

SISTEMA ROBÓTICO MULTIAGENTE PARA TRANSPORTE COLABORATIVO

JhielsonMontino Pimentel¹, Fabiana Cristina Bertoni²

¹Bolsista PROBIC, Graduando de Engenharia de Computação, Universidade Estadual de Feira de Santana (UEFS), email:jhielson@gmail.com

²Orientador, Departamento de Exatas, Universidade Estadual de Feira de Santana (UEFS), email: fcbertoni@gmail.com

PALAVRA-CHAVE:Multiagentes, Quadtree, Dijkstra.

INTRODUÇÃO

A robótica móvel consiste numa subcategoria que pesquisa e desenvolve agentes autônomos capazes de extrair informações do ambiente e utilizar esse conhecimento para se deslocarem com segurança de modo significativo e intencional, atuando e executando tarefas [Pio et al. 2006]. Uma das áreas de aplicação da robótica móvel que tem despertado interesse nas comunidades científicas está relacionada ao controle e planejamento de caminhos a serem percorridos e a execução de tarefas em cenários estruturados e não estruturados.

Antes de investir tempo e dinheiro em um robô móvel real, os pesquisadores podem explorar uma hipótese ou um novo algoritmo de planejamento de caminho através de ambientes virtuais de simulação. Sendo assim, os simuladores se tornam ferramentas poderosas para agilizar o ciclo de desenvolvimento de sistemas de controle robóticos eliminando desperdício de recursos, tanto financeiros como computacionais. Também pode evitar danos através da detecção de possíveis riscos de colisões do robô com obstáculos ou até mesmo com partes da própria peça manipulada que podem se encontrar no ambiente de trabalho, o qual caracteriza todos os pontos que o robô consegue alcançar a partir da sua base fixa) [Heinen et al. 2002].

Durante o processo de controle e planejamento de caminhos, o simulador deve realizar cálculos que definem a posição relativa dos objetos ao redor do robô e também deve determinar as rotas a serem seguidas pelos agentes. Assim, a aquisição ou construção de um mapa local torna-se fundamental. O mapa descreve de forma realística o ambiente em que os agentes robóticos serão posteriormente inseridos. Para o simulador, o mapa consiste em uma estrutura de dados de simples interpretação ao sistema, podendo ser um grafo conectando pontos de livre acesso no ambiente. [Dias et al. 2010].

Após o mapeamento e localização dos agentes, o grafo gerado deve ser interpretado para a identificação da melhor rota a ser seguida pelos agentes. Uma das formas de análise consiste na aplicação do algoritmo Dijkstra. Este constitui uma técnica com o objetivo de obter o menor caminho entre um dado vértice fixo e todos os demais vértices do grafo. [Sampaio et al. 1998].

Neste contexto, este artigo visa apresentar o projeto de desenvolvimento de um simulador de multiagentes robóticos com a tarefa de carregar objetos em um cenário estruturado. Cada agente deve identificar o objeto a ser manipulado, se posicionar adequadamente, em conjunto, transportá-lo até um local pré-determinado. O objetivo é que os dois agentes possam empurrar um objeto pelo labirinto de forma síncrona, realizando, quando necessário, movimentos de rotação no objeto em torno do eixo perpendicular ao solo. O mapeamento irá orientar os agentes, indicando o quanto e quando devem empurrar o objeto e quando devem realizar a rotação e de quantos graus.

PROJETO

O sistema desenvolvido consiste num simulador capaz de definir rotas e estratégias para o transporte colaborativo multiagente de um objeto. A linguagem de programação escolhida para implementação do projeto foi JAVA, sendo os pacotes AWT (*java.awt-Abstract Windowing Toolkit*) e SWING (*javax.swing - Swing Components*) usados na etapa de interface gráfica com o usuário (GUI).

A escolha dos algoritmos de mapeamento, Quadtree, e planejamento, Dijkstra, foi devido ao já uso em outros projetos e pela garantia de um bom resultado em tempo de resposta adequado ao simulador.

MAPA

O mapa virtual do ambiente foi projetado a partir das características de um labirinto real de dimensões 220x270 cm² fornecido pela Universidade Estadual de Feira de Santana ao projeto. Além do labirinto, a universidade forneceu acesso ao laboratório de pesquisa para a execução das atividades. Para a realização precisa dos cálculos de deslocamento e da orientação dos agentes, foi necessário projetar todo o ambiente virtual proporcional ao real.

SIMULADOR

O simulador, como pode ser visto na figura 01, é uma interface composta por um frame de controle, contendo as principais funcionalidades do sistema, e por um frame de visualização, no qual se observa as etapas do processo do transporte colaborativo. O usuário pode inserir novos elementos no mapa virtual, definir as posições inicial e final do objeto e gerar as rotas entre os agentes e o objeto. Caso queira realizar alguma modificação, o usuário pode reiniciar o sistema voltando para a configuração inicial. No frame de visualização é representado um mapa métrico com as dimensões equivalentes ao ambiente real, contendo as paredes de um labirinto por onde os agentes irão percorrer.

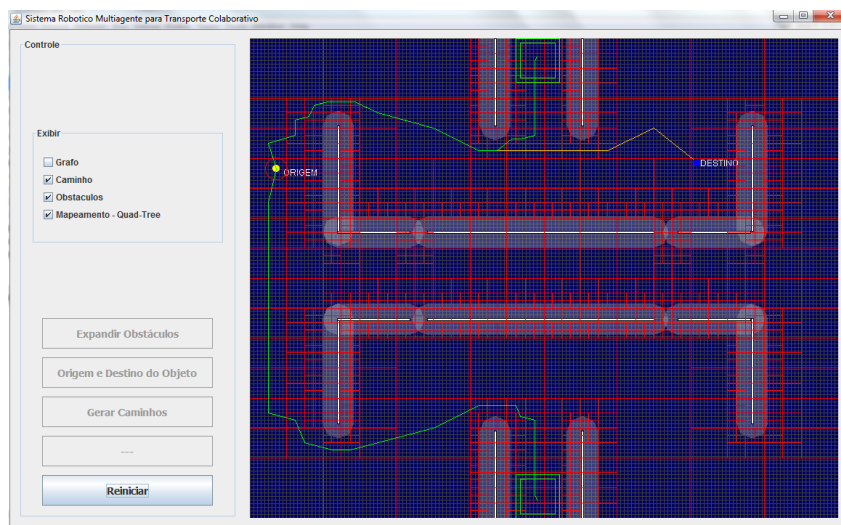


Figura 1 Simulador

Na opção de inserção dos obstáculos, o usuário possui três padrões de tamanho que podem ser inseridos no mapa com a orientação que o usuário desejar. Como a modificação do ambiente ocorre antes do processo de mapeamento, o mapa continua com a característica de estruturado ou estático.

A etapa seguinte à inserção dos obstáculos consiste na expansão dos elementos (paredes e obstáculos). Este processo é usual em ambientes de simulação de robôs, pois torna mais simples as operações de deslocamento dos agentes. Cada agente passa a ser representado por um ponto no mapa métrico, sendo este definido através do centro de massa ou centro do eixo das rodas. Para que essa representação ocorra, é necessária a expansão dos elementos do mapa de acordo com o maior raio de largura dos agentes. Como os agentes trabalhados nesse projeto possuem características semelhantes em relação às dimensões e componentes, o raio será único. Assim, quando um ponto ultrapassar uma das regiões de expansão, significa que o agente colidiu com um elemento do mapa.

Antes de iniciar o processo de gerar rotas, o usuário deve indicar no mapa a posição inicial e final do objeto. A partir desses pontos, o sistema iniciará o processo de mapeamento e planejamento de rotas.

MAPEAMENTO

O mapa irá passar por uma etapa de simplificação da estrutura para o processamento das rotas. O algoritmo Quadtree é então executado, analisando os espaços livres. Inicialmente, o algoritmo irá caracterizar o mapa como uma única célula contendo todos os elementos do ambiente. Em seguida, passa a uma estrutura de repetição no qual o critério de parada é definido como sendo o tamanho mínimo das novas células geradas.

No laço, a cada repetição, são analisadas as células, as classificando em livre, ocupada, ou parcialmente ocupada. Para as células ocupadas, nada mais será feito. Quanto às células livres e parcialmente ocupadas, para cada uma será realizada uma subdivisão em quatro novas células de dimensões equivalentes. Ao término da análise de todas as células não ocupadas, o laço é reiniciado.

Este algoritmo gera uma árvore Quadtree como estrutura de dados, na qual cada nó representa uma célula que pode possuir quatro filhos representando a subdivisão realizada. Ou seja, as folhas irão representar as células livres do mapa ou células com as dimensões mínimas. A partir dessa árvore e das características métricas do labirinto, o sistema gera um grafo fundamental para a análise das rotas.

PLANEJAMENTO

Ao término da etapa de mapeamento, o simulador, com o grafo já formado, inicia o procedimento de planejamento do caminho. Para cada agente, o algoritmo de Dijkstra é executado

tendo como ponto inicial a própria posição no mapa. O mesmo procedimento é realizado tanto para os agentes quanto para o objeto. Assim, os caminhos são gerados como pode ser observado na figura abaixo.

O algoritmo Dijkstra desenvolvido foi baseado nos seguintes passos:

Na primeira etapa são atribuídos valores iniciais aos vértices. Assim, cada vértice terá como atributos um valor infinito que representa a distância dele ao vértice inicial (custo mínimo) e um valor nulo que indica a não existência de conexão com outro vértice. Ao término do laço, é atribuído o valor zero ao custo mínimo do vértice inicial, por não existir distância entre o ponto inicial e ele mesmo. Em seguida, o algoritmo estrutura os vértices em dois vetores: os que já possuem custo de menor caminho (S) e os restantes (Q). O vetor que corresponde aos vértices restantes deve ser ordenado em ordem crescente de acordo com a variável de custo mínimo. Enfim, o algoritmo entra num laço que retira o primeiro elemento do vetor Q. Inicialmente, este vértice deve ser o inicial (s) com o custo mínimo igual a zero. Diretamente, o vértice (u) é inserido no vetor S para em seguida iniciar a análise. Para cada vértice adjacente (v) de u é verificado o peso e o custo mínimo. Assim, caso o custo mínimo de v seja maior que o somatório do peso de u para v com o custo mínimo de u, então um novo custo mínimo será atribuído a v com conexão em u. Essa condicional permite que o caminho de menor custo seja construído durante a estrutura de repetição, estabelecendo as conexões.

CONCLUSÃO

Este artigo apresentou o projeto de desenvolvimento de um simulador de multiagentes robóticos com a tarefa de carregar objetos em um cenário estruturado. Cada agente deve identificar o objeto a ser manipulado, se posicionar adequadamente e, em conjunto, transportá-lo até um local pré-determinado. O projeto encontra-se em sua fase de adaptação, pois durante o desenvolvimento do transporte colaborativo foi observada a elevada complexidade da implementação do sistema de sensoriamento dos agentes. Nesse ponto, optou-se pela utilização do simulador MobileSimRobot, bastante utilizado por pesquisadores, em conjunto com o sistema já desenvolvido nesse projeto de mapeamento e planejamento.

Também estão sendo realizados diversos testes no simulador, envolvendo as etapas de mapeamento e planejamento, com o intuito de detectar e solucionar possíveis falhas, evitando que no ambiente real os agentes sofram danos e que se economize tempo, garantindo uma maior segurança, confiabilidade e robustez ao sistema.

REFERÊNCIAS

- DIAS, C., HOUNSELL, M. d. S., ROSSO Jr., R. S. U., et al. (2010). Análise de Colaboratividade Usando Telerobótica. Conferência Internacional de Educação em Engenharia e Tecnologia, 1:538–542.
- HEINEN, F. J. et al. (2002). Sistema de Controle Híbrido para Robôs Móveis Autônomos. Dissertação de Mestrado.

PIO, J. L. S., CASTRO, T. H. C., CASTRO JÚNIOR, A. N., et al. (2006). A robótica móvel como instrumento de apoio a aprendizagem de computação. XVII Simpósio Brasileiro de Informática na Educação, 1:197–206.

SAMPAIO, R. M., YANASSE, H. H., et al. (1998). Estudo e Implementação de Algoritmos de Roteamento sobre Grafos em um Sistema de Informações Geográficas. Selvatici, A. H. P. et al. (2009). Construção de Mapas de Objetos para Navegação de Robôs. Tese de Doutorado.