

SISTEMA DE MAPEAMENTO DE AMBIENTES INTERNOS DINÂMICO UTILIZANDO SENSORES MICROSOFT KINECT

Stephanie A. A. Castueira ¹, Rebecca Lovato Franco ¹; Murilo Zanini de Carvalho ²

¹ Aluna de Iniciação Científica da Escola de Engenharia Mauá (EEM/CEUN-IMT);

² Professor da Escola de Engenharia Mauá (EEM/CEUN-IMT).

Resumo. *O projeto de pesquisa teve como objetivo o desenvolvimento de um estudo sobre as técnicas de mapeamento em ambientes dinâmicos utilizando o sensor Kinect, possibilitando a utilização de suas informações para a navegação de robôs autônomos. Diante disso, primeiramente foram realizados estudos sobre assuntos relacionados ao tema, por exemplo, SLAM (Navegação e Mapeamento Simultâneo de Ambientes) e comunicação MASTER SLAVE. Em seguida, deu-se o processo de familiarização com o sensor escolhido, Microsoft Kinect, para captura de imagens. E, por fim, realizou-se o refinamento e implementação dos métodos selecionados para o projeto, utilizando: RTABMAP, software responsável por criar o mapa; Raspberry, como pré-processador das imagens; Máquina-virtual Ubuntu, como o processador principal; ROS, sistema operacional de robôs, como um meio de transmissão entre máquina virtual e o Raspberry. Destacam-se, como resultados obtidos, a construção de mapas de obstáculos para o sistema autônomo de navegação em desenvolvimento pelo grupo de pesquisa de veículos autônomos da instituição.*

Introdução

Levando-se em consideração o atual cenário brasileiro, que possui um alto índice de acidentes no trânsito, e o cenário mundial, que mostra um grande desenvolvimento tecnológico. Nota-se que, em um futuro muito próximo, haverá uma maior demanda por tecnologias capazes de solucionar problemas causados por seres humanos. Como por exemplo os Veículos Autônomos que, ao eliminar o fator do erro humano, são capazes de evitar acidentes no trânsito.

O projeto faz parte da iniciativa do grupo Sistemas Mecatrônicos Inteligentes e Robótica para a construção de um veículo híbrido autônomo. Dessa maneira, este trabalho visa o desenvolvimento de mapas para ambientes internos utilizando fusão dos dados de sensores Kinect.

No que tange ao sistema de mapeamento propriamente dito, o objetivo é construir um sistema capaz de utilizar as informações adquiridas pelo sensor e processadas pelo computador, de modo a permitir que robôs autônomos consigam criar sua própria rota. Além disso, ser apto para disponibilizar mapas e dados do ambiente para utilização em outras aplicações.

O SLAM é uma técnica essencial para o desenvolvimento de dispositivos autônomos que cada vez mais se mostram determinantes para auxiliar na solução de questões da sociedade atual. Eles podem ser implementados em diversas áreas como por exemplo no transporte, com carros autônomos e veículos aéreos e aquáticos não tripulados.

A navegação e mapeamento simultâneo de ambientes (SLAM em inglês) é uma técnica fundamental para a robótica móvel e para a construção de veículos autônomos. Uma vez que, neste processo é possível realizar o mapeamento de ambientes desconhecidos, e ao mesmo tempo, navegar nesse local utilizando o seu próprio mapa. Diversas técnicas são utilizadas para realizar essa tarefa, sendo um tema muito relevante em inúmeras áreas de pesquisa.

Ao analisar alguns artigos verifica-se que o mapeamento de ambientes em uma única sessão é um tópico amplamente discutido [1][2], porém para que o projeto possa ser aplicado em situações mais realistas, o mesmo foi direcionado para o mapeamento de ambientes em múltiplas sessões, no qual compreende-se que o robô pode ser desligado e movido para outro local. Consequentemente, ao ser religado o robô não sabe a sua posição atual com relação ao mapa que foi criado anteriormente. Em uma das abordagens utilizadas para lidar com este problema o robô se localiza no mapa já existente, porém há a necessidade que o robô esteja em um local já mapeado. Portanto, uma solução alternativa foi escolhida, onde o robô executa um novo mapeamento e ao percorrer um

local conhecido, cria-se um novo mapa a partir da fusão dos anteriores [3]-[6].

Além disso, encontra-se um tipo de protocolo de rede onde é possível comandar o robô móvel que está vinculado com o sensor por outro dispositivo central (MASTER). Enquanto a comunicação MASTER SLAVE, é caracterizada pela transmissão de dados pelo SLAVE para a rede através da autorização do MASTER, que efetua a coordenação do tráfego.

Nesse mesmo contexto de desenvolvimento tecnológico, o conceito que associa todas as informações adquiridas por diversas aplicações do sensor é o de fusão de dados. Onde ocorre integração das informações de diversas fontes não correlacionadas que representam ou referenciam o mesmo objeto.

Essa tecnologia vem evoluindo exponencialmente e ao ser implementada trará impactos positivos quando se trata da resolução de erros causados pelo homem. Um exemplo são os carros autônomos, capazes de realizar as mesmas tarefas que o cérebro humano ao dirigir e com menos probabilidade de cometer erros, pois o seu tempo de resposta ao analisar uma situação é menor do que o de uma pessoa. De acordo com um estudo da Organização Mundial de Saúde (OMS) cerca de 1,3 bilhões de pessoas morrem em acidentes de trânsito ao redor do mundo e, com a implementação de veículos autônomos, a expectativa é que esse número seja reduzido em milhão de mortes por ano.

Além da diminuição do número de acidentes, os carros autônomos trarão outras consequências positivas que irão auxiliar no dia a dia da sociedade, como a redução do tempo perdido no trânsito, a possibilidade de o passageiro utilizar a duração da viagem para realizar outras tarefas, a diminuição do número de carros na rua e, conseqüentemente, as emissões de gás carbônico.

O desenvolvimento da tecnologia a ser implementada em dispositivos autônomos impactará não somente o cotidiano da sociedade, mas também trará uma gama de possibilidades no meio acadêmico, irá incitar novas pesquisas e possibilitará o desenvolvimento e aprimoramento de novas tecnologias.

Dessa maneira, o Microsoft Kinect, junto com Raspberry, com a transmissão fornecida pelo ROS e com o controlador (PC) foram apresentados como um dispositivo capaz de implementar e integrar todos os conceitos exigidos pelo trabalho. Dentre os diversos atributos que essa tecnologia trouxe ao mercado, vale destacar o custo dessa solução como muito inferior aos competidores disponíveis.

Material e Métodos

Como o projeto tem como objetivo fazer o mapeamento de ambientes internos, nota-se que para o desenvolvimento da pesquisa utilizou-se dois tipos de materiais: Hardware e Software.

Como Hardware utiliza-se: Kinect, para obter o melhor resultado ao capturar imagens; um computador com máquina virtual e sistema operacional Ubuntu, como o processador principal conectado com SLAVE pelo ROS (seção 2.2.4); um Raspberry com sistema operacional Ubuntu Mate, como pré-processador das imagens, no qual está conectado ao sensor Kinect pelo USB.

O sensor Kinect além de utilizado em vídeo games como o Xbox 360, também pode ser aplicado para fins acadêmicos, uma vez que seu custo não é tão alto ao ser comparado com os outros sensores e este ainda possui uma alta qualidade de processamento e na obtenção de dados.

Dessa maneira, podemos caracterizar o Kinect como um sensor de movimentos que possui um emissor de infravermelho, um sensor de profundidade, uma câmera RGB e um microfone direcional com um bom sistema de reconhecimento de voz.

O emissor é usado para projetar quadriculado, como um projetor comum, porém esse utiliza luz infravermelha. O sensor, ou receptor, é usado para capturar a imagem do padrão projetado com luz infravermelha e as distorções do padrão são interpretadas como variações na distância do objeto. Desta forma ele realiza o processamento em tempo real necessário para extrair o mapa de profundidade a partir da imagem do quadriculado. A câmera RGB é totalmente tradicional, colorida, como um webcam qualquer.

Além disso a Microsoft integrou ao Kinect softwares para diferentes sistemas operacionais: para Windows, utiliza-se o Visual Studio Community Edition 2015, com o SDK 1.8 (vinculado com API v1 para Kinect para o Xbox 360); para o Linux utiliza-se o freenect e openCV.

Já o Raspberry Pi, outro dispositivo utilizado no projeto, é um computador com o tamanho reduzido, sendo apenas a representação do processador e, assim, pode ser conectado com um monitor e com um teclado, formando um minicomputador de baixo custo.

Enquanto aos softwares operou-se com: ROS, sistema operacional de robôs, como um meio de transmissão entre Ubuntu e Raspberry; RTABMAP, para fazer o mapeamento simultâneo do ambiente.

Robot Operating System (ROS) é uma coleção de softwares que auxilia no desenvolvimento de sistemas robóticos. Um dos seus principais objetivos é promover a comunicação entre o usuário, o sistema operacional e o hardware. Logo, um dos benefícios da sua utilização é a habilidade de controlar dispositivos sem a necessidade de conhecer todos os seus detalhes.

Além disso, esse sistema possui o código fonte aberto (Open-Source), que possibilita o seu uso sem a necessidade de licença; viabiliza a alteração do seu código pelo usuário, caso haja necessidade; e seus custos são baixos, o que lhe torna vantajoso em relação a outros sistemas.

Dessa maneira, um dos principais objetivos do Sistema Operacional de Robôs (ROS), um dos softwares escolhidos para o desenvolvimento do projeto, é viabilizar a comunicação entre o usuário, o sistema operacional do computador (Linux) e o hardware externo, neste caso o Raspberry que está conectado e recebendo informações do Kinect.

Com o propósito de facilitar o entendimento do processo de comunicação implementado, é necessário o entendimento de alguns conceitos relacionados ao ROS.

Primeiramente, ROS Nodes são processos que realizam determinadas tarefas independentes e são capazes de se comunicar mandando e recebendo mensagens, que podem conter dados, comandos e outras informações necessárias, com o ROS Master e com outros Nodes no sistema.

Como dito anteriormente, alguns Nodes fornecem informações para outros e o conceito que define quais dados serão enviados e/ou recebidos por determinados Nodes é denominado tópico.

Analogamente, pode-se pensar em um tópico como uma variável global de uma aplicação. Ou seja, um Node pode se inscrever em um tópico e, então, todas as mensagens transmitidas para ele serão recebidas pelo Node inscrito. Um outro conceito importante são as mensagens, que são os próprios dados enviados e recebidos pelos Nodes.

Por fim, ROS Master é o responsável pelo gerenciamento das comunicações que ocorrem no sistema. O Master é encarregado de fornecer nomes e registros para os Nodes, para que esses sejam capazes de se localizar e se comunicar sem que haja perda ou envio de informações para destinos errados. Os Nodes não podem se comunicar enquanto o Master não notificar ambos da existência um do outro.

Na abordagem utilizada para a comunicação, a Máquina Virtual foi selecionada como ROS Master e, portanto, para iniciar a comunicação utilizando o ROS, primeiramente o ROS Master foi criado utilizando um processo chamado roscore no Ubuntu. Para que a comunicação seja bem-sucedida, todos os dispositivos necessitam estar conectados na mesma rede Wi-Fi. Além da necessidade do Master, é preciso escolher qual será o dispositivo que receberá as informações e qual as enviará. Neste caso, o Ubuntu está inscrito no tópico que está sendo publicado pelo Raspberry e que contém as informações das imagens coletadas.

Para que esta comunicação entre Publisher e Subscriber seja viável, é necessário obter o endereço Internet Protocol (IP) de cada dispositivo e, então, exportar em ambos o endereço do Master e do IP do dispositivo. Isto é exemplificado no código a seguir:

```
...
```

```
$ export ROS_MASTER_URI=http://<IP address of TurtleBot>:11311
```

```
$ export ROS_IP=<IP address of remote computer>
```

```
...
```

Como o objetivo do projeto é ser capaz de obter o mapa remotamente e, futuramente, isso será realizado utilizando um robô móvel, é importante que o Raspberry seja acessado remotamente.

Para isso, foi eleita a conexão Secure Shell (SSH) e um exemplo de acesso remoto a um dispositivo é apresentado a seguir:

```
...  
$ ssh <username>@<IP address of TurtleBot>  
...
```

Utilizando essa abordagem foi possível a comunicação entre a Máquina Virtual e o Raspberry o que, conseqüentemente, permitiu a aquisição dos dados obtidos com o Kinect pelo Ubuntu, aonde é realizado o processamento.

Outro software utilizado foi o RTAB-MAP, um software encontrado no pacote do ROS, onde é abordado o conceito de SLAM para criar o gráfico RGB-D a partir do incremento contínuo (loop) realizado pelo sensor. Assim, esse aplicativo gerencia a memória para limitar o tamanho dos mapas usado para localização e mapeamento de ambientes.

Dessa maneira, para realizar o SLAM com robôs, nos dias de hoje, é necessário que este atualize o mapa de maneira contínua, isto significa que quanto mais tempo ele está ligado maior será volume de informações encontrada no mapa, e é neste ponto que entra o software. Ele limitará e gerenciará o conteúdo do mapa atual, alterando sua memória de trabalho (Working Memory), e a memória que não será imediatamente utilizada e que, portanto, será descartada da atual, mas salva numa memória de longo prazo (Long Time Memory) para qualquer tipo de recuperação em qualquer momento caso necessário.

A solução escolhida para os problemas previamente citados foi a utilização do RTABMAP, uma abordagem de gestão de memória utilizada para que seja possível a realização da detecção em loop em tempo real por robôs móveis em ambientes grandes e em operações de longa duração.

Esta solução consiste na utilização de um certo número de ambientes, para limitar o tempo de processamento, enquanto ainda seja possível o acesso ao mapa completo, caso seja necessário.



Figura 1 - Materiais conectados utilizados: Raspberry (junto com o display) e Kinect

A metodologia utilizada na iniciação científica foi dividida em algumas etapas, em um primeiro momento foi realizado uma pesquisa em artigos científicos sobre os assuntos relacionados com o sistema que seria desenvolvido, de mapeamento interno dinâmico.

Após essa pesquisa ocorreu o processo de familiarização com o sensor escolhido, Kinect. Assim, instalou-se no próprio Windows o API v1 (Kinect para Xbox 360), e o Visual Studio Community Edition (2015) para assim, a partir dos exemplos fornecidos pelo próprio sistema do sensor, realizar algumas alterações nos códigos de captura de imagem com a câmera e com o sensor IR.

Ainda nessa mesma etapa, elaborou-se um código para a criação de um arquivo de texto com os dados obtidos na captura de distância medidas pelo sensor IR.

E por fim, para elaborar e implementar um mapa dinâmico foram testadas algumas maneiras distintas:

Kinect Fusion: Cria uma cena 3d, mas não salva um mapa. A criação do mapa é importante para que, assim como a dinâmica do ambiente, sejam realizadas apenas alterações em lugares em que houve movimento.

Reconstruct me: Salva apenas os objetos mais próximos, ignorando fundos importantes para o mapeamento.

RTAB-Map: Construção de mapas dinâmicos de uma maneira lenta, porém de alta qualidade, identificando o caminho do sensor para a construção do ambiente 3D.

Dentre as opções testadas, o que obteve melhor resultado foi o RTAB-Map, mas seu processamento no Windows não era tão bom por ser uma aplicação do ROS.

Dessa maneira realizou-se algumas pesquisas e obtivemos como resultado que o sistema operacional mais compatível com o ROS é o sistema Linux, assim, instalou-se uma máquina virtual no computador com o Linux e o ROS. Depois de instalados, testou-se o Kinect nesse ambiente da máquina virtual, mas pelo fato desta ser um processamento periférico do computador, o compartilhamento do Kinect não era compatível.

Como uma das etapas finais do desenvolvimento da iniciação houve a implementação do Raspberry Pi 3, com o sistema operacional Ubuntu Mate. Mas, como a própria marca o caracteriza como pequeno computador, nota-se que o processamento e a memória disponível para a utilização não são grandes.

Para a resolução desse problema realizou-se uma pesquisa e concluiu-se que é necessário obter um sistema em rede conectado. Isto é, ligar o Kinect no Raspberry Pi 3, uma vez que não é compatível vinculá-lo diretamente na máquina virtual, e, depois, conectar esse Raspberry com a máquina virtual de forma que o computador se tornaria o Master e o controlaria por comandos na rede.

Por fim, depois de realizar essa comunicação é necessário transferir a imagem obtida pelo Kinect no Raspberry para a máquina virtual e, então, realizar o mapeamento dinâmico. Nessa última etapa em um primeiro momento o mapeamento foi realizado no ambiente do rviz, mas como o Kinect utilizado não estava integrado com um robô móvel esse procedimento foi descartado, e, por fim, utilizou-se RTAB-Map.

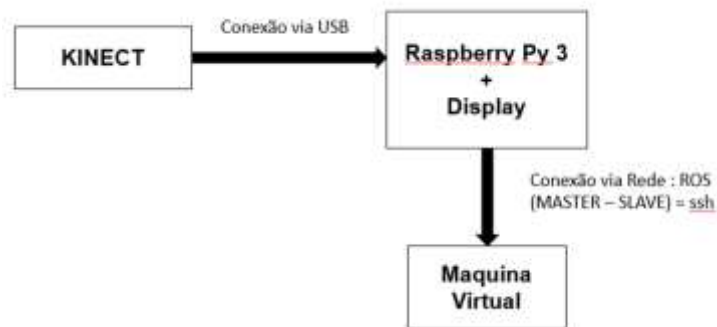


Figura 2 - Diagrama de blocos da conexão do sistema

Resultados e Discussão

Durante o trabalho foram realizados diversos ensaios, mas os que mais se relacionam com o tema do trabalho estão demonstrados a seguir.

Como dito anteriormente, em um primeiro momento, foi elaborado um programa para que este criasse um arquivo de texto cada vez que clicasse no botão denominado “Button”. Desta maneira, eram obtidos os valores das distâncias de cada objeto encontrado no ambiente e encontrado pelo Kinect.



Figura 3 - Interface do programa criado para transformar distancia em arquivo texto e demonstração do arquivo

Depois de entender o funcionamento do sensor Microsoft Kinect e de pesquisar a existência de várias possibilidades para realizar o mapeamento com este, como: Kinect Fusion, Reconstruct me e RTAB-Map.

Dentre os ensaios realizados, a opção que obteve um resultado mais adequado para o desenvolvimento do projeto foi o RTAB-Map, mas por ser uma aplicação do ROS, que possui um melhor processamento em Linux, faz com que seu processamento no Windows não seja adequado.



Figura 4 - mapeamento RTAB-Map no Windows

Em uma segunda etapa, utilizou-se esse mesmo software, mas em outro sistema operacional, o Linux, acrescentando o hardware Raspberry Pi. Assim como na primeira etapa, utilizou-se o método de mapeamento de longa duração em ambientes dinâmicos pois, ao atualizar o mapa continuamente em ambientes ativos, o seu tamanho tende a aumentar significativamente ao longo do tempo. Conforme o tamanho do mapa cresce, o tempo necessário para realizar o mapeamento e a detecção em loop também crescem, o que eventualmente irá limitar o tamanho do ambiente que pode ser mapeado.

Além disso, na segunda etapa houve uma comunicação entre a máquina virtual e o Raspberry através do sistema de redes do ROS. Isto é, um node, programa capaz de rodar simultaneamente em vários sistemas, captura as imagens do sensor Kinect e as manda para outro node, que realiza o processamento das mesmas.



Figura 5 - Imagem RGB (esquerda) e de distância (direita) do Kinect adquirida pela conexão ROS entre Raspberry e Máquina Virtual

Desta maneira, os resultados obtidos durante o projeto mostram que com a utilização do RTAB-Map, a abordagem de gestão de memória escolhida, é possível alcançar as condições exigidas para que o robô realize a Navegação e Mapeamento Simultâneo de Ambientes em um longo período de tempo e em ambientes dinâmicos.

Além disso, notou-se que este software tem um melhor desempenho em um sistema operacional de robôs, que por sua vez tem melhor performance no sistema Linux (Ubuntu) e, portanto, este foi adotado.

Porém, observa-se o processamento é superior quando realizado diretamente na máquina, uma vez que o Kinect não consegue emitir dados para a máquina virtual. Assim, para solucionar esse problema, instalou-se o Linux e o ROS no Raspberry, liberando ferramentas que auxiliam na construção de aplicações para robôs.

A conexão entre a máquina virtual e o Raspberry foi feita por rede, utilizando seus IP foi possível comunicar um dispositivo com o outro. E, assim, criar um modo de comunicação MASTER-SLAVE, onde a máquina virtual encontrada no computador gera comandos sobre os Raspberry, e este fornece os dados solicitados.

Por fim, depois da comunicação, obteve-se o resultado final, no qual foi o realizado o mapeamento do ambiente interno dinâmico a partir dos dados obtidos pelo Kinect e compartilhado pela rede ROS implementada no Raspberry.

Conclusões

O projeto de pesquisa apresenta o estudo das técnicas de mapeamento de ambientes de forma dinâmica utilizando o sensor Microsoft Kinect.

Destacam-se, como resultados, a disponibilização de dados do ambiente para utilização em outras aplicações; a possibilidade de repetição de procedimentos executados em ambientes internos para ambientes externos, realizando as devidas adaptações no sensor utilizado; e, a contribuição para a construção do sistema autônomo de navegação em desenvolvimento pelo grupo de pesquisa.

Atualmente, as informações obtidas são salvas no computador em que é realizado o processamento final e somente um dispositivo tem acesso a tais dados. Porém, futuramente, para que seja possível o acesso às informações coletadas por diversos dispositivos, será necessária a elaboração de um banco de dados, preferencialmente em uma nuvem. Este irá armazenar todos os dados coletados e possibilitará que os dispositivos acessem informações de ambientes antes mesmo de estes terem realizado o próprio mapeamento. O que também resultará na atualização constante dos mapas criados pelos dispositivos que por eles transitarem.

Também como trabalhos futuros, encontra-se a implementação desse sistema utilizando um robô controlado remotamente e a utilização deste num ambiente externo, como uma iniciativa para que, eventualmente, torne-se uma solução para sanar problemas sociais de trânsito. Dessa maneira, os Veículos Autônomos seriam capazes de evitar erros humanos o que culminaria em um menor índice de acidentes na cidade.

Referências Bibliográficas

- [1] M. Bosse, P. Newman, J. Leonard, and S. Teller, “Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework,” *Int. J. of Robotics Research*, vol. 23, no. 12, pp. 1113–39, 2004.
- [2] S. Thrun and M. Montemerlo, “The graph SLAM algorithm with applications to large-scale mapping of urban structures,” *Int. J. of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.
- [3] A. Angeli, S. Doncieux, J. Meyer, and D. Filliat, “Incremental vision based topological SLAM,” in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 1031–1036.
- [4] T. Botterill, S. Mills, and R. Green, “Bag-of-words-driven, single-camera simultaneous localization and mapping,” *J. of Field Robotics*, vol. 28, no. 2, pp. 204–226, 2011.
- [5] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, “View-based maps,” *The Int. J. of Robotics Research*, vol. 29, no. 8, pp. 941–957, July 2010.
- [6] M. Labbé and F. Michaud, “Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation,” in *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 734-745, 2013. (IEEE Xplore)