

AMBIENTE VIRTUAL PARA SIMULAÇÃO DE ROBÔS MÓVEIS INTELIGENTES APLICADOS NO COMBATE DE INCÊNDIOS FLORESTAIS

VIRTUAL ENVIRONMENT FOR SIMULATION OF INTELLIGENT MOBILE ROBOTS APPLIED IN FOREST FIRE COMBAT

**Gustavo Pessin¹, Fernando Osório¹, Soraia Musse², Vinícius
Nonnemacher³ e Sandro Souza Ferreira³**

RESUMO

Objetiva-se, neste artigo, detalhar o projeto e o desenvolvimento de um ambiente de simulação que representa uma área florestal e que tem a finalidade de suportar a operação de um sistema multirrobótico fisicamente simulado. A meta é aplicar o sistema no processo de automatização da identificação e combate de incêndios em áreas florestais. O ambiente suporta uma série de características fundamentais para a simulação realística da operação, como terrenos irregulares, processos naturais e restrições físicas na criação e no uso dos robôs móveis. Os primeiros resultados demonstram que esse ambiente de simulação pode ter um papel muito importante no planejamento e na execução de operações reais de robótica móvel em áreas florestais.

Palavras-chave: ambiente de simulação, robôs móveis, simulação realística.

ABSTRACT

The main goal of this article is to describe the design and implementation of a simulation environment that represents a forest area and support a multi-robotic physically simulated operation. This system aim to be applied in the automation process of identification and fire combat in forest areas. The environment provides components for realistic simulation of the operation, as irregular terrains, natural fire simulation processes and physical restrictions in the creation and use of the

¹ PIPCA – Universidade do Vale do Rio dos Sinos (UNISINOS) Av. Unisinos 950 – São Leopoldo – RS – Brasil,

² Faculdade de Informática – PUC-RS Av. Ipiranga, 6681 – Porto Alegre – RS – Brasil.

³ GT JEDi – Universidade do Vale do Rio dos Sinos (UNISINOS) Av. Unisinos 950 – São Leopoldo – RS – Brasil.
{pessin, fosorio, vnonnemacher, sandro.s.ferreira}@gmail.com, soraia.musse@puccrs.br

mobile robots. The first results demonstrated that this simulation environment can become a very important tool for the planning and execution of real mobile robotics operations in forest areas.

Keywords: simulation environment, mobile robots, realistic simulation

INTRODUÇÃO

Com a evolução das pesquisas em robótica, cada vez mais os robôs tornam-se complexos em termos físicos. A grande variedade de estudos em morfologia robótica tem desenvolvido variações de robôs dotados de diversos meios de locomoção (*e.g.* pernas, rodas, esteiras). Paralelamente a esse desenvolvimento, tem-se a evolução constante de uma gama extremamente grande de sensores (*e.g.* sistemas de visão, posicionamento, detecção de obstáculos). O desenvolvimento de algoritmos e de técnicas para coordenarem esses conjuntos físicos em um ambiente dinâmico é um desafio extremamente complexo (GO et al., 2004).

Dotar robôs autônomos de capacidade de raciocínio inteligente e de interação com o meio em que estão inseridos é uma área de pesquisa que atrai a atenção de um grande número de pesquisadores (DUDEK; JENKIN, 2000). Existem diversas áreas em que a habilidade de um único agente autônomo não é suficiente para a realização de uma tarefa, em alguns desses casos, como patrulhamento, vigilância, resgate ou exploração, o mais indicado é a aplicação de sistemas multirrobóticos nos quais robôs autônomos trabalham cooperativamente a fim de cumprir uma missão, podendo existir interação entre os robôs ou não (OSAGIE, 2006).

Um grande sonho da sociedade é a aplicação de sistemas robóticos nas atividades que colocam em risco a vida humana, nas que a atuação de humanos é deficitária ou quando a atuação humana, de certa forma, é ineficiente. A tarefa de monitoração e combate de incêndios em áreas florestais é um caso em que a aplicação de um sistema multirrobótico poderia diminuir, consideravelmente, os prejuízos humanos, materiais e ambientais. Para que seja possível desenvolver uma simulação realística, em que um grupo de robôs autônomos trabalhe cooperativamente, a fim de realizar com sucesso a identificação e o combate de incêndios em áreas florestais, sem intervenção humana, necessita-se de um ambiente que combine diversas funcionalidades.

Nos estudos e pesquisas, constata-se que não existe um ambiente de simulação aberto que combine todas as características requisitadas para esse

trabalho, que são: (i) realismo físico de modelagem robótica; (ii) realismo físico de interação de agentes em terrenos 3D não planos (terreno irregular, como um ambiente natural); (iii) facilidades de comunicação entre os agentes; (iv) capacidade de aplicação de diversos métodos de controle (*e.g.* regras, redes neurais artificiais); (v) capacidade de aplicação de métodos de evolução de estratégias (*e.g.* Algoritmos Genéticos); e (vi) simulação de processos naturais (como a propagação do fogo). Assim, optou-se por desenvolver um ambiente próprio.

Ambientes com terrenos 3D, gerados com a biblioteca de programação OSG (OSG, 2007), com a biblioteca de programação *Demeter* (DEMETTER, 2007) permitem combinar o mapa de elevação com uma determinada distribuição de vegetação, criando, assim, um terreno bastante realista (OSÓRIO et al., 2006). Além disso, o uso da biblioteca de programação ODE (ODE, 2007) permite implementar atributos físicos (*e.g.* massa, atrito, fricção, gravidade, colisão), tornando o sistema ainda mais realístico.

Com relação a incêndios florestais, anualmente, registram-se cerca de 45.000 em florestas da Europa. Entre 1989 e 1993, somente na zona mediterrânea, 2,6 milhões de hectares florestais foram destruídos pelo fogo, o equivalente ao desaparecimento do mapa de um território com a dimensão da Bélgica em cinco anos (CE, 2006). Os incêndios florestais causam diversos tipos de danos humanos, materiais e ambientais. Quanto a prejuízos humanos, por exemplo, na Austrália, em 1983, um incêndio que atingiu cerca de 400.000ha matou 75 pessoas. No Brasil, um incêndio no Paraná, em 1973, provocou 110 mortes (LIF, 2006). A extensão territorial do Brasil, a diversidade de sua cobertura vegetal e o número expressivo de ocorrências de incêndios florestais verificados são fatores que enfatizam a necessidade de um sistema cada vez mais aprimorado e consistente de detecção e combate de incêndios florestais (BATISTA, 2004).

As principais metas relacionadas a esse projeto são: (i) recolher informações sobre dados florestais, tipos de vegetação, topografia, e comportamento de incêndios para criar o ambiente virtual mais realista possível; (ii) simular incêndios em florestas, reproduzindo de forma bastante realista o ambiente e a propagação de incêndios; (iii) pesquisar ferramentas e técnicas reais de combate a incêndios florestais; (iv) implementar a simulação de agentes móveis autônomos colaborativos capazes de formar uma brigada de combate a incêndios; (v) determinar métodos de aprendizado de máquina e suas vantagens para o modelo; e (vi) estudar a robustez das ações dos agentes pela leitura de dados de sensores sujeitos a erros.

Neste artigo, apresenta-se, na seção 2, uma pequena descrição de conceitos de simulação e modelagem, com as ferramentas pesquisadas e as bibliotecas de simulação selecionadas para o desenvolvimento do ambiente. Na seção 3, descrevem-se características importantes de robótica móvel que tenham relação com as necessidades deste trabalho. Na seção 4, detalha-se o desenvolvimento do ambiente e os exemplos de experimentos com a simulação do sistema multirrobótico. Finaliza-se, apresentando a conclusão do trabalho.

SIMULAÇÃO E MODELAGEM

Experimentos em robótica móvel realizam-se de duas formas: diretos em um robô real ou em um robô simulado, em um ambiente virtual realista (PFEIFER; SCHEIER, 1999). Usualmente, experimentos em robótica móvel que utilizam um robô real exigem muito tempo e recursos financeiros. Para que seja possível a implementação física real, o sistema multiagente proposto deve ser projetado, desenvolvido e testado anteriormente em ambientes de simulação realística. A simulação de sistemas robóticos é especialmente necessária para robôs caros, grandes, ou frágeis (GO et al., 2004), pois agiliza o ciclo de desenvolvimento de sistemas de controle robótico, eliminando o desperdício de recursos financeiros e computacionais.

Para que uma simulação seja útil, ela deve capturar características importantes do mundo físico, no qual o termo *importantes* tem relação ao problema em questão (GO et al., 2004). No caso deste trabalho, é fundamental que existam restrições físicas no modelo e possibilidade de trabalho em um terreno irregular, provido de obstáculos. Para o desenvolvimento do trabalho, foram pesquisadas algumas ferramentas de simulação, porém nenhuma mostrou possuir o conjunto completo das características requisitadas. A seguir, apresenta-se uma pequena descrição de cada uma das ferramentas pesquisadas.

No *microsoft robotics studio* (msdn.microsoft.com/robotics) o objetivo é prover um ambiente para desenvolvimento de simulação robótica. Ele permite a simulação em terrenos irregulares e é livre apenas para uso não comercial. O desenvolvimento dos sensores, dos controles inteligentes e da comunicação entre os robôs depende de programação (*e.g.* C#, VB.NET). O *webots* (www.cyberbotics.com) é um simulador de robôs móveis que tem como base a biblioteca de simulação física ODE, inclui modelos prontos de alguns robôs comerciais e modela sensores com capacidade de detecção

de obstáculos e de manipuladores simples. O usuário pode programar cada robô utilizando C/C++, que é um produto comercial. Outros ambientes de simulação estudados foram o *khepera simulator* (diwww.epfl.ch/lami/team/michel/khep-sim), o *mission simulation facility* (ase.arc.nasa.gov/msf), o *JUICE* (www.natew.com/juice) e o *simulator BOB* (www.tu-harburg.de/ti6/mitarbeiter/pst/Sim/Simulator.html).

Como optou-se por desenvolver um ambiente próprio, todas as bibliotecas de programação selecionadas para o desenvolvimento do ambiente e dos agentes robóticos são *software* livre, multiplataforma e em linguagem C/C++. A seguir, apresenta-se uma descrição de cada uma das bibliotecas utilizadas:

Open Dynamics Engine (ODE, 2007): é uma biblioteca desenvolvida para a simulação física de corpos rígidos articulados (SMITH, 2006). Uma estrutura articulada é criada quando corpos rígidos de vários tipos são conectados por algum tipo de articulação, como, por exemplo, um veículo terrestre que tem a conexão de rodas em um chassi. A ODE foi projetada para ser utilizada de modo interativo em simulações de tempo real e é especialmente indicada para a simulação de objetos móveis em ambientes dinâmicos. A ODE não tem como objetivo realizar simulações de outras dinâmicas, além da dinâmica de corpos rígidos (não inclui partículas, roupas, ondas, fluidos, corpos flexíveis ou fraturas). Ela possui contatos rígidos, assim, quando dois corpos colidem não há penetração. O sistema de detecção de colisão é nativo e suporta diversas primitivas (e.g. esfera, caixa, cilindro, plano, raio).

A utilização da ODE no ambiente estudado é fundamental por fornecer restrições físicas, principalmente, na definição da morfologia dos robôs. Do ponto de vista físico, um robô pode ser considerado simplesmente como um conjunto de corpos rígidos conectados (SMITH, 2006). Cada um desses corpos pode interagir com os demais, como um motor faz com que um veículo se movimente, e a atuação da gravidade e da inércia devem estar presentes.

Essas propriedades são tratadas em dois conceitos: os corpos rígidos e as articulações. Com relação aos corpos rígidos, suas propriedades, do ponto de vista da simulação, que mudam com o passar do tempo, são: (a) vetor de posição do ponto de referência do corpo; (b) velocidade linear do ponto de referência do corpo; (c) orientação do corpo; (d) velocidade angular; (e) posição do centro de massa; e (f) matriz de inércia. As articulações (Figura 1) são relacionamentos entre corpos de modo que possam existir posições e orientações que sejam relativas a ambos - esse relacionamento realiza restrições/obrigações de movimento.

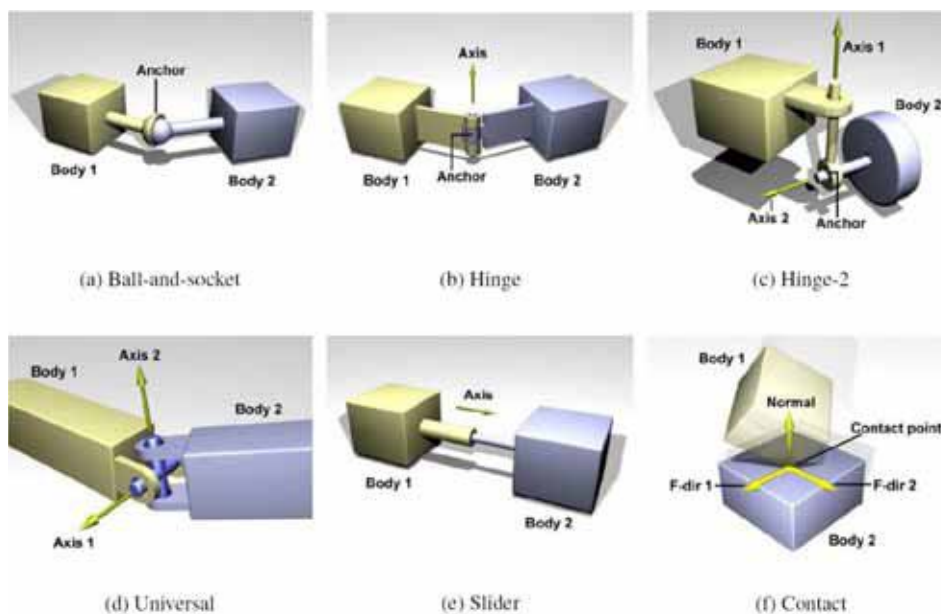


Figura 1 – Articulações disponíveis na ODE (SMITH, 2006).

Open Scene Graph (OSG, 2007): é uma biblioteca para desenvolvimento de aplicações gráficas 3D de alta performance. Baseada no conceito de grafos de cena, provê ao desenvolvedor um ambiente orientado a objeto sobre a *OpenGL* (www.opengl.org), liberando-o da necessidade de implementação e otimização de chamadas gráficas de baixo nível. O grafo de cena permite a representação de objetos em uma cena com uma estrutura que libera a criação de grupos que podem compartilhar propriedades. Assim, pode-se definir uma propriedade comum em um nível hierárquico mais alto e todos os objetos inferiores receberão esta propriedade (BARROS et. al., 2007).

DrawStuff (ODE, 2007): como a ODE é completamente independente de visualizador, iniciar a criação dos corpos e das simulações pode ser uma tarefa bastante árdua, caso não se tenha uma forma simples e fácil de visualizar os objetos. Por esse motivo, a biblioteca *drawstuff* é disponibilizada em conjunto com a ODE. Basicamente, o *drawstuff* é um ambiente de visualização de objetos 3D que tem o propósito de permitir a demonstração visual da ODE, sendo uma biblioteca bastante simples e rápida.

Demeter (DEMETER, 2007): é uma biblioteca desenvolvida para “renderizar” terrenos 3D, ter rápida performance e boa qualidade visual, ou

grandes terrenos em tempo real, sem necessidade de *hardware* especial, ainda, depende da *simple directmedia layer* (www.libsdl.org) para realizar o tratamento das texturas do terreno e da *geospatial data abstraction library* (www.gdal.org) para carregar arquivos de elevação.

ROBÓTICA MÓVEL

Um robô móvel é um dispositivo mecânico, montado sobre uma base não fixa, que age sob o controle de um sistema computacional equipado com sensores e atuadores que lhe permitem interagir com o ambiente (MARCHI, 2001; BEKEY, 2005). A interação com o ambiente acontece através de ciclos de percepção-ação, os quais consistem em três passos fundamentais: (i) obtenção de informação através de sensores; (ii) processamento das informações para seleção de ação e (iii) execução da ação por meio do acionamento dos atuadores. Esse conjunto de operações, em uma análise superficial, pode parecer simples, porém o controle robusto de sistemas robóticos tem complicações físicas, mecânicas, eletrônicas e computacionais que tornam a criação de um conjunto de regras de controle uma tarefa árdua e sujeita a erros.

Sensores, os mecanismos de percepção de um robô, realizam medições físicas (e.g. contato, distância, orientação, temperatura) e provêm sinais ou dados crus que precisam ser interpretados pelo “cérebro” do robô. A interpretação desses sinais deve ser a única maneira de um robô autônomo entender o ambiente que o cerca para realizar as mudanças de ação necessárias (JPL/NASA, 2007).

Atuadores, os mecanismos de ação de um robô (e.g. motores, pistões, braços manipuladores), são controlados por circuitos eletrônicos que recebem valores de ação, os quais devem ser calculados pelo “cérebro” do robô e devem estar de acordo com especificações dos fabricantes (JPL/NASA, 2007).

IMPLEMENTAÇÃO DO PROTÓTIPO

O ambiente simulado conta com um grupo de n robôs de comportamento reativo (reação sensorial-motora) e com um agente deliberativo (mecanismo de planejamento), assim, outra característica do ambiente é permitir a definição de tipos de comportamentos diferentes para cada tipo de agente. Em se considerando arquiteturas de sistemas multiagentes, o ambiente desenvolvido possui controle centralizado de planejamento e controle distribuído de ações (local em cada robô móvel).

Usa-se o ambiente para simular a seguinte operação: um agente monitor (satélite) monitora o terreno da área florestal e, ao identificar uma área com foco de incêndio, envia uma mensagem para o agente líder. Essa mensagem contém a

posição aproximada (x,y) do incêndio (simulando uma posição UTM) e a densidade da vegetação na área. O agente líder é o agente responsável pela definição das posições de atuação dos robôs bombeiros no combate ao incêndio. Após receber o *aviso de incêndio* do agente monitor, o agente líder envia para todos os agentes de combate uma mensagem informando *início de incêndio na posição (x,y)* e recebe a distância vetorial d (Equação 1) de cada agente de combate em relação ao incêndio.

$$d(P,Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Após o agente líder receber as respostas dos agentes de combate, envia mensagens de solicitação de formação para atuação da seguinte forma: se não existe vento, solicita os 8 agentes mais próximos ao incêndio em formação circular equidistante com um raio predeterminado; se existe vento, solicita os 4 agentes mais próximos ao incêndio em formação semicircular (ferradura) no sentido contrário ao do vento, com raio definido, de acordo com a intensidade do vento. Essa formação é baseada em regras e pode ser visualizada nas figuras 7 e 8. A regra pré-programada que define a posição final de cada robô na composição da formação da equipe (quando circular equidistante) pode ser vista nas Equações 2 e 3.

$$x_f = x_a + r \times \cos\left(i \times \frac{360}{q} \times \frac{\pi}{180}\right) \quad (2)$$

$$y_f = y_a + r \times \text{sen}\left(i \times \frac{360}{q} \times \frac{\pi}{180}\right) \quad (3)$$

Considere x_f e y_f como as coordenadas da posição final do agente, x_a e y_a como as coordenadas da posição central do incêndio, r como o raio de atuação, q a quantidade total de agentes e i o índice do agente. As posições são negociadas e confirmadas com a comunicação entre os agentes da equipe. Eles possuem quatro sensores de temperatura que servem como alerta para, quando a temperatura de um deles exceder o máximo especificado, o agente se deslocar no sentido do sensor com a menor temperatura e solicitar ao agente líder a atualização da formação do time. O comportamento de combate é reativo, deslocando-se à posição de seu objetivo específico e se desviando de obstáculos e de áreas densas de vegetação. O método de combate de incêndio simulado é o método indireto (LIF, 2006) e os agentes de combate simulados são motoniveladores que têm como finalidade cercar o foco de

incêndio e criar um aceiro (área livre de vegetação onde o fogo se extingue pela falta de combustível). Essa operação pode ser entendida com as figuras 7 e 8. O desenvolvimento dos objetivos mais específicos do trabalho são detalhados a seguir:

SIMULAÇÃO DE PROCESSOS NATURAIS

A fim de melhor entender como proceder no combate a incêndios florestais e de planejar as estratégias a serem implementadas nos agentes autônomos, foi realizado um estudo sobre as técnicas reais de operação, o qual teve como base os trabalhos de Antunes (2006), Batista (2004), CPTEC/INPE (2006) e LIF (2006). Para a implementação da propagação do fogo, obteve-se de (KOPROSKI, 2005) medições reais de velocidades. O estudo dos modelos de florestas e resíduos florestais é de grande importância para o aprimoramento dos modelos de simulação a serem implementados em ambientes virtuais (PESSIN et al., 2007a). A criação dos mapas teve como base cartas topográficas (Figura 2(a)) e o mapa de modelo de combustíveis florestais do Ministério da Agricultura do Brasil (Figura 2(b)).

Para a simulação da vegetação e a correta propagação do incêndio, existe uma matriz oculta sob o terreno. Ela possui, para cada área do terreno, o tipo de vegetação presente. O vento, sua intensidade, e sua orientação podem ser gerados aleatoriamente ou configurados a partir de dados parametrizados pelo usuário. O tempo de permanência do fogo em uma área tem relação direta com o tipo de vegetação presente e se comporta baseado nos valores de tipo de vegetação, inclinação do terreno, intensidade e orientação do vento.

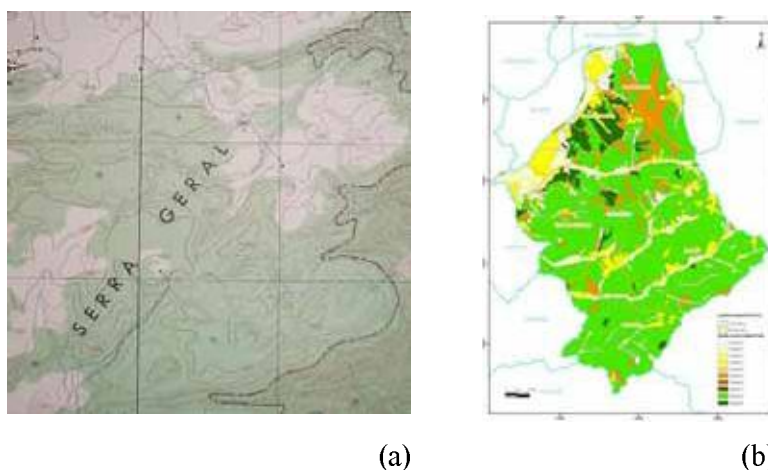


Figura 2 – Fragmento de carta topográfica representando um quadrante de 6km x 6km e (b) Mapa de modelos de combustíveis (MAB, 2006).

FACILIDADES DE COMUNICAÇÃO ENTRE OS AGENTES

A comunicação permite aos agentes, em um ambiente multiagente, a troca de informações que servem de base para coordenar suas ações e realizar cooperação. Desenvolveu-se o ambiente com possibilidade de comunicação baseado em quadro-negro (*blackboard*) (REZENDE, 2003). Existem casos de comunicação um para um e de um para todos. A fila usada como função de quadro-negro armazena as seguintes informações: indicador de remetente, indicador de destinatário, *timestamp* e tipo da mensagem (*e.g.* aviso de incêndio, aviso de fim de incêndio, negociação de times, posição do incêndio).

CAPACIDADE DE APLICAÇÃO DE DIVERSOS MÉTODOS DE CONTROLE

O ambiente desenvolvido permite que cada agente tenha seu próprio método de controle. Por exemplo, os robôs móveis foram programados de duas formas: inicialmente, foi aplicado um conjunto de regras de controle motor e, posteriormente, uma rede neural artificial que permite aos robôs móveis a realização de uma navegação autônoma entre dois pontos, evitando, desse modo, colisão com obstáculos. O desenvolvimento do controle utilizando redes neurais artificiais em um ambiente bidimensional pode ser visto em (PESSIN et al. 2007c) e, em um ambiente tridimensional, com simulação baseada em física (utilizando ODE e *drawstuff*) pode ser visto em (PESSIN et al., 2007b). A figura 3 apresenta o resultado de uma simulação em que um grupo de veículos realiza navegação e desvio satisfatórios.

CAPACIDADE DE APLICAÇÃO DE MÉTODOS DE EVOLUÇÃO DE ESTRATÉGIAS

Criou-se, para este trabalho, uma unidade monitora (agente deliberativo) que possui regras de atuação, explicadas na seção 4. Outra forma de atuação dessa unidade monitora consiste na utilização de algoritmos genéticos para evoluir estratégias de posicionamento dos robôs móveis (agentes reativos). A evolução de estratégias é implementada com a biblioteca de programação de algoritmos genéticos GALib (lancet.mit.edu/ga) e o ambiente permite que, durante a evolução, a saída gráfica seja desabilitada, possibilitando a aceleração das simulações, até atingir o valor esperado da função objetivo (*fitness*).

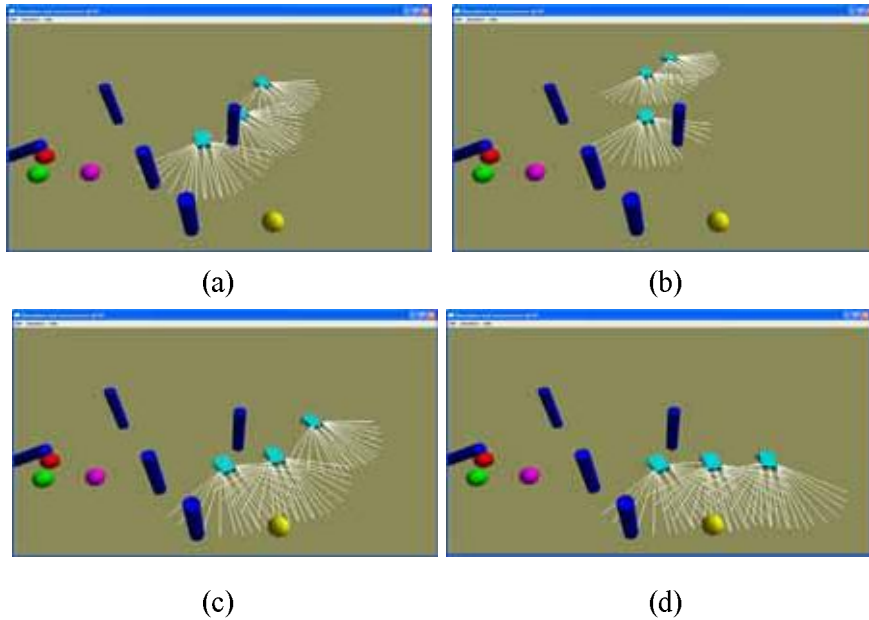


Figura 3 – Sequências de uma simulação com navegação e desvio satisfatórios.

CAPACIDADE DE APLICAÇÃO DE MÉTODOS DE EVOLUÇÃO DE ESTRATÉGIAS

Criou-se, para este trabalho, uma unidade monitora (agente deliberativo) que possui regras de atuação, explicadas na seção 4. Outra forma de atuação dessa unidade monitora consiste na utilização de algoritmos genéticos para evoluir estratégias de posicionamento dos robôs móveis (agentes reativos). A evolução de estratégias é implementada com a biblioteca de programação de algoritmos genéticos GALib (lancet.mit.edu/ga) e o ambiente permite que, durante a evolução, a saída gráfica seja desabilitada, possibilitando a aceleração das simulações, até atingir o valor esperado da função objetivo (*fitness*).

O *fitness* desenvolvido acumula os seguintes valores finais de cada simulação: (i) total de área queimada: busca minimizar a área queimada; (ii) total de área com aceiro: tende a enfraquecer a área de trabalho dos robôs; e (iii) erro médio absoluto: visa a diminuição da diferença entre a média geral de aceiros úteis em relação ao aceiro útil de cada indivíduo, assim, o tamanho das áreas de trabalho tende a se equalizar.

O genoma desenvolvido para as simulações iniciais pode ser visto na tabela 1. Nele encontram-se as informações de todo o grupo dos agentes envolvidos. Desse modo, o tamanho do genoma depende da quantidade de agentes no sistema. Realizaram-se simulações, considerando a existência de quatro agentes de combate e uma das simulações, com resultado satisfatório, os quais apresentam-se na figura 4.

Tabela 1 - Genoma dos indivíduos.

Gene	Função	Valor Mín.	Valor Máx.
0	Posição inicial do agente 0	0,000º	360,000º
1	Posição final do agente 0 e inicial do agente 1	0,000º	360,000º
2	Posição final do agente 1 e inicial do agente 2	0,000º	360,000º
3	Posição final do agente 2 e inicial do agente 3	0,000º	360,000º
4	Posição final do agente 3	0,000º	360,000º
5	Raio inicial do agente 0	10,000m	100,000m
6	Raio final do agente 0 e inicial do agente 1	10,000m	100,000m
7	Raio final do agente 1 e inicial do agente 2	10,000m	100,000m
8	Raio final do agente 2 e inicial do agente 3	10,000m	100,000m
9	Raio final do agente 3	10,000m	100,000m



Figura 4 - Resultado da simulação com 150 indivíduos e 700 gerações. A área preta define a área queimada, as linhas azuis são os trajetos dos robôs e a linha bege representa a área de criação de aço.

O genoma resultante da simulação de 700 gerações com 150 indivíduos apresentou, na última geração, os seguintes valores absolutos de *fitness*: média=3.076,88; maior=3.099,00; menor=3.040,00; desvio padrão=10,56. O cromossomo final pode ser visto na tabela 2.

Tabela 2 - Indivíduo resultante da simulação vista na figura 4.

225,72°	199,26°	174,12°	155,20°	136,38°	27,35m	30,45m	33,94m	38,08m	35,96m
---------	---------	---------	---------	---------	--------	--------	--------	--------	--------

REALISMO FÍSICO DE MODELAGEM ROBÓTICA

Iniciou-se o desenvolvimento focando o realismo de simulação física, por meio das bibliotecas de programação ODE e *drawstuff*. A partir das bibliotecas, pode-se definir a morfologia do veículo móvel dos sensores e dos atuadores (figura 5). A figura 6 permite a visualização de um grupo de 3 veículos interagindo com o ambiente.

Um dos sensores simulados é o sensor de posicionamento GPS. Em levantamento de campo realizado com um GPS *Garmin Etrex* (www.garmin.com), obteve-se um erro médio de 18,6 metros. Considerando que cada robô possui seu próprio GPS, o tratamento desse erro é crucial na criação dos aceiros (PESSIN et al., 2007a). Essa informação foi tratada de duas maneiras, a primeira faz com que o erro médio desse sensor, durante o deslocamento, seja usado somado à distância de criação do aceiro e também ao final da área de criação, como se mostra nas figuras 7(b) e 8.

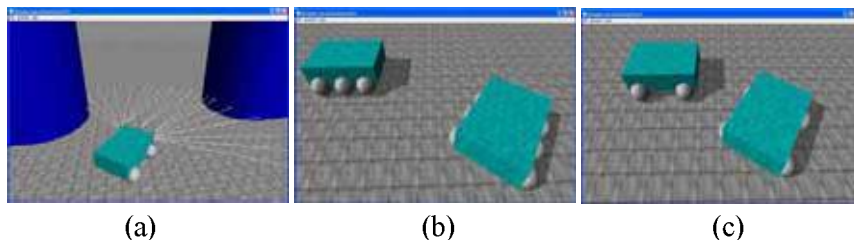
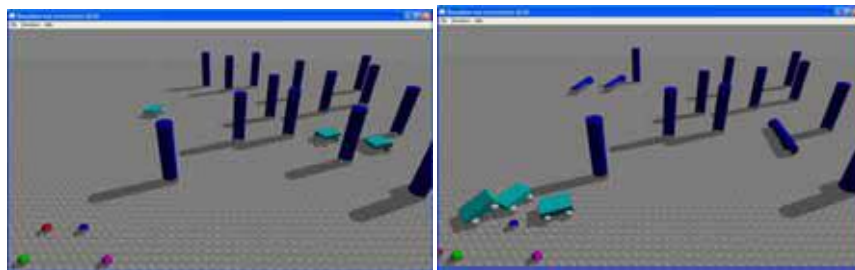


Figura 5 - Simulações de: (a) Sensores de distância, (b) Cinemática de esteira e (c) Cinemática de automóvel.

Na figura 7(a), apresenta-se o esquema de movimentação com “posicionamento perfeito”, o que não ocorre em uma situação real. Devido ao erro de posicionamento do GPS, não é possível se estabelecer uma rota que se encaixe perfeitamente como a apresentada na figura 7(a). Desse modo, a

preparação do aceiro (semicírculo indicado na figura) não será executada de forma correta. A figura 7(b) demonstra as correções adicionadas ao algoritmo e à obtenção de uma trajetória que permita a criação um aceiro incluindo o erro do GPS (e_{GPS}) no modelo de deslocamento dos agentes. Nas figuras 8(a) e 8(b), encontram-se os resultados obtidos em uma das simulações realizadas com o protótipo 2D.

Dada a existência de restrições físicas, a única maneira de controlar o veículo da figura 5(c) é com a aplicação de forças em seus dois motores simulados, um motor angular (para o giro do volante) e um motor linear (para o torque).

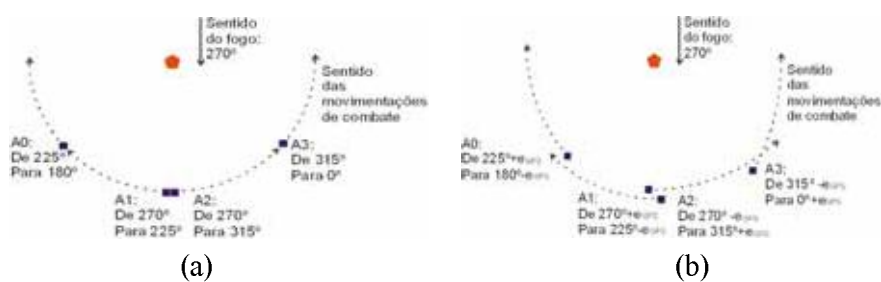


(a)

(b)

Figura 6 - Grupo de 3 veículos (baseado em regras) em deslocamento até um objetivo: (a) Início e (b) chegada ao objetivo sem controle de colisão, cilindros derrubados e dois veículos “acidentados”.

O veículo da Figura 5(b) foi simulado com cada lateral possuindo três rodas e um motor linear, assim, apresenta dinâmica de veículos com esteira e possui controle de giro e direcionamento baseado na aplicação de forças independentes em cada motor linear.



(a)

(b)

Figura 7. Exemplo de formação de combate quando o sentido do vento é de 270° e existem 4 agentes de combate: (a) não considera erro no GPS e (b) considerando erro no GPS, apresentando limites propositalmente redundantes

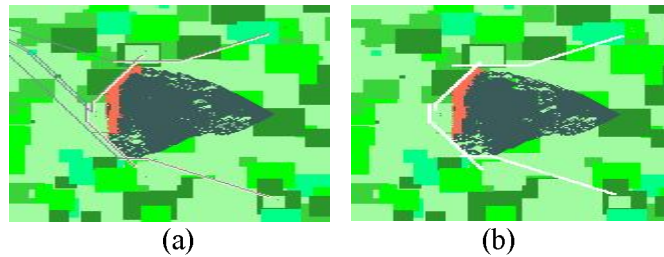


Figura 8. Simulação com o protótipo 2D, considerando o erro de GPS: (a) trajetórias dos robôs-bombeiros e (b) criação do aceiro. Ambas apresentam limites propositalmente redundantes.

REALISMO FÍSICO DE INTERAÇÃO DE AGENTES EM TERRENOS 3D IRREGULARES

A base do realismo de interação consiste no uso integrado das bibliotecas de programação ODE, OSG e *Demeter*. Esse conjunto permite que os robôs simulados fisicamente respeitem questões como gravidade, inércia e atrito.

Manter, por exemplo, uma força f constante nos motores lineares (torque) proporcionará a um veículo uma velocidade v em regiões planas, em regiões de declive terá v maior e, em aclives, v menor. Na figura 9, mostra-se o ambiente desenvolvido. O

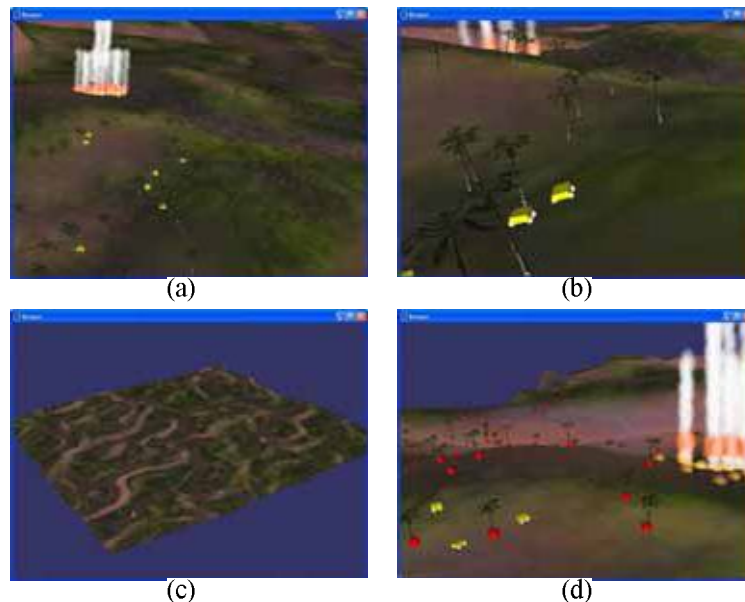


Figura 9 - Ambiente completo, (a) um veículo preso em uma árvore, (b) vista do grupo em deslocamento até o incêndio, (c) vista superior e (d) visualização do ambiente com a representação dos cilindros físicos junto às árvores.

demeter, com a SDL e o GDAL, são os responsáveis pela criação do terreno 3D. A OSG permite a leitura do terreno e a leitura dos modelos de árvores e do fogo. Cada árvore existe como um modelo OSG e como um cilindro ODE, desse modo, o veículo da figura 9(a) está, na verdade, colidindo com um cilindro (não visualizado). O ambiente permite a habilitação ou não da exibição dos objetos físicos ao invés de seu modelo gráfico, nesse exemplo, uma árvore.

CONCLUSÃO

O objetivo do estudo foi detalhar o projeto e o desenvolvimento de um ambiente de simulação que representa uma área florestal, a partir da finalidade de suportar a operação de um sistema multirrobótico. Esse sistema será aplicado no processo de automatização da identificação e combate de incêndios em áreas florestais. O ambiente da análise suporta uma série de características fundamentais para a simulação realística da operação, como terrenos irregulares, processos naturais e restrições físicas na criação e no uso dos robôs móveis que planejam suas ações baseados no uso de dados de seus sensores (*e.g.* GPS, odômetro, sensor de temperatura, bússola).

O terreno simula diversos aspectos naturais (tipos de vegetação e topografia) e respeita questões, como taxa de propagação do fogo, baseada na intensidade e orientação do vento, vegetação e topografia. A integração das bibliotecas de programação permitiu o desenvolvimento de um ambiente completamente livre, em C++, integrando as características requisitadas para a operação multirrobótica em áreas florestais. Os primeiros resultados demonstraram que esse ambiente de simulação pode ter um papel importante no planejamento e execução de operações reais de robótica móvel em áreas florestais, além de poder ser utilizado em outras aplicações, como jogos ou aplicações de realidade virtual.

AGRADECIMENTOS

Agradecemos ao CNPq, Capes, Unibic (Unisinos), FAPERGS e a FINEP (Projeto Rede Brasileira de Visualização) por apoiarem este trabalho.

REFERÊNCIAS

ANTUNES, M. A. H. **Uso de satélites para detecção de queimadas e para avaliação do risco de fogo**, Ação Ambiental, 12:24-27, 2000.

BARROS, L. M., GONZAGA, L., RAPOSO, A. B. **Open Scene Graph**: conceitos básicos e aplicações em realidade virtual, Tutorial. In: IX SYMPOSIUM ON VIRTUAL AND AUGMENTED REALITY (SVR), 2007.

BATISTA, A.C. Detecção de incêndios florestais por satélite, **Revista Floresta**, Curitiba, Paraná, n. 34, Mai/Ago, 237-241, 2004.

BEKEY, G. A. **Autonomous Robots**: From Biological Inspiration to Implementation and Control. Cambridge, Massachusetts, USA: The MIT Press, 2005.

CE - Comissão Européia. **O que faz a Europa? Incêndios florestais**. Disponível em: <<http://ec.europa.eu/research/leaflets/disasters/pt/forest.html>>. Acesso em: set., 2006.

CPTEC/INPE. **Centro de previsão do tempo e estudos climáticos - Instituto nacional de pesquisas espaciais**. Disponível em <<http://www.cptec.inpe.br/queimadas>>. Acesso em: out., 2006.

DEMETER, **Demeter Terrain Engine**. Disponível em: <<http://www.tbgssoftware.com/>>. Acesso em: jun., 2007

DUDEK, G.; JENKIN, M. **Computational Principles of Mobile Robotics**, Cambridge, London, UK: The MIT Press, p. 280 , 2000.

GO, J. et al. **Accurate and flexible simulation for dynamic, vision-centric robots**. In: INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS, 2004.

KOPROSKI, L.P. **O fogo e seus efeitos sobre a herpeto e a mastofauna terrestre no parque nacional de Ilha Grande**, 2005, Dissertação (Mestrado) - UFPR, 2005.

JPL/NASA. **Jet Propulsion Laboratory / NASA**.. Disponível em: <<http://www.robotics.jpl.nasa.gov>>. Acesso em: maio, 2007.

LIF - Laboratório de Incêndios Florestais. **Pesquisas e projetos em prevenção e combate de incêndios florestais**. Disponível em <<http://www.floresta.ufpr.br/~firelab>>. Acesso em: set., 2006.

MARCHI, J. **Navegação de robôs móveis autônomos: Estudo e implementação de abordagens**. Dissertação (Mestrado). Universidade Federal de Santa Catarina, 2001.

ODE, **Open Dynamics Engine**, Disponível em: <www.ode.org>. Acesso em: jun., 2007.

OSAGIE, P. **Distributed Control for Networked Autonomous Vehicles**, Dissertação (Mestrado), KTH CSC, Royal Institute of Technology, Sweden, 2006.

OSG, **Open Scene Graph**, OSG Community. Disponível em: <<http://www.openscenegraph.com>>. Acesso em: jun., 2007.

OSÓRIO, F. S., et al. **Increasing Reality in Virtual Reality Applications through Physical and Behavioural Simulation**, Research in Interactive Design - Proceedings of the Virtual Concept Conference (Summer School). Springer Verlag, v.2, p. 1-45, 2006.

PESSIN, G., OSÓRIO, F., MUSSE, S., NONNEMMACHER, V. FERREIRA, S. **Simulação Virtual de Agentes Autônomos para a Identificação e Controle de Incêndios em Reservas Naturais**. In: IX SYMPOSIUM ON VIRTUAL AND AUGMENTED REALITY (SVR), v.1, p. 236-245, 2007.

PESSIN, G., OSÓRIO, F., MUSSE, S., NONNEMMACHER, V. FERREIRA, S. **Utilizando Redes Neurais Artificiais no Controle de Robôs Móveis Aplicados ao Combate de Incêndios Florestais**. In: XVI Seminário de Computação, FURB, Blumenau, SC, 2007.

PESSIN, G., OSÓRIO, F., MUSSE, S., NONNEMMACHER, V. FERREIRA, S. **Utilizando Agentes Autônomos com Aprendizado para a Identificação e Combate de Incêndios em Áreas Florestais**. In: VII SIMPÓSIO DE INFORMÁTICA DO PLANALTO MÉDI, Passo Fundo, RS, 2007.

PFEIFER, R.; SCHEIER, C. **Understanding Intelligence**. Cambridge, Massachusetts, USA: The MIT Press, p. 697, 1999.

REZENDE, S.O. **Sistemas inteligentes: fundamentos e aplicações**. São Paulo: Manole, 2003.

SMITH, R. **Open Dynamics Engine v0.5 User Guide**. 2006