

Relatório de projeto de iniciação à investigação científica

13/9/2016

João Gonçalo Veloso, 77083, 2º ano de MIEM

Em colaboração com:

Tiago Simões, 5º ano de MIEM

Prof. Dr. Vítor Santos

Introdução

Este projeto de iniciação à investigação científica esteve inserido no âmbito da dissertação do MIEM do colega Tiago Simões, com a finalidade de complementá-la.

O projeto foi desenvolvido ao longo do 2º semestre do ano letivo 2015/16 e tinha como objetivo o “Estudo da atividade cooperativa entre manipuladores industriais e humanos usando visão 3D”.

O trabalho previu a integração de ferramentas de perceção e manipulação usando um robô FANUC e uma câmara Kinect. A opção escolhida para implementar o objetivo do trabalho foi um jogo de tabuleiro entre o sistema robotizado e um ser humano. O sistema integra um manipulador, uma câmara 3D, software de interface e perceção, e o software de estratégia e planeamento de jogo “inteligente”. De modo a comunicar com o sistema este terá de assumir a forma uma biblioteca de funções.

Portanto, o objetivo deste projeto de iniciação à investigação foi desenvolver um programa que implementasse a estratégia de um jogo de damas contra um adversário.

Desenvolvimento

Uma vez que o tempo disponível e os meus conhecimentos de programação não eram suficientes para criar o *software* de raiz, comecei por realizar diversas pesquisas sobre o tema. As pesquisas centraram-se principalmente na palavra-chave “checkers” (damas em inglês) e nas linguagens de programação C, C++, Python e Java, pois estas linguagens de programação eram dominadas pelos membros da equipa.

Da pesquisa concluiu-se que o programa mais adequado era o CheckerBoard [2]. Trata-se de um jogo com uma interface gráfica, executado em Windows e escrito em C e C++. Com acesso ao código base foi possível retirar toda a interface gráfica e manter apenas a sua funcionalidade de base, de modo a desempenhar o papel de biblioteca de funções em comunicação com o sistema ROS (Robot Operating System), software usado na manipulação do robô. Com objetivo de filtrar o código, descartar os blocos desnecessários e testar o programa Checkerboard [2], foi utilizado iterativamente o compilador Cmake, com o objetivo de compilar o *software*, detetar erros e corrigi-los. No entanto, o programa não estava completo e esta opção teve de ser abandonada.

A solução final encontrada para cumprir o objetivo traçado surgiu no programa Checker Bitboards [8]. Este é igualmente escrito em C e C++, mas tem uma interface gráfica na própria linha de comandos. Utilizando o mesmo método iterativo do programa CheckerBoard [2] foi possível modificá-lo. Isto conseguiu-se criar e modificar blocos de código que permitissem obter as informações necessárias para a construção da jogada por parte do *software*, em retorno de uma situação de jogo introduzida pelo utilizador.

Na fase final do projeto conciliou-se o programa Checker Bitboards [8] modificado com o ROS, num modelo “cliente-servidor”. Através do trabalho do colega Tiago Simões, o ROS recolhia e enviava, a situação do jogo e o *software* retomava a jogada que o manipulador deveria realizar. Cumpriu-se assim o objetivo inicial do trabalho.

Descrição (geral) do código do *software*

A filosofia do código é baseada num sistema de Bitmaps. Este sistema armazena a posição de peça por uma matriz de números hexadecimais matriciais, que representam a posição de uma peça ao longo do tabuleiro. As peças são categorizadas em 3 categorias: brancas, pretas e rainhas. Com base nisto, o *software* tem uma série de funções para pesquisar jogadas, verificar a sua veracidade e executá-las. Isto é possível através do resultado de operações lógicas de OR (“|”),

AND (“&”) e XOR (“^”).

Conclusão

A função do *software* foi de encontro ao objetivo inicial, permitindo assim a atividade cooperativa entre o sistema robotizado e um ser humano. É possível jogar com o manipulador um jogo de damas de maneira muito semelhante a um jogo com outro ser humano. O *software* tem algumas limitações, em termos de inteligência, dado que os jogos feitos na interface gráfica não foram os mais desafiantes e em termos da comunicação do *software* com o ROS.

Perspetiva de melhoramento

Tendo em conta as limitações do *software* mencionadas na conclusão, futuramente pode ser melhorado nomeadamente na sua inteligência e na comunicação com o sistema ROS. Relativamente à inteligência, é necessário rever o código alpha-beta e Ai-apha-beta e a sua implementação no *software*, que não foi implementado devido à limitação quer de tempo, quer de conhecimento. A sua comunicação com o ROS pode ser otimizada, pois neste momento é feita com valores obtidos de um modo pouco eficiente.

Webgrafia

- [1] - AI Checkers - <http://www.Planet-Source-Code.com/vb/scripts/ShowCode.asp?txtCodeId=8330&lngWId=1>
- [2] - CheckerBoard - <http://www.fierz.ch/checkers.htm>
- [3] - Checkers in cpp - <https://sourceforge.net/projects/checkersincpp/>
- [4] - Cobra Draughts - <https://github.com/THeK3nger/Cobra-Draughts>
- [5] - Gui Checkers - <http://www.3dkingdoms.com/checkers.htm>
- [6] - Raven Checkers - <https://github.com/bcorfman/raven-checkers>
- [7] - Robot Checkers - <https://github.com/AshishPrasad/Checkers>
- [8] - Checker Bitboards - <http://www.3dkingdoms.com/checkers/bitboards.htm>