

Verificação Automática de Situações de Colisão na Navegação de Veículos Aéreos Não Tripulados

Felipe Leonardo Lôbo Medeiros, Elcio Hideiti Shiguemori, Marcus Vinicius Teixeira Monteiro, Marco Antônio Pizani Domiciano, Mauricio Pozzobom Martins

Instituto de Estudos Avançados (IEAv) – Comando-Geral de Tecnologia Aeroespacial (CTA)

São José dos Campos - SP - Brasil

{felipe, elcio, monteiro, pizani, mauricio}@ieav.cta.br

Abstract. *This work presents a technique for automatic verification of collision situations in the navigation of Unmanned Aerial Vehicles (UAV's) through computational representations of the navigation environment. A collision situation is that one where the UAV will be navigating in direction, or very near, to an obstacle of the environment. The air navigation, considered in this work, is the bidimensional one, that is, to constant altitudes. The verification technique was implemented and connected to an air navigation system. After that, they had been tested in aircraft simulated for the computational program FlightGear. Results are presented in this work.*

Resumo. *Este trabalho apresenta uma técnica para verificação automática de situações de colisão na navegação de Veículos Aéreos Não Tripulados (VANT's) através de representações computacionais do ambiente de navegação. Uma situação de colisão é aquela em que o VANT se encontrará navegando em direção, ou muito próximo, a um obstáculo do ambiente. A navegação aérea, considerada neste trabalho, é a bidimensional, isto é, a altitudes constantes. A técnica de verificação foi implementada e acoplada a um sistema de navegação aérea. Em seguida, foram testados em aeronaves simuladas pelo programa computacional FlightGear. Resultados são apresentados nesse trabalho.*

1. Introdução

Os sistemas de localização da maioria dos VANT's são baseados na integração entre sensores inerciais e o *Global Positioning System* (GPS). No projeto Processamento de Imagens em Tempo Real (PITER), em desenvolvimento no Instituto de Estudos Avançados (IEAv), está sendo pesquisado um sistema de navegação autônoma baseado em visão computacional [Martins *et al.* 2006]. Na fase atual do projeto, a navegação considerada é a bidimensional, isto é, a altitudes constantes. A navegação tridimensional também será abordada em fases futuras do projeto.

O sistema de visão computacional consiste em capturar imagens do solo e compará-las, em tempo real, com um mosaico de imagens georeferenciadas, com o objetivo de estimar a localização/coordenada do veículo. As imagens capturadas são processadas e passam por um subsistema de reconhecimento de padrões para que a localização do VANT seja estimada. O sistema inercial do VANT é então corrigido com a localização estimada.

O intervalo de tempo necessário para a estimação da localização é um grande problema para o sistema de navegação do projeto PITER. Durante este intervalo de tempo, tal sistema de navegação utiliza a localização proporcionada pelo sistema inercial do VANT. Porém, o sistema inercial gera um erro acumulativo na localização, o que pode fazer com que o sistema de navegação coloque o VANT em uma situação de colisão, isto é, navegando em direção, ou muito próximo, a um obstáculo. Para solucionar este problema, surgiu a necessidade de dotar o sistema de navegação autônoma do projeto PITER de um subsistema de verificação de situações de colisão do VANT e de um subsistema de replanejamento de trajetórias [Medeiros *et al.* 2006].

O subsistema de replanejamento de trajetórias gera uma trajetória segura, evitando obstáculos quando o subsistema de verificação informa que o VANT se encontra em uma situação de colisão. Esse subsistema foi elaborado a partir do algoritmo A* [Ferguson *et al.* 2006].

Este trabalho apresenta a técnica usada no subsistema desenvolvido para a verificação de situações de colisão. O subsistema é baseado em representações computacionais do ambiente de navegação por meio de grades regulares [Konar 2000, Bader 2003], em que cada célula da grade corresponde a uma região do ambiente de navegação e indica se há ou não obstáculo à navegação na respectiva região.

A seção 2 apresenta uma descrição da forma de representação computacional do ambiente de navegação de VANT's. A seção 3 descreve a técnica criada para a verificação de situações de colisão à navegação aérea. As seções 4 e 5 apresentam, respectivamente, os resultados obtidos e as conclusões.

2. Representação do ambiente de navegação aérea

A representação do ambiente de navegação é gerada automaticamente por meio de grades regulares binárias [Konar 2000, Bader 2003 e Carlisle 2006], através da matriz de altitudes do modelo digital de elevação do ambiente. Cada célula da matriz de altitudes indica a altitude/elevação de uma pequena região do ambiente. E cada célula da grade corresponde a um conjunto de células adjacentes da matriz de altitudes, correspondendo a uma região maior do ambiente de navegação. Cada célula da grade binária assume o valor 1 se há pelo menos uma célula da matriz de altitude com valor igual ou superior a certo limite de segurança da altitude desejada de navegação. A célula com valor 1 assume a denominação de célula obstáculo. Esse limite garante segurança à navegação, pois é igual à diferença entre a altitude de navegação e uma altura de segurança de vôo. A altitude de navegação, a altura de segurança e o tamanho da célula da grade dependem da dinâmica de cada VANT. É conveniente adotar o tamanho da célula de acordo com o raio de curva do VANT, de forma a permitir que o mesmo consiga realizar uma curva de 90° dentro da célula. As Figuras 2.1 e 2.2 apresentam exemplos da representação do ambiente de navegação por meio de grades regulares.

Na Figura 2.1 é possível verificar a obtenção dos valores de uma seqüência de uma grade regular por meio da análise da matriz de altitudes de uma seção de um ambiente de navegação, da altitude de navegação e da altura de segurança. As células 6 a 9, 12 e 13 representam os obstáculos à navegação.

Na Figura 2.2 há uma representação bidimensional de um ambiente de navegação. Na Figura 2.2.a, as regiões na cor preta indicam os obstáculos à navegação, que são células da matriz de altitude com valor superior ao limite de segurança. A Figura 2.2.b

representa a grade regular equivalente com as células obstáculo na cor preta indicando o valor 1.

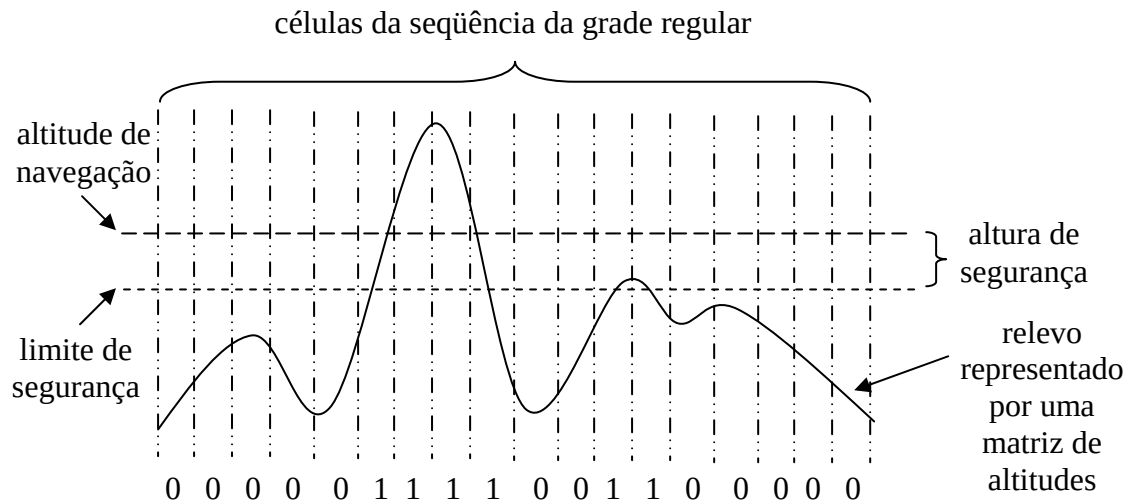


Figura 2.1. Exemplo de obtenção de grade regular

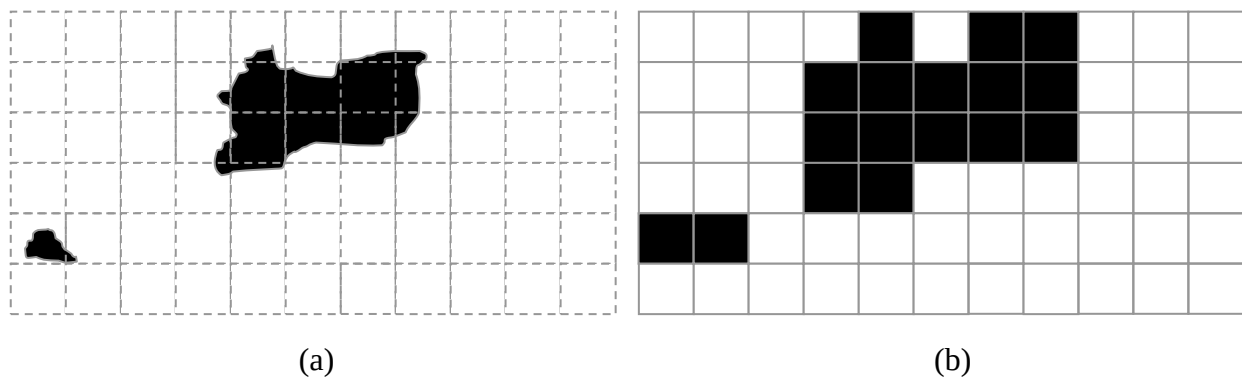


Figura 2.2. Exemplo de representação de ambiente por meio de grade regular

Na representação por meio de grades regulares o aumento do número de células (ordem) da grade gera uma melhor representação dos obstáculos do ambiente de navegação. Porém, gera um aumento do custo computacional tempo. Mas a redução significativa do número de células gera situações em que apenas uma pequena parte da região representada pela célula é um obstáculo.

3. Técnica de verificação de situações de colisão

A técnica de verificação de situações de colisão evita que um VANT navegue em direção, ou muito próximo, a obstáculos do ambiente. A cada intervalo de tempo Δt , a técnica consiste em verificar se o segmento de reta, que vai da coordenada atual (ca) do VANT até a próxima coordenada (cp) de navegação a ser alcançada, corta ou passa muito próximo a algum obstáculo do ambiente. Essa verificação é um processo iterativo em que uma sequência de coordenadas c_i são obtidas e analisadas se pertencem ou não à região representada por alguma célula obstáculo. Duas coordenadas c_{i1} e c_{i2} , cujo segmento de reta $c_{i1}c_{i2}$ é perpendicular ao segmento de reta $cacp$, são calculadas a partir de c_i com a finalidade de verificar quão próxima está o segmento de reta $cacp$ de algum obstáculo. De forma semelhante a c_i , analisa-se se cada c_{i1} e c_{i2} pertence a alguma

região representada por uma célula obstáculo. O processo termina quando alguma coordenada c_i , c_{i1} ou c_{i2} pertence a alguma célula obstáculo ou quando o número total de iterações/passos é alcançado. A Figura 3.1 apresenta um exemplo da aplicação da técnica de verificação de situações de colisão. Na Figura, p é o tamanho do passo, h_i são as alturas obtidas a cada iteração e α é o ângulo usado para o cálculo das coordenadas e das alturas. O ângulo α é obtido a partir da proa do VANT, o que explica a presença das indicações N (Norte), S (Sul), L (Leste) e O (Oeste). As coordenadas c_i e c_f são respectivamente:

- a coordenada inicial que delimita o ambiente de navegação. Corresponde à primeira coordenada do canto inferior à esquerda da região representada pela célula do canto inferior à esquerda da grade regular; e
- a coordenada final que delimita o ambiente de navegação. Corresponde à última coordenada do canto superior à direita da região representada pela célula do canto superior à direita da grade regular;

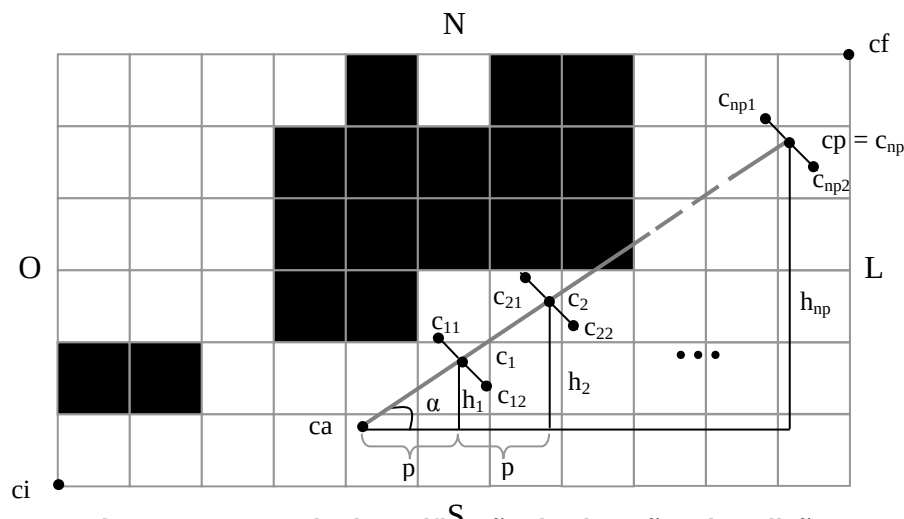


Figura 3.1. Exemplo de verificação de situações de colisão

O número de passos ou iterações deve ser especificado de forma a permitir que a verificação seja válida. Um número elevado de passos tende a gerar um elevado custo computacional de tempo. Um número muito reduzido de passos causa distâncias maiores entre as coordenadas c_i calculadas, o que pode acarretar um erro, pois uma célula obstáculo pode ficar entre duas coordenadas c_i do segmento de reta formado por ca e cp . Este problema pode ser verificado através de uma simples alteração no exemplo da Figura 3.1. Supondo que a primeira coordenada, c_1 , seja a coordenada c_2 na Figura 3.1, o corte da célula obstáculo não seria verificado, porque c_2 seria obtida após a célula obstáculo. Observando que as coordenadas c_i são igualmente espaçadas. Para solucionar este problema, foi adotado o cálculo do número de passos np baseado no comprimento da base ou da lateral da célula da grade regular. O cálculo de np se divide em dois casos, de acordo com o ângulo β formado pelo segmento de reta $cacp$ e o Norte geográfico. Portanto β é uma direção de vôo desejada, dada em graus de bússola. A divisão em dois casos ocorre com o objetivo de facilitar o cálculo das coordenadas quando a longitude de ca tende à longitude de cp . A Figura 3.2 apresenta as possíveis variações do ângulo β .

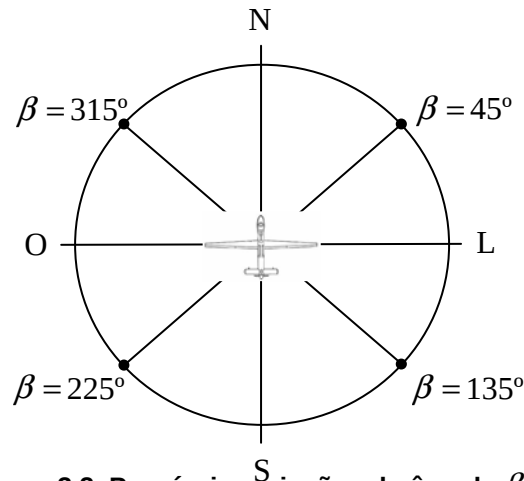


Figura 3.2. Possíveis variações do ângulo β

Os dois casos do cálculo de np são apresentados a seguir.

Caso 1 - Se $45^\circ \leq \beta \leq 135^\circ$ e $225^\circ \leq \beta \leq 315^\circ$

$$np = \frac{|(\text{longitude}(ca) - \text{longitude}(cp))|}{p} \quad (3.1)$$

Sendo que

$$p = \frac{cb}{C} \quad (3.2)$$

$$cb = \frac{|(\text{longitude}(ci) - \text{longitude}(cf))|}{nc} \quad (3.3)$$

Caso 2 - Para os demais valores de β

$$np = \frac{|(\text{latitude}(ca) - \text{latitude}(cp))|}{p} \quad (3.4)$$

Sendo que

$$p = \frac{cl}{C} \quad (3.5)$$

$$cl = \frac{|(\text{latitude}(ci) - \text{latitude}(cf))|}{nl} \quad (3.6)$$

Onde:

- cb é o comprimento da base de uma célula da grade;
- cl é o comprimento da lateral de uma célula da grade;
- nc é o número de colunas da grade regular;
- nl é o número de linhas da grade regular; e
- C é um parâmetro de controle do tamanho do passo.

Essa forma de calcular o tamanho do passo e o número de passos permite que pelo menos uma coordenada c_i , c_{i1} ou c_{i2} seja gerada dentro de cada célula cortada pelo segmento $cacp$, ou próxima ao segmento. O parâmetro C no cálculo do tamanho do passo foi obtido empiricamente, podendo ser alterado de acordo com a necessidade. Deve ser observado que quanto maior o valor de C , menor o tamanho do passo e maior o número de passos, logo, maior o tempo computacional exigido.

O cálculo das coordenadas c_i , c_{i1} e c_{i2} também está baseado nos dois casos apresentados anteriormente para o cálculo de np . Porém, cada um deles é dividido em quatro tipos de subcasos: 1.1, 1.2, 1.3 e 1.4 para o caso 1; e subcasos 2.1, 2.2, 2.3 e 2.4 para o caso 2. Os oito subcasos são apresentados na Figura 3.3. O primeiro tipo de subcaso ocorre quando a latitude de ca é menor ou igual à latitude de cp e a longitude de ca é menor ou igual à longitude de cp . O segundo tipo é derivado do primeiro tipo com a substituição de ca por cp . O terceiro tipo de subcaso ocorre quando a latitude de ca é menor que a latitude de cp e a longitude de ca é maior que a longitude de cp . O quarto tipo é derivado do terceiro com a substituição de ca por cp .

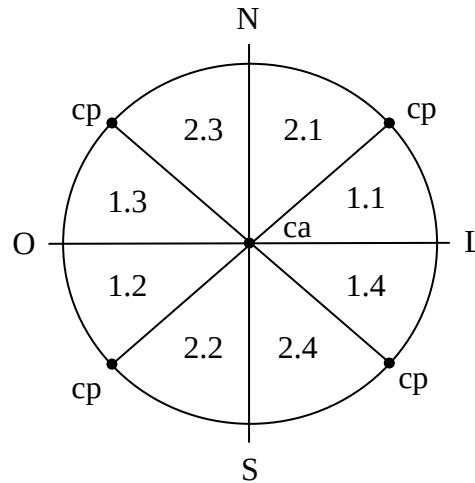


Figura 3.3. Casos e subcasos para o cálculo de c_i , c_{i1} e c_{i2}

Os cálculos das coordenadas c_i , c_{i1} e c_{i2} , para os subcasos 1.1 e 2.1, são apresentados na Figura 3.4. No subcaso 1.1, as coordenadas são calculadas por

$$longitude(c_i) = longitude(ca) + i p \quad (3.7)$$

$$latitude(c_i) = latitude(ca) + h_i \quad (3.8)$$

Sendo que

$$h_i = i \operatorname{ptg}(\alpha) \quad (3.9)$$

$$\alpha = 90^\circ - \beta \quad (3.10)$$

Onde:

- i é o índice do passo/iteração ($i = 1, \dots, np$);
- p é calculado por 3.2 e 3.3; e
- dms é um parâmetro de controle da distância de segurança entre o segmento de reta $cacp$ e um obstáculo. Quanto maior o valor de dms , maior o segmento de

reta $c_{i1}c_{i2}$, portanto, maior será a segurança para a navegação. Neste trabalho, o valor de dms foi obtido empiricamente como sendo $dms = 5$.

No subcaso 2.1, as coordenadas são calculadas por

$$longitude(c_i) = longitude(ca) + h_i \quad (3.11)$$

$$latitude(c_i) = latitude(ca) + i p \quad (3.12)$$

Sendo que

$$h_i = i p \operatorname{tg}(\beta) \quad (3.13)$$

Onde:

- p é calculado por 3.5 e 3.6.

As coordenadas c_{i1} e c_{i2} para os dois subcasos são calculadas por

$$longitude(c_{i1}) = longitude(c_i) - dms p \operatorname{sen}(\alpha) \quad (3.14)$$

$$latitude(c_{i1}) = latitude(c_i) + dms p \cos(\alpha) \quad (3.15)$$

$$longitude(c_{i2}) = longitude(c_i) + dms p \operatorname{sen}(\alpha) \quad (3.16)$$

$$latitude(c_{i2}) = latitude(c_i) - dms p \cos(\alpha) \quad (3.17)$$

A técnica, criada e apresentada neste trabalho, é computacionalmente mais rápida que a verificação da própria matriz de altitudes do modelo digital de elevação, pois há uma discretização da representação. Porém, como mencionado anteriormente, a utilização de uma grade regular leva a perda da definição dos obstáculos. Isto implica que há casos em que a técnica verifica a existência de uma situação de colisão, sendo que apenas uma pequena porção da região representada pela célula é um obstáculo.

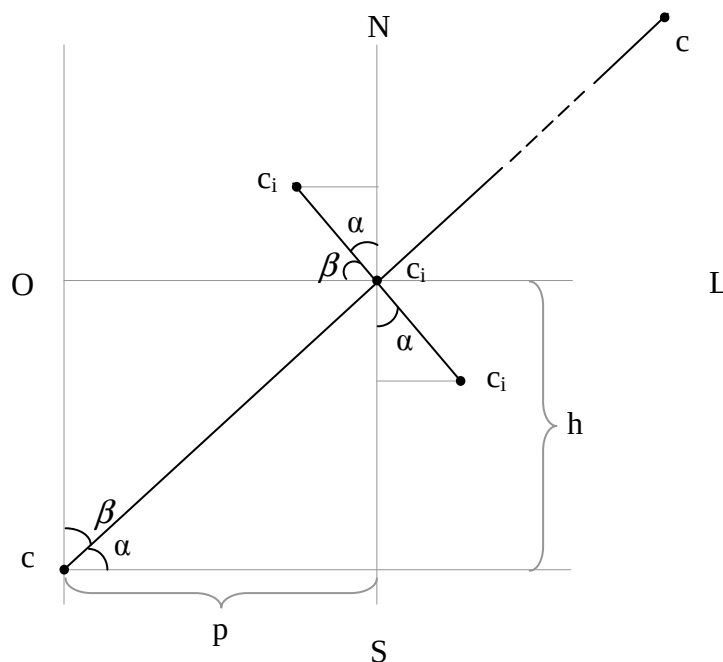


Figura 3.4. Cálculo de c_{i1} , c_{i2} e c_{i3} nos subcasos 1.1 e 2.1

É importante ser observado que o tempo computacional, para a aplicação de uma implementação da técnica, é diretamente proporcional ao tamanho da grade regular e o parâmetro C .

4. Resultados

Os testes da técnica de verificação de situações de colisão foram baseados em simulações de um VANT navegando a uma altitude e velocidade constantes através de uma rota planejada. Nos testes, o VANT deveria ser capaz de replanejar uma trajetória de segurança sempre que uma situação de colisão fosse verificada. A trajetória de segurança deveria ser calculada de forma a permitir, se possível, o VANT retornar à rota planejada, após o desvio do obstáculo.

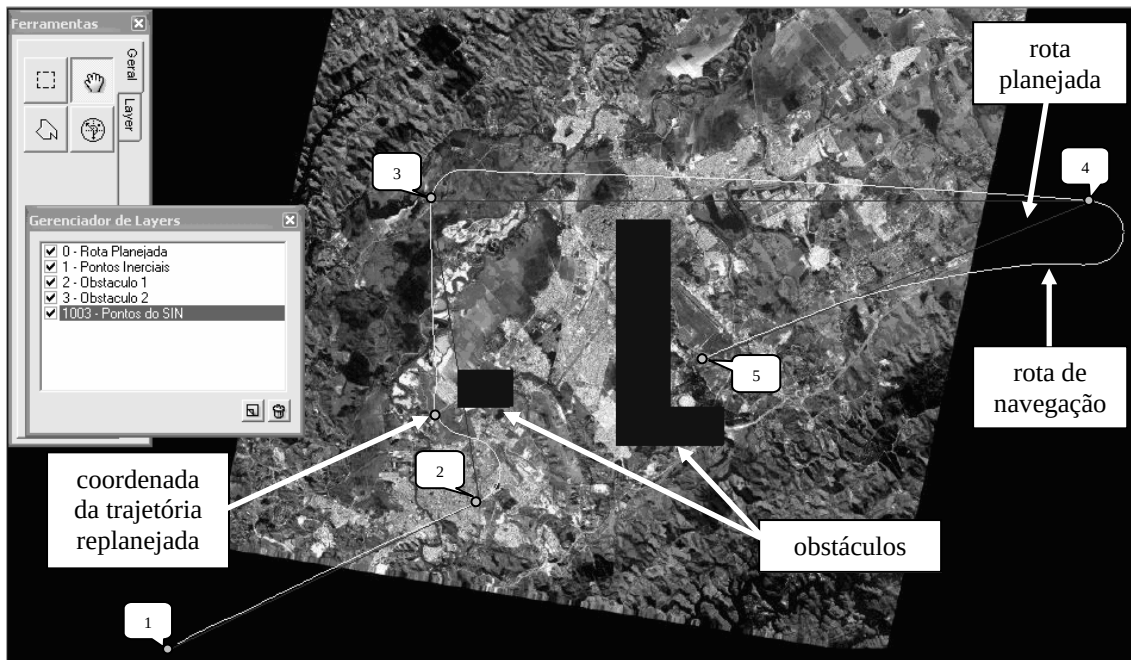
A navegação do VANT foi realizada através do sistema de navegação autônoma do projeto PITER. A técnica de verificação de situações de colisão foi implementada como um subsistema desse sistema de navegação. O subsistema de verificação de situações de colisão informa ao subsistema de replanejamento de trajetórias se há ou não uma situação de colisão. Neste trabalho, o subsistema de verificação é acionado a cada segundo, isto é, $\Delta t = 1s$. Este intervalo de tempo foi considerado como o limite para uma eficiente verificação de situação de colisão.

O VANT foi simulado por meio do simulador de vôo *FlightGear* [Medeiros *et al.* 2006]. O *FlightGear* é um programa computacional de código aberto, que permite a simulação de diversas aeronaves em condições climáticas variadas e a utilização de modelos digitais de elevação. Permite também a especificação de diferentes modelos dinâmicos para cada aeronave, de acordo com o grau de realismo desejado. A aeronave usada nos testes, para simular um VANT, é um Cessna 310. Um modelo digital de elevação de São José dos Campos foi o ambiente de navegação escolhido para todos os testes. Este modelo foi inserido no *FlightGear*. A Figura 4.1 exibe imagens capturadas de um teste realizado considerando o modelo digital de elevação de São José dos Campos.

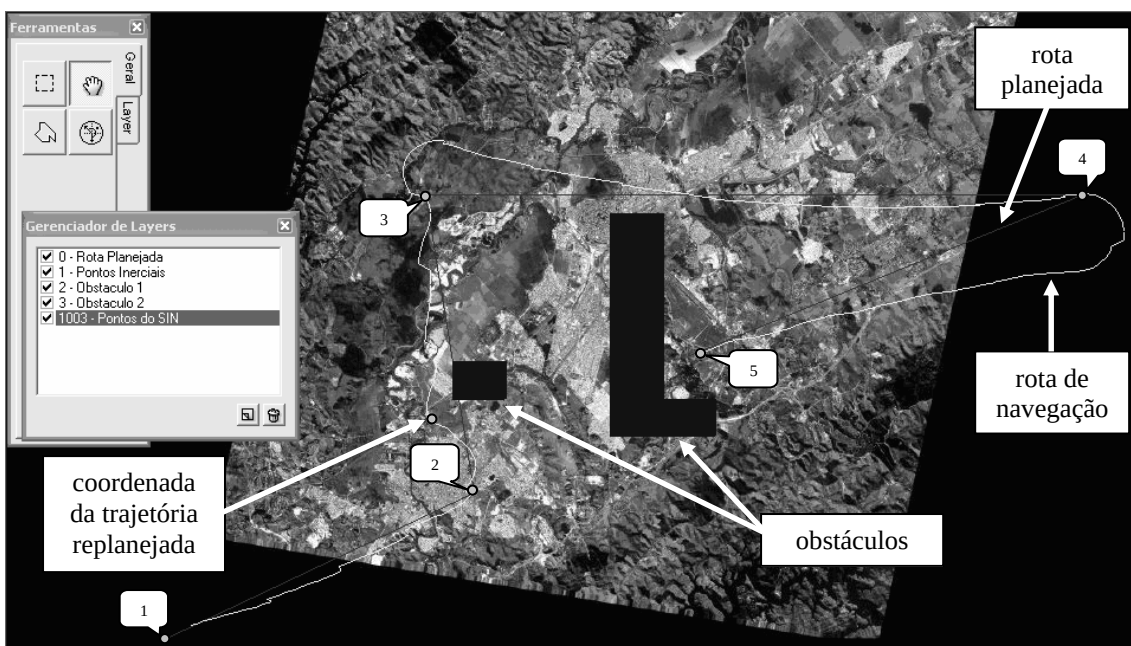
Os testes foram feitos com diferentes valores do parâmetro C e grades de diferentes ordens, variando de 10×10 a 200×200 . Uma análise do custo computacional tempo, considerando os piores casos, é apresentada na Tabela 4.1. Observa-se que vários testes ocorreram dentro do limite $\Delta t = 1s$. O computador usado foi um Pentium IV de 2 GHz e 1 GB de RAM. A Figura 4.2 apresenta dois testes feitos com a técnica, para uma grade regular de ordem 20×20 e as mesmas configurações de vôo (aeronave, rota, altitude, velocidade, etc). Propositamente, o segmento de reta formado pela segunda e a terceira coordenada da rota planejada corta um obstáculo do ambiente. A Figura 4.2.a apresenta uma navegação, em que a estimação da posição é baseada na integração entre sensores inerciais e o GPS. Na Figura 4.2.b a estimação é baseada na integração entre sensores inerciais e o sistema de visão computacional do PITER. Percebe-se que a navegação por inercial/GPS é mais suave que a navegação por inercial/visão computacional. Este fato é explicado pelo intervalo de tempo necessário para o processamento de imagem e estimação da posição. A técnica de verificação de obstáculos foi eficiente, permitindo um replanejamento de trajetória logo após o VANT alcançar a segunda coordenada, o que possibilitou o desvio do obstáculo com segurança. As imagens foram geradas através do programa computacional Sistema de Vigilância por Satélites (SIVISAT), desenvolvido no IEAv.



Figura 4.1. Imagens capturadas de um teste com o *FlightGear* simulando um Cessna 310



(a)



(b)

Figura 4.2. Dois testes considerando diferentes formas de estimação de posição

Tabela 4.1. Comparações de tempo para diferentes grades e valores do parâmetro C

Ordem da grade	C = 2	C = 5	C = 10	C = 20
10x10	$< 10^{-3}$ s	$< 10^{-3}$ s	$< 10^{-3}$ s	$< 10^{-3}$ s
20x20	$< 10^{-3}$ s	$< 10^{-3}$ s	$< 10^{-3}$ s	0.007 s
50x50	0.016 s	0.031 s	0.062 s	0.125 s
100x100	0.094 s	0.234 s	0.453 s	1.016 s
200x200	0.734 s	1.843 s	3.672 s	7.343 s

5. Conclusões

Os resultados obtidos por meio dos testes comprovaram a eficiência da técnica criada para a verificação de situações de colisão na navegação aérea. Em todos os testes, as situações de colisão foram verificadas em tempo hábil para o replanejamento de trajetórias e a correção da navegação. Uma aplicação promissora para a técnica está nos VANT's de combate, em que há a necessidade de replanejamento de trajetórias com a finalidade de controlar o VANT nas situações de combate e de enviá-lo a unidades de reabastecimento. Outra aplicação promissora seria a utilização da técnica como alerta em situações de risco à navegação de aeronaves tripuladas. A técnica também poderia ser aplicada a problemas de replanejamento em ambientes desconhecidos. Desde que os sensores possam fornecer informações através das quais seja possível gerar, em tempo real, uma representação do ambiente de navegação.

O próximo trabalho será a adaptação da técnica para verificação de situações de colisão na navegação a altitudes variáveis, ou navegação tridimensional. Neste caso, haverá a necessidade de pesquisas relacionadas a representações tridimensionais do ambiente de navegação aérea.

Referências

- Carlisle, P. (2006). A Combat Flight Simulation AI Framework. In *Game Programming Wisdom 3*, páginas 517-528. Charles River Media.
- Ferguson, D., Likhachev, M., Stentz, A. (2005) "A Guide to Heuristic-based Path Planning", Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems, International Conference on Automated Planning and Scheduling (ICAPS), p. 9-18.
- Konar, A. (2000), Artificial Intelligence and Soft Computing – Behavioral and Cognition Modeling of the Human Brain, CRC press.
- Martins, M. P., Monteiro, M. V. T., Shiguemori, E. H., Medeiros, F. L. L., Domiciano, M. A. P. (2006) "Navegação Aérea Autônoma por Imagens", VI WAI – Workshop Anual de Pesquisa e Desenvolvimento do Instituto de Estudos Avançados, São José dos Campos.
- Medeiros, F. L. L., Ferreira, L. C. A., Shiguemori, E. H., Domiciano, M. A. P., Monteiro, M. V. T., Martins, M. P. (2006) "Integração do Módulo de Navegação Autônoma do Projeto PITER com o SIVISAT e o Simulador de Vôo FlightGear", VI WAI – Workshop Anual de Pesquisa e Desenvolvimento do Instituto de Estudos Avançados, São José dos Campos.
- Tozour, P. (2003). Search Space Representations. In *AI Game Programming Wisdom 2*, páginas 85-102. Charles River Media.