



COMPROMETIDA COM A PROMOÇÃO DO DESENVOLVIMENTO DA ENGENHARIA E DAS CIÊNCIAS MECÂNICAS

VI CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA
VI NATIONAL CONGRESS OF MECHANICAL ENGINEERING
18 a 21 de agosto de 2010 – Campina Grande – Paraíba - Brasil
August 18 – 21, 2010 – Campina Grande – Paraíba – Brazil

MAPEAMENTO DE AMBIENTES “INDOOR” UTILIZANDO MAPAS MULTI CAMADAS DE SUPERFÍCIE

Antônio Baptista Júnior, bodninja@sc.usp.br¹
Jorge Pablo Moraga Galdames, jorge.pablo@gmail.com¹
Marcelo Becker, becker@sc.usp.br¹
Glauco Augusto de Paula Caurin, gcaurin@sc.usp.br¹
Denis Fernando Wolf, denis@icmc.usp.br²

¹ Escola de Engenharia de São Carlos - USP
Departamento de Engenharia Mecânica
Av. Trabalhador São-Carlense, 400
CEP 13566-590 - São Carlos - SP

² Instituto de Ciências Matemáticas e de Computação - USP
Av. Trabalhador São-carlense, 400 - Centro
Caixa Postal: 668 - CEP: 13560-970 - São Carlos - SP

Resumo: *A tarefa de mapear um ambiente torna-se indispensável na robótica móvel, porque os robôs necessitam de informações do ambiente para realizar tarefas de alto grau de dificuldade. Neste trabalho mostramos uma forma para representação de mapas 3D que pode ser utilizada para planejamento de rota e localização. Para representar o mapa utilizamos o mapeamento em “grid” discreto já conhecido na robótica. Mas para que possamos mapear ambientes 3D fizemos as modificações que são propostas nos mapas de multi camadas de superfície, esta proposta armazena múltiplas superfície em cada “grid” da célula. Esta tarefa foi possível com a utilização de um robô ActiveMedia Pioneer equipado odometria e sensor laser. Os dados adquiridos pelo robô são processados por um algoritmo de mapeamento em “grid” o qual foi observado a necessidade da implementação futura de um algoritmo para SLAM, implementado com filtro de Kalman. Com estes métodos conseguimos representar um ambiente “indoor” de forma precisa e armazenando uma quantidade pequena de informações. Nesta abordagem a representação de estruturas verticais, como portas, janelas e rampas são possíveis. O experimento foi em ambiente “indoor”. Foram criados logs de navegação do robô utilizando o software simulador Player para que pudéssemos repetir o experimento sem a necessidade do robô e estando disponível para trabalhos futuros.*

Palavras-chave: *mapeamento robótico, mapeamento grid de ocupação, player*

1. INTRODUÇÃO

Este trabalho tem como finalidade o estudo e desenvolvimento do módulo de mapeamento 3D que será utilizado em robôs móveis para ambientes “indoor”. A proposta do desenvolvimento do módulo atende a robôs móveis “indoor” que desempenham as tarefas de transporte e armazenagem de materiais, guias de museus, robôs militares para reconhecimento de área, para todos os robôs que necessitam da tarefa de mapeamento e reconhecimento de ambientes. Por exemplo: armazéns, minas de carvão e indústrias. Segundo Thrun (2002) “Mapeamento robótico tem sido o foco das pesquisas nas áreas de robótica e inteligência artificial nas 2 últimas décadas.”

Para realização do trabalho foi utilizado um robô equipado com odometria e sensor laser. Os dados coletados pelo robô são processados e armazenados por um algoritmo de mapeamento “grid” discreto. Para obter maior precisão na construção do mapa, é necessário um algoritmo para mapeamento e localização simultânea (SLAM) que utiliza um filtro de kalman, aplicado ao método de criação de mapas multi camadas de superfície, para obter uma localização precisa e correção da posição do robô, possibilitando a criação de um mapa preciso.

2. TRABALHOS RELACIONADOS

A tarefa de mapear ambientes vem sendo estudada e trabalhada desde a proposta apresentada por Elfes (1987), que propôs a criação de mapas planares conhecidos por mapas de ocupação em “grid”, logo em seguida descobriu-se a

necessidade da representação no espaço criando-se o que conhecemos na robótica como mapas elevados apresentado por Bares et al (1989) e por Hebert et al (1989). No mesmo sentido do mapeamento de ambientes “indoor” uma proposta apresentada por Thrun (2002) mostrou um foco especial ao mapeamento, apresentando várias técnicas probabilísticas para mapeamento e mostrou os problemas encontrados na tarefa de mapeamento. No trabalho apresentado por Pfaff e Burgard (2005), uma nova abordagem para criação de mapas, sendo então conhecido como uma extensão dos mapas elevados, nas pesquisas apresentadas logo em seguida chamaram de mapas multi-camadas de superfície e mostradas por (Tribel et al, 2006; Lamon et al, 2006; Pfaff et al, 2007 e Kümmerle et al, 2007).

A vantagem da utilização dos mapas multi-camadas de superfície é mostrada na figura (1), onde vemos detalhes maiores sendo apresentados em comparação com mapas elevados. A vantagem da abordagem de mapas MLS em comparação com mapas elevados padrão. Em contraste com mapa MLS (direita) a mapa elevado (à esquerda) não tem a capacidade para modelar estruturas verticais, porque ele faz a médias de todos os valores de altura medidos. Uma vez que a distância do ponto de extremidade de um feixe de laser para o próximo ponto no mapa elevado pode ter desvios substanciais da verdadeira distância, a localização torna-se difícil.

2.1. Problemas apontados na tarefa de mapeamento

Um ambiente dinâmico como um laboratório de pesquisa, onde encontramos caixas (espalhadas, empilhadas e nos mais diversos tamanhos), corredores de máquinas, pessoas e pontos de interesse (bases de alimentação, pontos de carga e descarga, passagens possíveis e impossíveis, sinalizações), neste ambiente incluímos a tarefa de transportar um objeto de um ponto ao outro. Podemos abstrair a ideia deste ambiente com a figura 4.

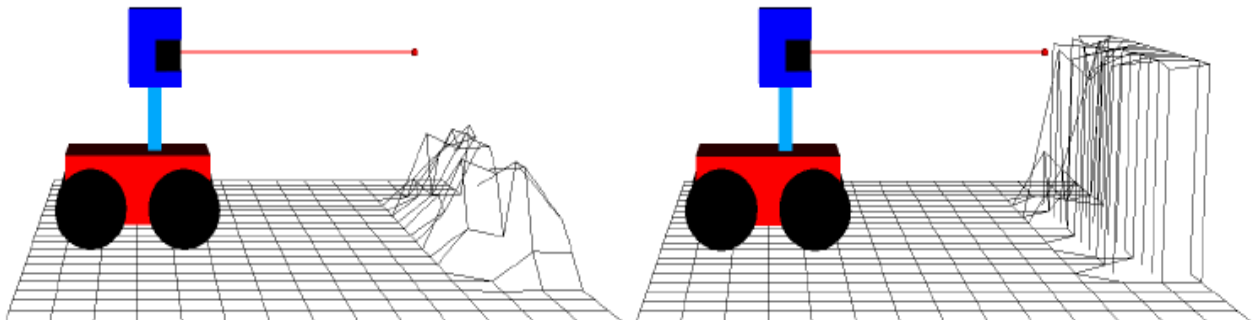


Figura 1: Mapa elevado à esquerda e Mapa MLS à direita, Kümmerle et al (2007). Cortesia de Rainer Kümmerle.

No ambiente proposto a necessidade de abstrair o meio em que o robô se encontra torna-se indispensável. Surge a necessidade de representar os seis graus de liberdade, torna-se ainda maior devido a passagens em rampas subterrâneas, portas, maquinário parado e operando. Esta necessidade de abstrair aumenta ainda mais na situação apresentada, quando imaginamos um robô realizando a tarefa de transportar caixas, peças ou paletes com lugares pré-definidos para carga ou descarga.

3. REALIZAÇÃO DO EXPERIMENTO

No experimento utilizamos um robô equipado com odometria e um sensor Laser SICK LMS 200. Para aquisição de dados optamos pelo software Player, muito utilizado em pesquisas de robótica devido as vantagens que ele fornece. Algumas das vantagens que apontamos neste trabalho são, transparência na aquisição de dados do sensor laser, transparência na aquisição dos dados de odometria, geração de arquivos de logs os quais podem ser utilizados inúmeras vezes podendo repetir o experimento mesmo que não seja no ambiente físico e podendo ser reutilizados em outros projetos e ser uma plataforma cliente/servidor controlada remotamente.

Como o foco do trabalho foi a criação de uma representação do ambiente em 3 dimensões adotamos a configuração do sensor laser mostrada na figura 2. Com esta configuração suprimos a necessidade de informações que precisamos para obtenção do ambiente 3D. As informações que foram utilizadas são, posição do robô fornecida pela odometria (x,

y , Θ) e posição dos objetos que circundam o robô com o laser fornecendo o eixo (z) a partir de cálculos simples, o sistema de coordenadas é apresentado na figura (3)..

Os demais ângulos necessários para criação de mapas multi camadas de superfície não foram utilizados neste primeiro experimento devido ao fato do trajeto realizado pelo robô, os ângulos (Φ e Ψ) não sofreram variações, o robô percorreu uma área plana e ocorreu nos sentidos indicados na figura 5, sendo o trajeto de ida partindo do ponto A para o ponto B e o trajeto de volta com partida em B para o ponto A . A distância percorrida foi aproximadamente 16 metros baseada na odometria e sem o devido tratamento para os erros.

O experimento aconteceu em 4 etapas, criando 4 arquivos de logs para trabalhos futuros e offline. Para que o software Player utilize estes arquivos foram necessários a criação de 4 arquivos de configuração para realizar a leitura dos arquivos e aquisição dos dados dos sensores.

Para repetir o experimento *offline* utilizamos o sistema operacional Linux/Ubuntu 9.04 rodando em uma máquina virtual construída com o software Sun Virtualbox rodando em um sistema operacional Windows Vista. Isso implica uma perda de desempenho que neste experimento não foi necessário considerar.



Figura 2: Configuração do sensor laser utilizado no experimento

4. RESULTADOS PARCIAIS

Nesta seção discutiremos alguns resultados obtidos com o experimento e a simulação com o software Player.

No experimento simulado extraímos os pontos que foram obtidos através do sensor, transformando as coordenadas polares em coordenadas espacial. Em seguida, observamos os pontos obtidos utilizando o software Matlab. O numero de pontos obtidos foi elevado, devido a falhas no sensor e redundância dos dados, tornando necessário a implementação de um algoritmo para mapeamento, que trouxe uma redução no quantidade de pontos, criando um mapa mais preciso e com quantidade de pontos menor.

Com pequenas modificações no código de mapeamento 3D, extraímos uma mapa 2D, para esta tarefa, foi considerado apenas as leituras na faixa de 0 a 45 graus e de 135 a 180, construindo uma representação 2D do ambiente, onde foi gerado uma imagem com a utilização da biblioteca openCV para linguagem C.

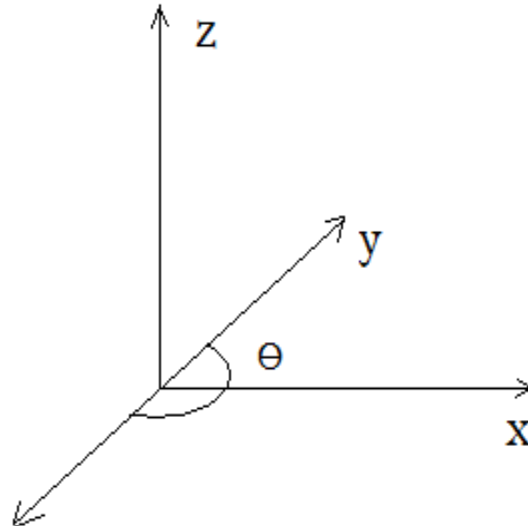


Figura 3: Sistema de coordenadas utilizado no experimento



Figura 4: Indicação do trajeto percorrido pelo robô no experimento

4.1. Resultado do raw data

A primeira informação que extraímos do experimento foram os dados não processados também conhecidos por “*raw data*”, com estes dados conseguimos observar algumas estruturas e formas do ambiente proposto, como também pessoas presentes no ambiente. Para exibição dos pontos adquiridos no experimento foi utilizado o software Matlab para plotagem do arquivo de pontos, gerado na máquina virtual através de um algoritmo em C, e geradas as figuras 5 e 6. A figura gerada possui aproximadamente 540.000 pontos no espaço sendo composto por x,y,z . Neste ponto do trabalho já foi observado a grande dificuldade em trabalhar com esta quantidade elevada de pontos em uma distância relativamente pequena no contexto da robótica. O processamento exigido para trabalhar com os pontos do “*raw data*” já foi elevado, como ainda não havia reduzido a quantidade de pontos, tarefas simples como, rotacionar os pontos

demandou por um processamento altíssimo no software Matlab. No trabalho o Matlab foi utilizado para exibição/plotagem dos pontos não realizando tarefas de processamento matemático.

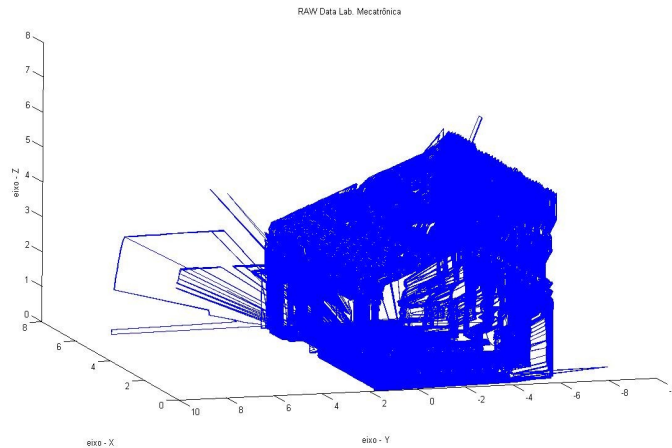


Figura 5: Imagem gerada com o software Matlab a partir do raw data

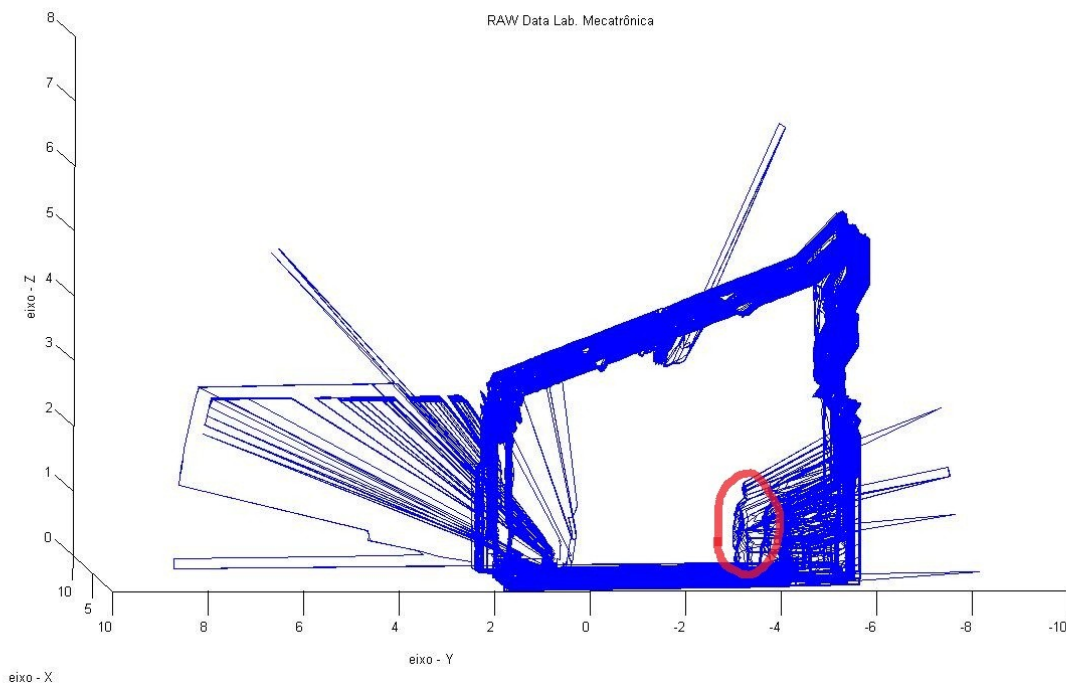


Figura 6: No destaque da imagem, há a forma de uma pessoa presente no experimento.

4.2. Algoritmo para mapeamento 3D

Devido a dificuldade para trabalhar com os dados, foi implementado um algoritmo utilizando técnicas probabilísticas, que são propostas por Thrun et al (2005), para um refinamento dos dados obtidos no experimento. Esta técnica baseia-se no princípio de aumentar a crença onde o laser atinge e reduzir a crença para as células que antecedem a célula de maior distância atingida, a redução dos pontos acontece devido a redundância dos dados que é eliminada.

Tabela 1. Algoritmo utilizado para eliminar as redundâncias.

Algoritmo atualiza_mapa(posição_atual, leitura_laser)

```

para posição_atual
  Se crença_da_célula[leitura_posição_atual] < MAX_CRENÇA
    Aumenta crença_da_célula
  Enquanto leitura_posição_atual > posição_atual
    leitura_posição_atual – TAMANHO_CELULA
  Se leitura_posição_anterior != leitura_posição_atual
    Diminui crença da célula
  
```

Nosso algoritmo mostrado na tabela (1) e (2), obteve um bom resultado na redução dos pontos coletados mantendo a qualidade e armazenando pontos de real interesse e reduzindo o erro na leitura do sensor laser. Cada ponto armazenado pelo nosso algoritmo equivale a uma área de 0,001 m³ ou a uma célula quadrada com dimensões iguais a 10cm. A quantidade de pontos iniciais era 540.000 pontos, após o uso de nosso algoritmo tivemos uma redução para 23.865 pontos correspondendo a um pouco menos que 5% dos pontos iniciais. O resultado é apresentado na figura (8).

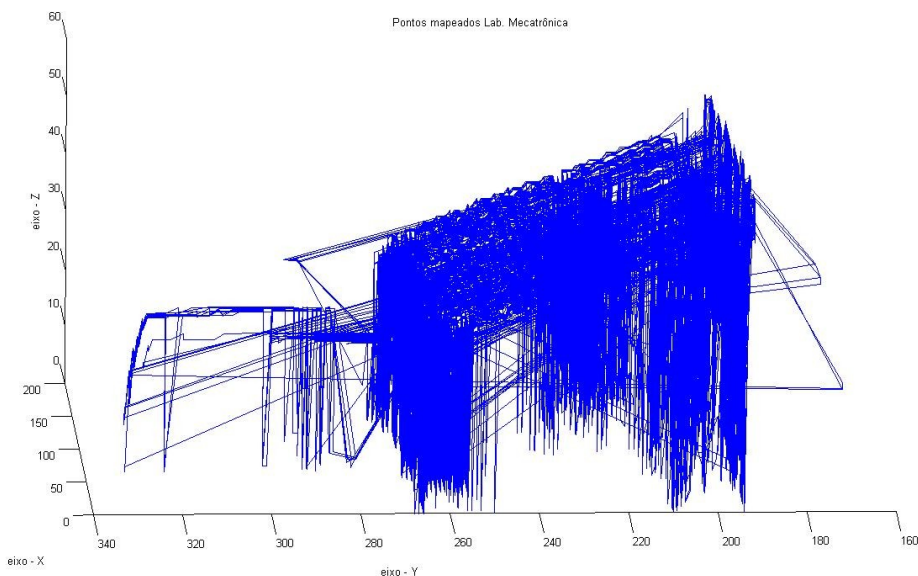


Figura 7: Imagem gerada com o software Matlab a partir dos dados processados pelo algoritmo de mapeamento

Tabela 2. Algoritmo reconstruir o mapa.

```

Algoritmo gera_mapa( )
  posição_atual = 0;
  para todas_células_do_mapa
    Se crença_da_célula[posição_atual] >= 75%
      Salva posição_atual
  
```

Para uma melhor visualização dos pontos adquiridos optamos pelo uso do software VRMLVIEW, onde exportamos os pontos obtidos para um código VRML simples e geramos as figuras com melhor detalhamento do ambiente, mostrado nas figuras (8), (9), (10) e (11).

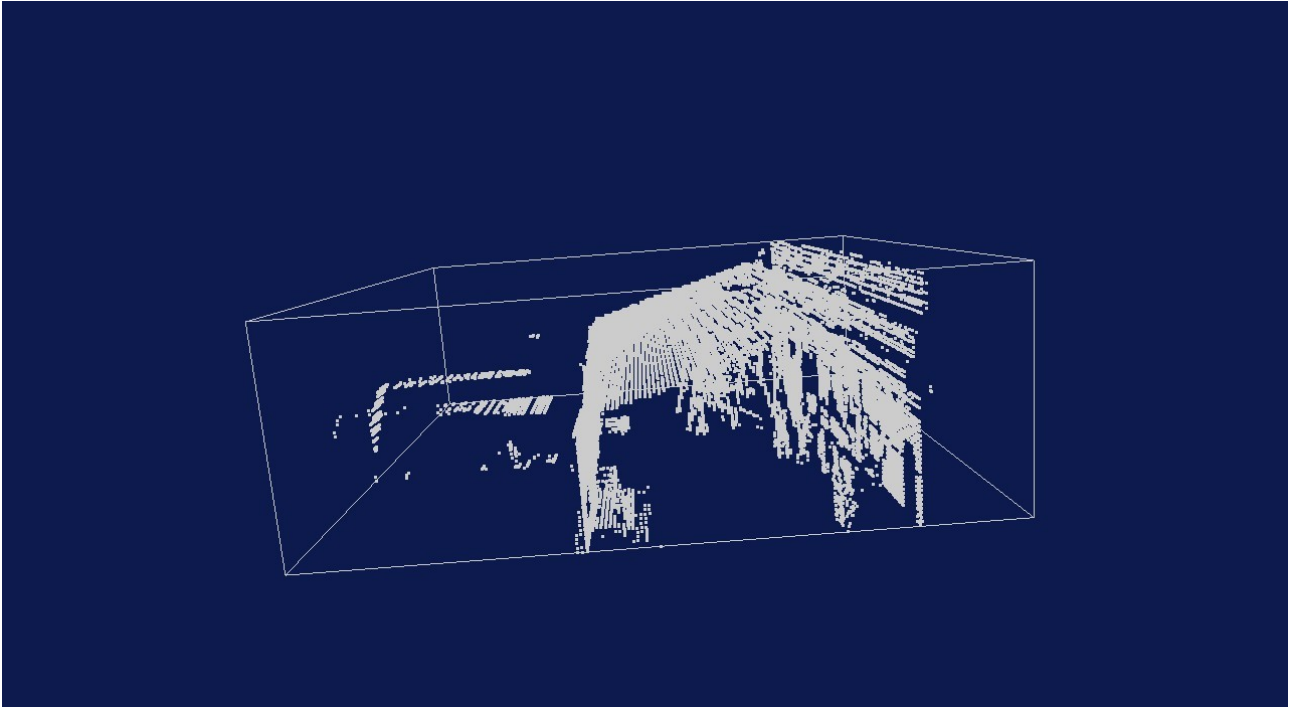


Figura 8: Imagem gerada com o software VRMLVIEW, pessoas no detalhe

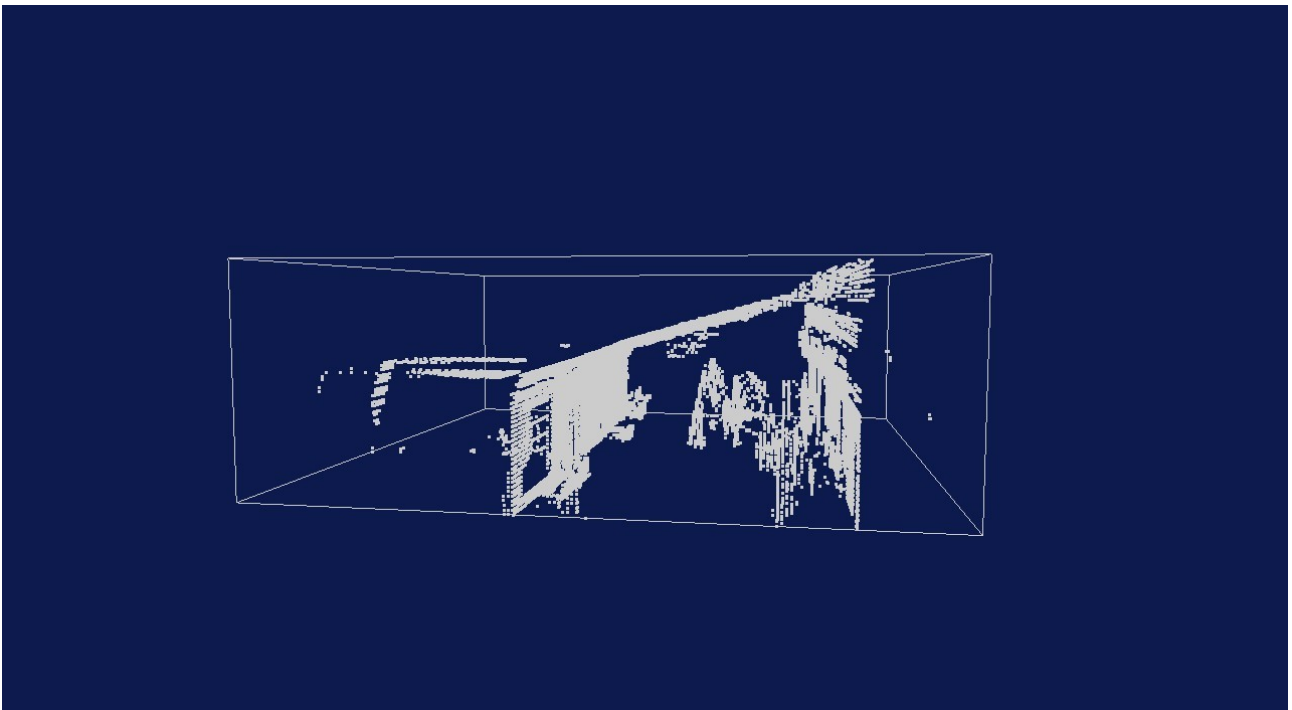


Figura 9: Estruturas verticais reconstruídas com o algoritmo de mapeamento.

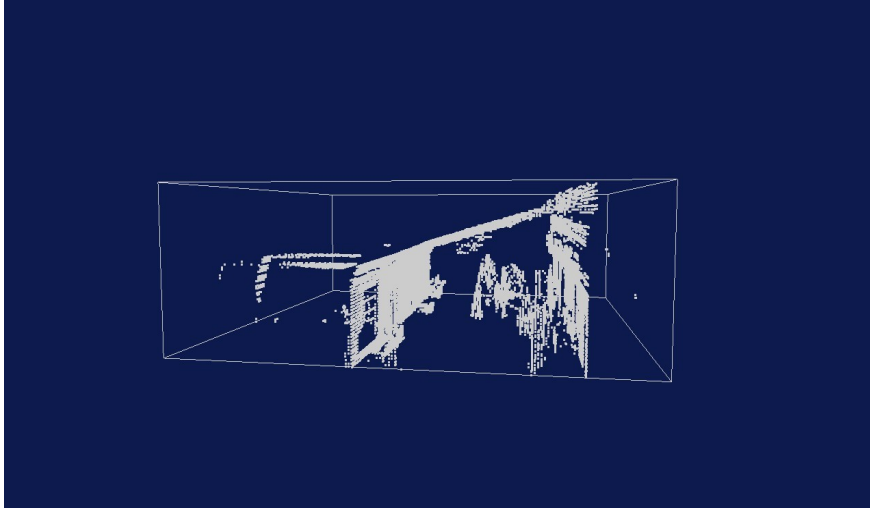


Figura 10: Imagem lateral do ambiente, observa-se os ventiladores no teto.

4.3. Algoritmo para mapeamento 2D

Com a possibilidade de repetir o experimento inúmeras vezes e um ambiente para simular e implementar diversos novos testes, criamos um algoritmo para mapeamento 2d, este mapa foi criado a partir das coordenadas x obtida da odometria e k obtida do sensor laser. Foi considerado nesta construção um y não variante o qual percebemos o aparecimento de alguns erros na leituras como mostrado na figura 12. Para implementação do algoritmo e extrair o mapa 2D, foram consideradas as leituras do sensor laser iniciando no ângulo 0 a 45 graus e de 135 a 180 graus, o posicionamento do sensor laser foi o mesmo apresentado na figura 3. No mapa 2D também foi possível observar o quanto de memória é necessário para representar uma área de $50 \times 50 \times 50 \text{ m}^3$. em nosso experimento utilizamos a representação com ponteiros, as áreas da figura que são apresentadas em verde são referentes a objetos fixos, como paredes, caixas e etc. A área em vermelho é referente ao trajeto percorrido pelo robô. E as regiões em preto são locais sem objetos ou sem explorar. A geração da imagem do mapa 2D foi criada com a utilização da biblioteca de programação openCV.

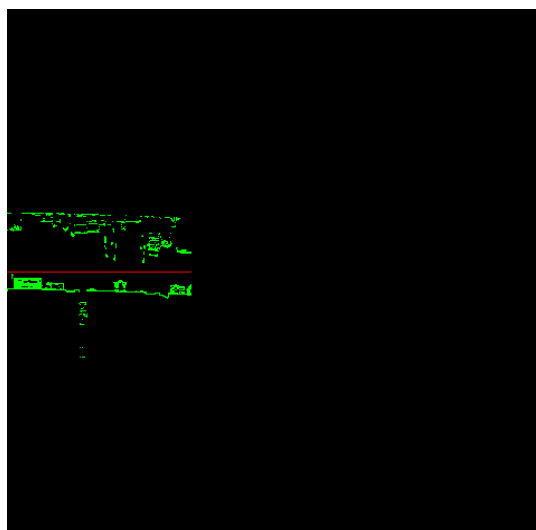


Figura 11: Mapa 2D extraído com o experimento.

Isso fez com que trabalhos futuros possam ser realizados. Entre os mais importantes podemos citar, implementações de rotação e “loop” fechado.

5. CONCLUSÕES E COLABORAÇÃO DO TRABALHO

Para criação de robôs móveis várias tarefas devem ser realizadas, pois o sistema como um todo é composto por componentes que realizam tarefas específicas como, planejamento de rota, navegação, localização e mapeamento. Este trabalho colabora com a implementação de algoritmos para mapeamento utilizados em robôs móveis, foi realizado um experimento em um ambiente real, o experimento pode ser utilizado pelo simulador inúmeras vezes, tendo a possibilidade de depuração do algoritmos, estudo de situações, base de desenvolvimento para algoritmos de robótica móvel, sem a necessidade de realizar novos experimentos, possibilitando o uso do mesmo “dataset” por vários pesquisadores.

O estudo mostrou a redução nos dados coletados e processados pelo algoritmo de mapeamento, mostrando que as técnicas probabilísticas utilizadas reduzem a incerteza sobre objetos observados pelo robô, eliminando a redundância dos dados obtidos e gerando um mapa com erros, os erros apresentados são gerados pela incerteza de posicionamento, tanto angular com relação a origem, como erros relacionados com o ponto referencial de observação, com estas informações nossa pesquisa será direcionada para estudos de localização e mapeamento simultâneo (SLAM), com estimativas de posicionamento, iremos gerar um mapa mais preciso e com correções de erros.

6. TRABALHOS FUTUROS

Com a observação dos erros, houve a motivação para implementação de um algoritmo para SLAM, pois como encontrado em diversos artigos publicados, para que haja um mapa preciso precisamos ter uma localização precisa do robô no ambiente onde esta inserido. Além de ser possível distinguir objetos fixos e móveis. Este algoritmo será parte da implementação cujo objetivo são mapas multi camadas de superfície e já serão realizadas implementações utilizando este método.

7. AGRADECIMENTOS

Agradecimentos a equipe de pesquisa do ICMC pelo suporte prestado na utilização do software Player, em especial ao Alberto Hata no auxílio do experimento e ao Leandro Carlos Fernandes. A equipe de pesquisa do laboratório de mecatrônica pelo suporte, em especial ao Valdinei Belini e ao Leonardo Pedro por apresentar alternativas para exibição e fornecimento de softwares auxiliares para análises dos dados do experimento.

8. REFERÊNCIAS

- Bares J. Hebert M., Kanade T., Krotkov E., Mitchell T., Simmons R. e Whittaker W. R. L., 1989, “Ambler: An autonomous rover for planetary exploration” IEEE Computer Society Press, 22(6):18–22.
- Elfes, A., 1987, “Sonar-based real-world mapping and navigation.” IEEE Journal of Robotics and Automation, VOL. RA-3 N° 3:p. 249–265, Junho.
- Hebert M., Caillas C., Krotkov E., Kweon I.S. e Kanade T., 1989, “Terrain mapping for aroving planetary explorer.” In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), pages 997–1002.
- Kümmerle R., Triebel R., Pfaff P., Burgard W., 2007, “Monte Carlo Localization in Outdoor Terrains using Multi-Level Surface Maps” In Proc. of the 6th International Conference on Field and Service Robotics (FSR 2007), Chamonix, France.
- Lamon P., Stachniss C., Triebel R., Pfaff P., Plagemann C., Grisetti G., Kolski S., Burgard W., Siegwart R., 2006, “Mapping with an Autonomous Car.” In Proc of the Workshop on Safe Navigation in Open and Dynamic Environments, IEEE International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China.



VI CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA
VI NATIONAL CONGRESS OF MECHANICAL ENGINEERING
18 a 21 de agosto de 2010 – Campina Grande – Paraíba - Brasil
August 18 – 21, 2010 – Campina Grande – Paraíba – Brazil

- Pfaff P., Burgard W., 2005, “An Efficient Extension of Elevation Maps for Outdoor Terrain Mapping”, In Proc. of the Int. Conf. on Field and Service Robotics (FSR), p. 165-176.
- Pfaff P., Triebel R., Stachniss C., Lamon P., Burgard W., Siegwart R., 2007, “Towards Mapping of Cities.” In Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA).
- Thrun S., 2002, “Robotic mapping: A survey.” In G. Lakemeyer and B. Nebel, editors, Exploring Artificial Intelligence in the New Millennium, Morgan Kaufmann.
- Thrun S., Burgard W., Fox, D., 2005, “Probabilistic Robotics”, The MIT Press
- Triebel R., Pfaff P., Burgard W., 2006, “Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing” In Proc. of the International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China.

9. DIREITOS AUTORAIS

MAPPING INDOOR ENVIRONMENTS USING MULTI LEVEL SURFACE MAPS

Antônio Baptista Júnior, bodninja@sc.usp.br¹
Jorge Pablo Moraga Galdames, jorge.pablo@gmail.com¹
Marcelo Becker, becker@sc.usp.br¹
Glauco Augusto de Paula Caurin, gcaurin@sc.usp.br¹
Denis Fernando Wolf, denis@icmc.usp.br²

¹ Escola de Engenharia de São Carlos – USP
Departamento de Engenharia Mecânica
Av. Trabalhador São-Carlense, 400
CEP 13566-590 - São Carlos - SP

² Instituto de Ciências Matemáticas e de Computação – USP
Av. Trabalhador São-carlense, 400 – Centro
Caixa Postal: 668 - CEP: 13560-970 - São Carlos - SP

Abstract: *The task of mapping an environment becomes crucial in mobile robotics because robots need information from the environment to perform tasks of high difficulty level. Here we show a way for representation of 3D maps that can be used for route planning and location. To represent the map used in the mapping grid discrete already known in robotics. But for us to map 3D environments made the changes that are proposed in the statements of multi level surface, this proposal holds multiple surface in each grid "cell". This task was possible with the use of a robot ActiveMedia Pioneer equipped with odometry and laser sensor. The data acquired by the robot are processed by a mapping algorithm in grid which was observed the need for future implementation of a SLAM algorithm, implemented with Kalman filter. With these methods can represent an environment indoor accurately and storing a small amount of information. In this approach the representation of vertical structures, such as doors, windows and ramps are possible. The experiment environment was indoor. Logs were created robot navigation using the software simulator Player so that we could repeat the experiment without the need of the robot and being available for future work.*

Keywords: *robotic mapping, occupancy grid mapping, player*

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído no seu trabalho.