

Integração de algoritmos de visão computacional para Lego Mindstorm (44)

Autor: Luís do Nascimento Carneiro Paiva Adães

Proponente: Miguel Coimbra

I Enquadramento do Trabalho

O propósito da visão robótica é permitir que os robots percepcionem o ambiente externo de forma a desempenharem um amplo de conjunto de tarefas como navegação, vigilância, tomadas de decisão com base em elementos visuais, categorização ou monitorização de elementos móveis ou estáticos. No amplo espectro de mecanismos existentes para percecionar e avaliar um ambiente (dispositivos laser, sónicos ou infra-vermelhos) a visão afirma-se como um dos mais pertinentes. No domínio da navegação autónoma a criação de mapas fiáveis que permitam uma navegação e localização precisas por parte de um dispositivo móvel tem sido, ao longo dos ultimos anos, um campo em constante evolução e relevância acrescida. Chamada de Odometria Visual, esta encontra-se intimamente ligada às técnicas de Visual SLAM (Simultaneous Localization And Mapping) e ,face à Odometria Convencional que acumula muitos erros quanto maiores as distancias percorridas, a Odometria Visual surge como uma alternativa pois a posição do dispositivo móvel é calculada através da informação obtida a partir das imagens que capta – pontos de referência ou campos óticos - sendo um método cada vez mais fiável e exequível dado o baixo custo de câmaras e poder de processamento de CPU e GPU na análise das imagens.

Aproximadamente 60 anos após a sua criação a LEGO lançou a o LEGO MINDSTORMS RCX Intelligent Brick and Robotics Invention System no Museu de Arte Moderna, em Londres, a Janeiro de 1998. Resultado de uma parceria que começou em 1985 entre o Media Lab do MIT e o LEGO Group, o LEGO Mindstorms é consequência da obra “Mindstorms:Children, Computers and Powerfull Ideas” escrita por Seymour Papert em 1980. Este livro procura apresentar uma nova forma de estimular e complementar os mecanismos naturais de aprendizagem das crianças melhorando a qualidade do conhecimento adquirido. É naturalmente uma linha juvenil de cariz pedagógico, no entanto, o LEGO Mindstorms pelo seu preço, facilidade de construção, integração com linguagens de programação de alto nível é uma ferramenta de investigação sólida na área da robótica. Em Janeiro de 2014, na disciplina de Visão Computacional, os alunos Edgard Quirino e Mário Pereira desenvolveram uma API em linguagem C# que permite aceder uma câmara montada num LEGO Mindstorm EV3, bem como funções de movimento e sensor de cor – a existência desta API e o meu gosto pessoal pela visão computacional surgem como motivações para este projeto.

A título poético o LEGO Mindstorms é a manifestação física deste projeto e a Odometria Visual a sua manifestação mental. Será controlado um LEGO Mindstorms EV3 através de um aplicação em C# para que, a partir de uma câmara de um smartphone montada no LEGO este seja capaz de realizar movimentos autónomos com vista a encontrar um objectivo definido pelo utilizador. Será tida por base a API anteriormente referida e utilizada a biblioteca OpenCV e o seu respectivo wrapper para C#(Emgu CV).

II Objectivos

O objectivo é criar uma aplicação para Windows em C# que, utilizando um smartphone com câmara integrada montado na estrutura do LEGO Mindstorms EV3, extraia informações acerca do ambiente a partir da imagem e promova o movimento autónomo em busca de um objectivo em terreno plano, apresentando ao utilizador uma interface que inclua um mapa (bi-dimensional e ponto de vista top-down), a imagem capturada pela câmara e a possibilidade de alterar o objectivo.

Define-se por objectivo uma imagem fornecida pelo utilizador (ex.: capa de um livro).

Tomando partido de diversas técnicas de navegação autónoma existentes no estudo da visão computacional, o robot deverá deslocar-se de forma autónoma devendo mapear a área envolvente e utilizar esse mapa na sua tomada de decisão relativa a movimentos futuros.

Atente-se que o propósito do robot é encontrar o objectivo fornecido pelo utilizador e não mapear a área completa, ou seja, encontrado o objectivo irá cessar a sua actividade comunicando ao utilizador o seu sucesso.

III Resultados Esperados

No final do projeto espera-se obter uma aplicação para Windows, que com ligação ao LEGO Mindstorm e através das imagens obtidas da câmara seja capaz de dotar o robot com as seguintes capacidades:

- 1 - Movimento autónomo por um ambiente arbitrário plano com detecção de obstáculos;
- 2 - Capacidade de Auto-localização e mapeamento da área envolvente (Odometria Visual);
- 3 - Análise do mapa em tempo real para promover o movimento autónomo;
- 4 - Detectar um objectivo definido pelo utilizador e cessar a sua actividade quando encontrado;
- 5 - Uma interface simples e directa que permita alterar objectivos e visualizar o percurso do robot bem como o ambiente envolvente através de um mapa top-down(bi-dimensional).

IV Identificação das Tarefas

1 Estudo Teórico e preparação dos recursos necessários: Nesta fase inicial serão escolhidos quais os algoritmos e técnicas necessárias, a configuração do robot e instalação das bibliotecas necessárias à programação do mesmo;

2 Reconhecimento de objectivos: Dada uma imagem tomada como objectivo, ser capaz de reconhecer a sua presença em frente da câmara. Este passo, embora apenas utilizado no final, é importante para criar uma maior familiarização com a biblioteca OpenCV;

3 Mapeamento estacionário: Aceder à camara, montada no LEGO Mindstorms, e manipular a imagem obtida de forma a obter contornos do ambiente, pontos de referência e respectivas distancias a obstáculos efectuando e apresentando um mapeamento primário da área. Nesta fase não será implementado qualquer movimento, mas é nesta fase que serão desenvolvidos os processos criação do mapa bem como a sua apresentação ao utilizador. Espera-se que nesta que todos os procedimentos de manipulação e análise de imagem estejam implementados mas não integrados com o LEGO.

4 Mapeamento dinâmico: Adaptar os processos desenvolvidos na fase precesente aos movimentos de translação e rotação do robot. Nesta fase, o movimento será controlado directamente pelo utilizador via a API desenvolvida pelos alunos Edgard Quirino e Mário Pereira sendo implementada a Odometria Visual em pleno;

5 Análise do mapa: Análise do mapa obtido e retirar conclusões em tempo real sobre o melhor destino a seguir. A análise rigorosa do mapa permitirá referenciar áreas de interesse que irão definir a essência do movimento autónomo;

6 Movimento Autónomo: Mediante as conclusões retiradas da análise do mapa implementadas na fase anterior, implementar o movimento autónomo do robot com os ajustes à polaridade dos motores e velocidade do robot;

7 Interface com o utilizador: Integração do mapa obtido, imagem da câmara e escolha de objectivos numa interface coesa e bem definida;

8 Testes e aprimorações da aplicação: Esta fase é de particular importância para calibrar motores, tratamento da imagem e validar procedimentos precedentes;

V Identificação de Milestones

1 Mapa obtido pela movimentação controlada pelo utilizador (Odometria Visual): A aplicação neste momento encontra-se a 50% da sua conclusão, os mecanismos que irão reger o movimento autónomo já estão completamente implementados e já é possível visualizar um mapa;

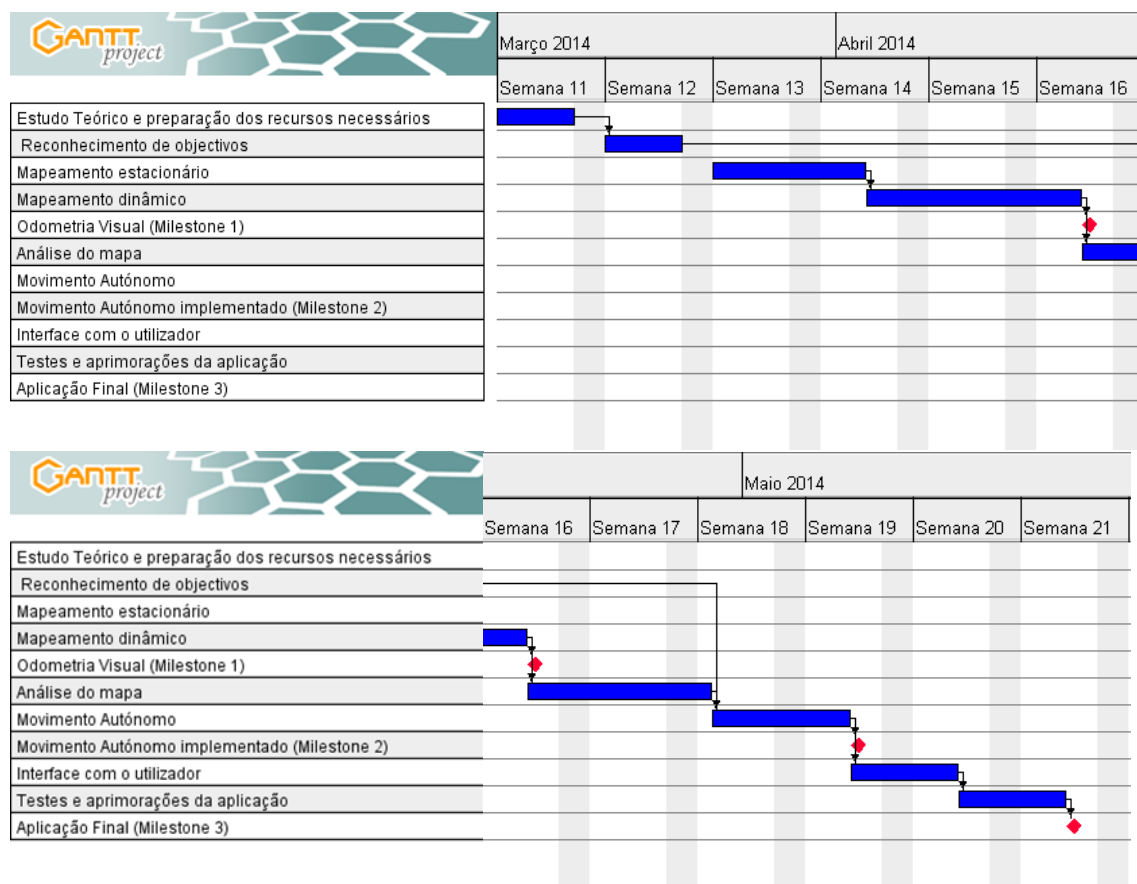
2 Movimento Autónomo implementado: Estando a Odometria Visual completamente desenvolvida esta fase representa o projecto completo a 80%. O Robot já retira conclusões do mapa obtido e já se movimenta de forma autónoma;

3 Aplicação Final com interface com o utilizador: Produto final com interface que permita ao utilizador visualizar resultados de forma intuitiva;

VI Recursos

- 1 - LEGO Mindstorms EV3;
- 2 - Computador com suporte para programação em C# e bluetooth para comunicação com o LEGO Mindstorms EV3;
- 3 - Biblioteca OpenCV e wrapper para C# (EMGU)
- 4 - Smartphone com câmara;
- 5 - API desenvolvida pelos alunos Edgard Quirino e Mário Pereira.

VII Mapa de Gant



Nota: O mapa de Gantt foi Dividido por uma questão de paginação;

Todo o planeamento foi feito por Semanas onde se espera trabalhar no projeto um pouco todos os dias para evitar possíveis atrasos;