas de controle industriais e residenciais, e as cameras, cada vez menor e mais baratas, tem se tornado elementos de sensoriamento poderosos e versáteis.

Uma área de destaque na utilização de Visão Computacional é a de desenvolvimento de veículos aéreos autônomos não-tripulados, os chamados UAVs ¹, que são aeronaves capazes de voar sem qualquer tipo (ou com

o mínimo) de intervenção humana, e realizar missões pré-estabelecidas. Para que tais tarefas sejam realizadas, é necessário que, num nível mais baixo de implementação, estejam atuando diversos controladores de estados da aeronave, como a altitude, a velocidade e os ângulos de orientação espacial, por exemplo. Se for possível estabelecer uma condição de vôo estável para uma aeronave autônoma, a idéia é que se possa estabelecer missões em um nível mais auto, que sejam fundamentadas em ordens para estes estados de mais baixo nível.

Uma vez que as malhas de controle internas estejam bem ajustadas, pode-se utilizar a camera embarcada em uma aeronave para estabelecer comandos de mais auto nível na hierarquia de controle, estabelecendo missões que se reduzem a uma simples tarefa de seguir trajetórias ou marcos (*waypoints*) visuais, por exemplo. Neste caso, foi implementado um algoritmo identificador de estradas para gerar uma trajetória a ser seguida pelo sistema de controle lateral da aeronave, realizando assim, a missão de seguir uma estrada criada dentro do simulador de vôo.

2. Metodologia

Para realização deste trabalho foram realizadas basicamente duas etapas: a comunicação do software de controle com o simulador de vôo e o processamento da imagem de vídeo fornecida por este último. Tais etapas são descritas a seguir. Será descrita ainda a estratégia para uma futura elaboração do sistema de controle baseado no processamento desta imagem.

2.1. Simulador de Vôo

O simulador de vôo utilizado foi o *FlightGear Flight Simulator*, um programa de código aberto e multi-

¹ do inglês, *Unmanned Air Vehicle Systems*.

plataforma criado e gerenciado por Curt Olson, cujo principal objetivo era o de desenvolver um sofisticado sistema de simulação de vôo para uso principalmente acadêmico e de pesquisas em geral.

A grande vantagem da utilização do *FGFS*, além de sua total transparência, é a possibilidade do acesso, via rede, aos dados de vôo de uma aeronave no exato instante da simulação, que permite o processamento e visualização por outras entidades computacionais, em outras máquinas. Com isso, é possível amostrar dados dos diversos sensores de uma aeronave, processá-los por meio de um controlador externo, e gerar ações de controle de volta para o simulador. A figura 1 apresenta um *UAV (Rascal 110)* implementado dentro do *FGFS*, onde já estão implementados algumas das malhas internas de controle citadas anteriormente.



Figura 1. FlightGear Flight Simulator.

A figura 2 apresenta uma possível forma de comunicação entre o simulador e o programa de controle da missão. Para a leitura de todos os dados necessários a este procedimento são utilizadas três portas de rede, uma para a transmissão de imagens do *FGFS*, outras para a transmissão dos dados de vôo, e uma terceira para o envio dos sinais de controle para a aeronave virtual.

Para configurar toda esta transmissão de dados pelo simulador, é necessário inicializar o programa, passando os seguintes argumentos para a execução:

- -jpg-httpd=8880
- -native-fdm='socket,out,50,localhost,8881,udp'
- -native-ctrls='socket,in,50, ,8882,udp' .

O primeiro comando configura o simulador para funcionar como um servidor de imagens através da porta 8880, por onde são transmitidas figuras retiradas da tela do simulador. Ainda, é necessário compilar o FlightGear com suporte a esta opção, o que é feito após a compilação da bib-

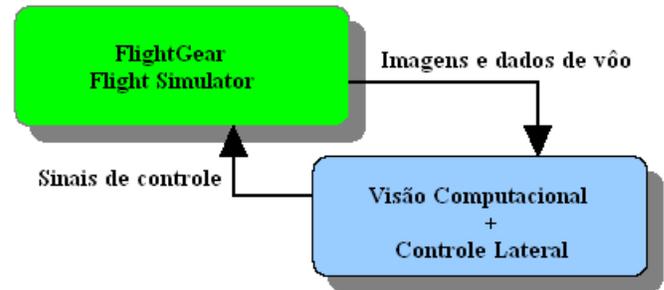


Figura 2. Comunicação de dados entre o simulador e o controlador.

lioteca *Simgear* (da qual depende o simulador) com a opção *JpgFactory*. Já o segundo comando fornece uma saída de dados de vôo do simulador, via protocolo *udp* através da porta 8881, a uma taxa de 50 hertz. O terceiro comando ainda configura o *FGFS* para escutar transmissões de dados de controle da porta 8882, também a uma taxa de 50 hertz. Os valores das portas e as taxas de transmissão de dados podem ser escolhidas arbitrariamente. Nas próximas sessões serão descritas as formas como estes dados são processados.

2.2. Visão Computacional

Conforme dito anteriormente, o sistema de Visão corresponde a uma etapa do processamento onde deseja-se encontrar a representação de uma estrada na imagem da camera embarcada na aeronave, para que esta possa segui-la em sua tarefa de navegação. Neste caso, a imagem adquirida corresponde à tela do simulador, que é transmitida via rede para o programa de controle, e a vista foi configurada apontando para baixo da aeronave virtual.

Nesta etapa do desenvolvimento do projeto foram encontrados basicamente dois sub-problemas menores, um relativo à identificação do que seria a estrada na imagem (detecção da estrada), e outro referente à transformação desta informação em dados relevantes para o controle da aeronave (localização da estrada).

2.2.1. Detecção da Estrada Para identificar a estrada em uma dada imagem, utilizou-se um algoritmo de segmentação de cores, baseado na idéia descrita em [1]. Neste artigo, a segmentação de cores foi utilizada para prover uma técnica robusta de subtração de fundo de uma dada imagem, para a qual, este fundo era conhecido. A idéia baseia-se no modelo apresentado na figura 3 onde as três cores, vermelha (*Red*), verde (*Green*) e azul (*Blue*) são orientadas como um conjunto de vetores ortogonais no espaço RGB.

Para cada pixel i da imagem avaliada, temos um ponto neste espaço, $I_i(r, g, b)$ que representa a cor daquele pixel na imagem. Já o vetor descrito por $E(r, g, b)$ representa a linha de crominância que vai da origem do referencial (cor preta) até uma dada cor específica, que é modelada pela norma deste vetor. Dois parâmetros importantes surgem a partir desta modelagem: α_i , denominado de *distorção de brilho*, e CD_i , chamado de *distorção de cor*. O primeiro mede a variação entre a intensidade de um pixel I_i e a cor E , ou seja, a diferença de luminosidade, dentro de uma mesma linha cromática. Já o segundo parâmetro, que aqui será mais relevante, mede a distância cromática entre o pixel I_i e a cor $\alpha_i E$, ou seja, a diferença entre os valores *RGB* de tais pontos.

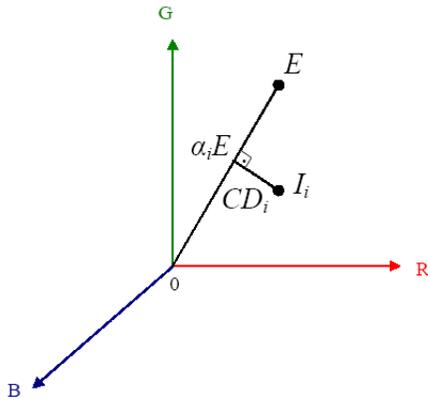


Figura 3. Modelo de cor no espaço RGB.

Assim, supondo que $E(r, g, b)$ represente o vetor que mais se aproxima da cor de uma estrada na imagem, então CD_i fornece uma medida da probabilidade de um dado pixel i pertencer ou não à uma estrada. A palavra probabilidade é utilizada por que, a modelagem na considera E como sendo um valor determinístico, mais sim, uma distribuição Gaussiana com $\mu(r, g, b)$ representando a média e $\sigma(r, g, b)$ o desvio padrão em *RGB* para os pixels da estrada.

A expressão 1 foi utilizada para o cálculo do parâmetro de distorção luminosa, ao passo que a expressão 2 nos dá a distorção cromática. Assim, para a imagem avaliada, temos uma matriz de valores de α_i com valores entre 0 e 1 (para baixas luminosidades) e maiores que 1 (para intensidades mais altas), e outra matriz de valores de CD_i , com valores sempre positivos.

$$\alpha_i = \frac{\sum_{c=r,g,b} \frac{I_i(c)\mu(c)}{\sigma(c)^2}}{\sum_{c=r,g,b} \left[\frac{\mu(c)}{\sigma(c)}\right]^2} \quad (1)$$

$$CD_i = \sqrt{\sum_{c=r,g,b} \frac{(I_i(c) - \alpha_i E(c))^2}{\sigma^2(c)}} \quad (2)$$

Para classificar um pixel como sendo *estrada* ou *não-estrada* escolher-se um certo valor τ_{CD} que representa um limiar para a distorção de valores de crominância entre os pixels da imagem e o modelo de estrada proposto. Assim, o algoritmo funciona da seguinte forma:

$$I_i(r, g, b) = \begin{cases} \text{estrada,} & CD_i < \tau_{CD}; \\ \text{não-estrada,} & \text{caso contrário.} \end{cases} \quad (3)$$

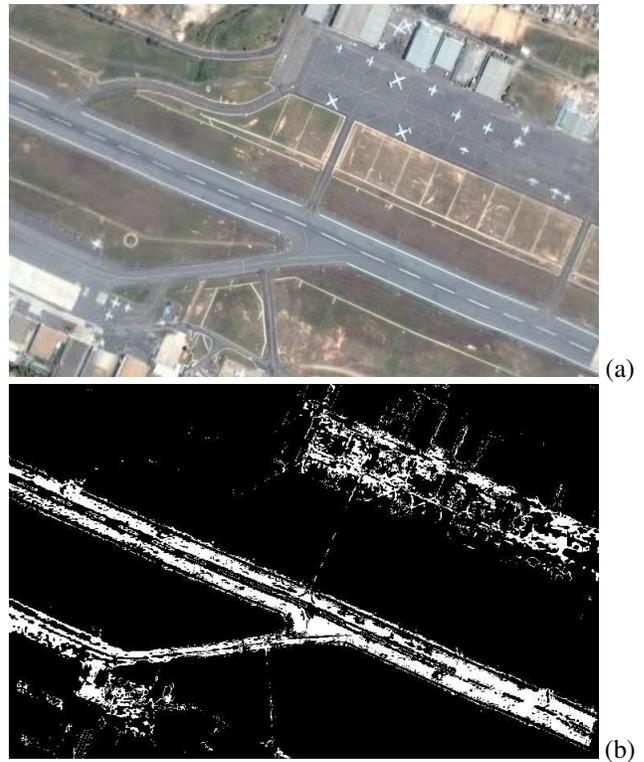


Figura 4. Algoritmo de identificação de estrada. (a) Imagem de uma vista aérea real. (b) Imagem segmentada.

A figura 4 apresenta o resultado da técnica aplicada a uma imagem real de uma pista, onde é possível ver que a estrada e outros elementos indesejáveis forma identificados com sucesso. Ainda para melhorar o resultado da imagem segmentada, pode ser necessária a aplicação de um filtro Gaussiano para eliminar o ruído, antes ou após a segmentação.

2.2.2. Localização da Estrada Uma vez detectados os pixels da imagem correspondentes à estrada, o passo seguinte foi o da localização espacial da estrada, com o objetivo de retirar as informações necessárias para guiar a aeronave. Basicamente, conhecendo-se a orientação e a posição da estrada, em relação à camera, torna-se possível instruir o sistema de controle lateral da aeronave à seguir a trajetória especificada.

Inicialmente, foi aplicado um filtro de detecção de bordas do tipo *Canny Edge Detector* [2], para a identificação principalmente dos limites da estrada na imagem. Este filtro, além de causar uma redução no ruído, produziu uma imagem de entrada para a identificação de retas através da função conhecida como *Transformada de Hough*. Por meio da utilização desta técnica, foi possível identificar todas as retas formadas pelas bordas da imagem segmentada da estrada, obtendo um indicativo de qual seriam as posições e orientações das linhas da estrada na imagem.

Foi escolhido um valor relativamente alto para o limiar da *Transformada de Hough*, de modo que fossem identificadas apenas as maiores retas da imagem amostrada. Com isso, a chance de identificação de linhas pertencentes à estrada aumentariam. Aplicando-se a técnica à imagem da figura 4(b), obteve-se o resultado mostrado na figura 5.

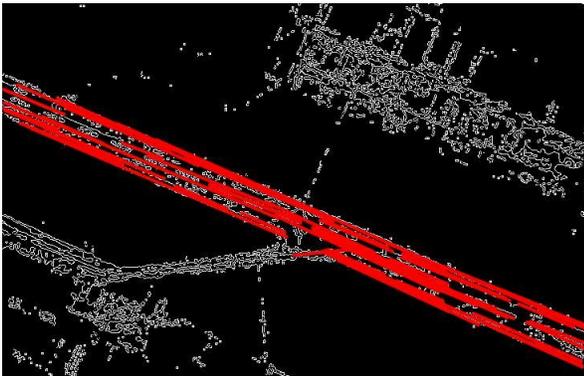


Figura 5. Transformada de Hough aplicada à imagem.

A maior parte das retas (em vermelho) encontradas na imagem possuem orientações e posições semelhantes à da estrada principal da figura real, conforme pode ser visto com a comparação entre as figuras. Segundo o algoritmo de *Hough*, uma reta i é modelada como sendo um par (ρ_i, θ_i) que correspondem à posição e a orientação da mesma no plano da imagem. Assim, para todas as N retas encontradas na imagem da estrada, temos um conjunto de N pares (ρ, θ) , cujo N pode variar a cada interação do algoritmo, pois depende do conteúdo da imagem.

Para o caso em que a estrada esteja visível na tela do simulador, e pelo menos uma reta seja identificada pelo processamento posterior, podemos assumir que a média dos valores θ_i ao longo das N retas fornecem um indicativo da orientação da estrada em relação à camera da aeronave. Da mesma forma, a média dos N valores de ρ_i correspondem à posição da estrada na imagem. Assim, temos que:

$$\rho_{ref} = \frac{1}{N} \sum_{i=1}^N \rho_i \quad (4)$$

$$\theta_{ref} = \frac{1}{N} \sum_{i=1}^N \theta_i \quad (5)$$

2.3. Controle Lateral

Após a etapa de processamento da imagem, e identificação dos parâmetros da representação da estrada na imagem da camera, o passo futuro seria a elaboração da estratégia de controle lateral para direcionar a aeronave dentro do simulador. Conforme dito anteriormente, o *FlightGear* permite o recebimento via rede dos comandos de superfície da aeronave, ou seja, leme, profundor, aileron e motor (conforme figura 6).

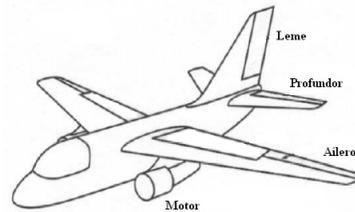


Figura 6. Superfícies de controle de uma aeronave.

O simulador já implementa toda uma gama de laços de controle internos para as aeronaves virtuais, como controle de altitude, velocidade e guinada², todos controladores do tipo PID. Uma possível abordagem para guiar a aeronave pela trajetória pela estrada seria, por exemplo, utilizar os controles de altitude e de velocidade do *FGFS* para estabilizar o vôo de uma aeronave, enquanto que o controle do ângulo de guinada seria implementado externamente, seguindo a referência de localização e orientação da estrada, calculadas em (4) e (5). Basicamente, o sinal do leme (δ_r)

² ângulo que a aeronave faz com o Norte da Terra (*heading*).

da aeronave, que possui maior influência sobre seu movimento lateral pode então ser calculado a partir da expressão (6) e enviado para o simulador, a cada interação do algoritmo. Aqui, $f()$ representa uma função linear que descreve o comportamento do controlador PID, e que recebe as variáveis de referências, e suas respectivas derivadas e integrais temporais.

$$\delta_r = f(\rho_{ref}, \theta_{ref}, \dot{\rho}_{ref}, \dot{\theta}_{ref}, \int \rho_{ref}, \int \theta_{ref}) \quad (6)$$

2.4. Resultados

As figuras de 7 à 10 apresentam alguns dos resultados alcançados. Elas representam todas as imagens amostradas e processadas em um mesmo laço de interação do controlador.



Figura 7. Imagem do simulador.

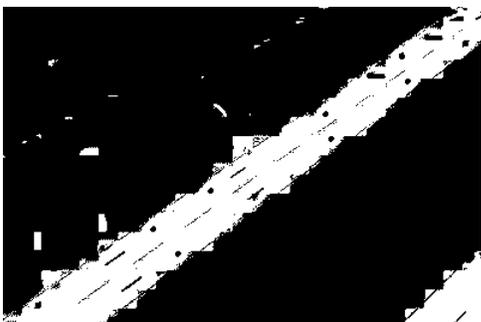


Figura 8. Detecção de estrada (resultado)

A primeira imagem é a tela captura do simulador, a qual serve de base para as etapas posteriores de processamento. Já na imagem de detecção da estrada, é possível observar que a grande maioria dos pixels foram bem identificados, apesar do ruído causado, entre outras razões, pela compressão *JPEG* da imagem da camera virtual.

As retas em vermelho, detectadas na última imagem são a saída do algoritmo de *Hough*, indicando quais são as principais *features* detectadas referentes à estrada. A reta verde representa a média das linhas encontradas, ou seja, a direção a ser seguida pela aeronave.

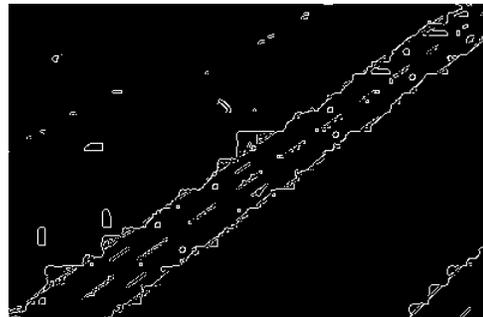


Figura 9. Detecção de bordas.

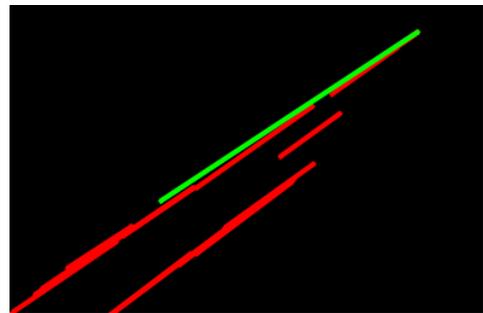


Figura 10. Localização da estrada (resultado). As retas vermelhas são a saída do Hough e a reta verde corresponde à referência.

As figuras de 11 à 14 representam um outro instante do *pipeline* de processamento do algoritmo.

2.5. Conclusões e Trabalhos Futuros

De um modo geral, os resultados encontrados mostraram-se bastante satisfatórios. Do ponto de vista visual, a técnica empregada se mostrou bastante robusta, embora a modelagem de cores no espaço RGB não corresponda à melhor forma de representação da imagem, para os problemas de segmentação.

Com relação à localização da estrada na imagem, pode-se dizer que a transformada de *Hough*, aplicada à imagem segmentada mostrou-se uma ferramenta eficiente para estimar a orientação da trajetória da aeronave. Contudo, o



Figura 11. Imagem do simulador.

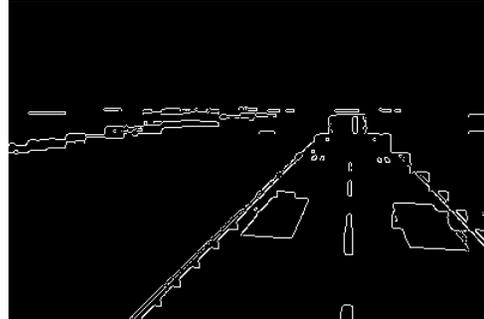


Figura 13. Detecção de bordas.

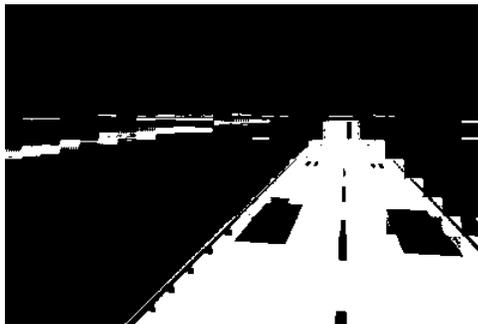


Figura 12. Detecção de estrada (resultado)

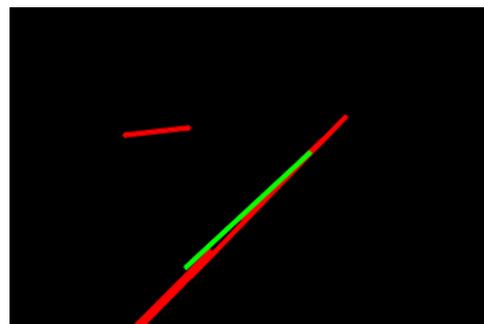


Figura 14. Localização da estrada (resultado). As restas vermelhas são a saída do Hough e a reta verde corresponde à referência.

cálculo da média para a estimativa desta direção não é a melhor estratégia, já que é bastante susceptível à ruído.

A próxima etapa no desenvolvimento desta tarefa, como já foi dito, constitui a elaboração da estratégia de controle lateral, externa ao simulador, para guiar a aeronave por sobre a estrada identificada. Este é sem dúvida, um problema que abre caminho para a implementação de outras tarefas e missões mais complexas para a aeronave autônoma.

Referências

- [1] Thanarat Horprasert, David Harwood e Larry S. Davis, "A Robust Background Subtraction and Shadow Detection," *ACCV 2000*.
- [2] Emanuele Trucco e Alessandro Verri, *Introductory techniques for 3-D computer vision*, ed. 2, Upper Saddle River, NJ: Prentice Hall, 1998. 343p..
- [3] Eric Frew, Tim McGee, ZuWhan Kim, Xiao Xiao, Stephen Jackson, Michael Morimoto, Sivakumar Rathinam, Jose Padiual, e Raja Sengupta, "Vision-Based Road-Following Using a Small Autonomous Aircraft," *Proceedings of the 2004. IEEE Aerospace Conference, 2004..*
- [4] Curt Olson, "FlightGear - Getting Start," www.flightgear.org/Docs/getstart/getstart.html, acesso 15/03/2007.
- [5] Jaesang Lee e Carl D. Crane, "Road Following in an Unstructured Desert Environment using Monocular Color Vision as

Applied to the DARPA Grand Challenge," *Proceedings of the 2004..*

- [6] S. Saripalli, J.F. Montgomery e G.S. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," *Proceedings of the 2002 International Conference on Robotics and Automation, 2002*.
- [7] Peter I. Corke, *Visual Control of Robots: High-Performance visual servoing*, ed. 1, Mechatronics, Research Studies Press (John Wiley), 1996.