

Segmentação em Nuvem de Pontos para Detecção de Superfícies Aquáticas

Valmir Ferreira Segundo Neto, Luiz Marcos Garcia Gonçalves
Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte - UFRN
Natal, Brasil
Email: valmirf2n@gmail.com, lmarcos@dca.ufrn.br

Rafael Beserra Gomes
Departamento de Informática e Matemática Aplicada
Universidade Federal do Rio Grande do Norte - UFRN
Natal, Brasil
Email: rafaelbg@dimap.ufrn.br

Segmentação em Nuvem de Pontos para Detecção de Superfícies Aquáticas

Abstract—The water surface detection is a fundamental application for decision-making and control of autonomous boats, and because of it, it is possible to define the navigable and the non-navigable areas of a specific place, helping to prevent possible accidents involving the autonomous vehicle, besides that, it can facilitate some significant activities, like the water monitoring. Utilizing point clouds from a stereo camera, the present work proposes an application that uses the logic from the RANDOM SAMple Consensus algorithm for plane detection applied to the water surface detection, and it also splits the application in three main steps: The image database acquisition, image enhancement and the segmentation. The results will be presented comparing the original image captured, the depth image and the point cloud obtained after the main algorithms conclusion with their respective runtime to evaluate the application efficiency.

Keywords—Plane Detection; Water Surface; Point Cloud; Stereo Camera; Computer Vision;

Resumo—A detecção de superfícies aquáticas é uma aplicação muito importante para o controle e tomada de decisões de embarcações autônomas, pois através dela é possível definir a área navegável da área não-navegável de um determinado lugar, ajudando na prevenção de possíveis acidentes envolvendo o veículo autônomo, além de facilitar certas atividades importantes como o monitoramento aquático. Utilizando nuvens de pontos capturadas por uma câmera estereó, o presente trabalho propõe a aplicação da lógica dos algoritmos de detecção de planos através do algoritmo RANDOM SAMple Consensus aplicado à detecção de superfícies aquáticas, além de dividir a aplicação em três passos principais: Aquisição do banco de imagens, pré-processamento e segmentação. Os resultados serão apresentados comparando a imagem original capturada, a imagem de profundidade e a nuvem de pontos obtidas após a conclusão dos algoritmos principais junto com seu tempo de processamento para avaliarmos a eficiência da aplicação.

Keywords—Detecção de Plano; Superfície Aquática; Nuvem de Pontos; Câmera Estéreo; Visão Computacional;

I. INTRODUÇÃO

Em decorrência do aumento das pesquisas relacionadas a criação de veículos autônomos, muitos centros de pesquisa e empresas estão investindo intensamente em pesquisas e desenvolvimento de veículos aquáticos autônomos, sejam eles Veículos Subaquáticos Autônomos (AUV) ou Veículos de Superfície Autônomos (ASV) de curto alcance que utilizam algum tipo de motor para sua locomoção. Desse modo, ASVs

de longo alcance, como veleiros, recebem um investimento bem menor em relação aos outros tipos de veículos. Isso se deve ao fato de apresentarem uma complexidade elevada para serem realmente funcionais pois, uma vez que não possuem motores, necessitam do vento para sua locomoção e, caso não seja autônomo, do esforço humano constante.

Contribuições: O presente trabalho propõe uma solução para a detecção de superfícies aquáticas em embarcações autônomas utilizando nuvens de pontos capturadas por sensores RGB-D, as quais são pré-processadas e, posteriormente, processadas através do algoritmo *RANdom SAMple Consensus*[1]. O resultado é a separação na nuvem de pontos entre a área navegável e a área não-navegável, o que auxilia a embarcação autônoma no seu deslocamento nas águas.

Os testes foram realizados utilizando imagens de uma piscina, pois durante o desenvolvimento do sistema o veleiro ainda não estava pronto para testes. O sistema apresentou resultados satisfatórios na detecção da superfície aquática em vários ângulos.

II. TRABALHOS RELACIONADOS

Superfícies aquáticas encontradas em águas calmas podem ser consideradas aproximadamente planas por não possuírem a formação de ondas constantes ou outros fatores que possam mudar sua forma de uma maneira abrupta. Por se tratarem de uma superfície relativamente plana, algoritmos relacionados a detecção de planos foram pensados na utilização deste trabalho.

Schnabel et al. propuseram desenvolver um sistema eficiente capaz de detectar bem as formas geométricas como planos, esferas, cilindros, cones e anéis em nuvens de pontos usando o RANSAC, pois se trata de um algoritmo que lida bem com grandes nuvens de pontos apresentando muitos dados espúrios, apesar de ter como pontos negativos sua falta de eficiência e seu grande consumo de memória [2].

Além do RANSAC, outro método bastante utilizado para a detecção de planos em nuvens de pontos é a transformada de Hough. Com isso em mente, Boormann et al. descreveram uma avaliação de diferentes variações da transformada de Hough aplicadas à detecção de planos em nuvens de pontos [3].

Tarsha-Kurdi et al. desenvolveram um algoritmo para detecção de tetos planos em prédios usando LiDAR [4]. Além da

deteção de planos, o trabalho demonstra uma comparação entre o RANSAC e a transformada de Hough na mesma aplicação. Após as comparações, concluíram que em termos de tempo de processamento e sensibilidade a nuvens de pontos o algoritmo RANSAC é ainda mais eficiente [5].

Rankin e Matthies propuseram um algoritmo de identificação de superfícies aquáticas através da variação de cores durante o dia para ser usado em veículos terrestres autônomos [6].

Badino propôs uma abordagem de detecção de planos envolvendo o método dos mínimos quadrados em imagens de disparidade [7], e segundo o autor, os parâmetros do plano no espaço euclidiano podem ser obtidos da imagem de disparidade a partir de uma simples transformação linear.

III. SISTEMA PROPOSTO PARA DETECÇÃO DE PLANO EM NUVEM DE PONTOS

O sistema proposto é dividido pensando em três passos essenciais: aquisição, pré-processamento e segmentação.

A. Aquisição

Fase inicial e uma das mais importantes do desenvolvimento do sistema proposto: determina as diretrizes da aquisição das imagens/nuvens de pontos, o hardware utilizado e o algoritmo inicial de interação com a câmera para a obtenção do seu controle pleno.

B. Pré-processamento

A fase de pré-processamento no projeto proposto possui a função de retirar uma grande quantidade de pontos espúrios das nuvens de pontos para fazer com que o algoritmo de detecção de superfícies aquáticas seja mais acurado. A lista a seguir descreve os dois passos desta etapa:

- Filtragem na nuvem de pontos através de um paralelepípedo: todos os pontos fora do volume são descartados (*crop*). Esse paralelepípedo é posicionado de forma a descartar a máxima quantidade de pontos que não pertencem à superfície aquática;
- Subamostragem da nuvem de pontos para redução do tempo de processamento;

C. Segmentação

A fase de segmentação no desenvolvimento do algoritmo tem como principal função a extração da superfície aquática de todo o resto das nuvens de pontos. Os seguintes passos explicam como funciona a segmentação:

- Entrada da nuvem de pontos;
- Começo das iterações do algoritmo do RANSAC;
- Escolha dos pontos aleatórios que possivelmente representam bem o modelo apropriado para um plano;
- Utilização e resolução do sistema de regressão linear 3D a partir dos pontos randomicamente selecionados para detectar o plano;
- Verificar se os pontos processados correspondem bem ao limite de distância dos *outliers*;

- Verificar se o modelo encontrado é melhor do que o modelo anterior;
- Separar o melhor modelo encontrado em uma nuvem de pontos apresentando apenas a superfície plana identificada.

IV. IMPLEMENTAÇÃO

Para implementar o sistema proposto, algumas ferramentas de software e hardware foram essenciais para o desenvolvimento dos algoritmos, dentre as ferramentas mais importantes podemos citar: Jetson TK1, Câmera ZED, *Point Cloud Library* (PCL), OpenCV (*Open Source Computer Vision Library*).

A. NVIDIA Jetson TK1

A NVIDIA Jetson TK1 possui a seguinte especificação: GPU que utiliza a tecnologia NVIDIA *Kepler* com 192 núcleos CUDA (*Compute Unified Device Architecture*) capazes de entregar até 300 *gigaflops*, uma CPU com arquitetura ARM e 4 núcleos de processamento com 2.33GHz, 2GB de DRAM DDR3L de 933MHz e uma memória de armazenamento eMMC de 16GB. Todas essas especificações do hardware fazem da NVIDIA Jetson TK1 uma boa placa para aplicações de visão computacional embarcadas [8].

B. Câmera ZED

A câmera ZED [9], desenvolvida pela Stereolabs, foi utilizada por três motivos principais. Primeiro, cumpre os requisitos do projeto, uma vez que ela é baseada em CUDA. Segundo, foi criada com a intenção de ser usada em ambientes externos, possuindo um alcance de 20 metros. Por fim, captura as imagens em alta definição, diminuindo assim a quantidade de ruídos gerados pela captura das imagens.

C. Point Cloud Library

A PCL (*Point Cloud Library*) foi uma biblioteca fundamental para o desenvolvimento do projeto proposto, dado que possui como função principal a manipulação de nuvens de pontos, além de ser bem otimizada para processadores gráficos da NVIDIA e é *open source* [10].

D. OpenCV

O OpenCV possui módulos de processamento de imagens e vídeos utilizando periféricos ou não, estrutura de dados, álgebra linear e interface gráfica do usuário básica, além de centenas de algoritmos de visão computacional, como: Filtros de imagens, calibração de câmera, reconhecimento de objetos, análise estrutural e outros [11].

V. EXPERIMENTOS

Uma vez que a embarcação ainda não possui autonomia, os experimentos não puderam ser realizados usando o veleiro. Dessa forma, o sistema foi posicionado próximo a uma piscina para a realização dos experimentos.

Em uma primeira etapa, os parâmetros relativos às etapas de pré-processamento e segmentação foram estabelecidos de forma a ter o melhor resultado possível. Uma vez que a definição de uma métrica para a qualidade dos resultados da

segmentação não foi definida, essa etapa foi feita através de inspeção visual. Os parâmetros definidos constam na Tabela I.

Paralelepípedo	X(-5000, 500), Y(-2000, 600), Z(0, 20000)
Subamostragem	8
RANSAC	Distância = 3800; Modelo Mínimo = 8; Iterações = 200

Tabela I: Parâmetros utilizados no experimento

Em seguida, o tempo de processamento das etapas de pré-processamento e segmentação foram medidos para 13 nuvens de pontos diferentes.

VI. RESULTADOS E DISCUSSÃO

As Figuras 1 e 2 exibem imagens da captura e o resultado da segmentação para as resoluções de, respectivamente, 480p e 720p.

A Tabela II exhibe o tempo médio do processamento das amostras em suas resoluções.

	Pré-processamento	Segmentação
480p	69481538ms	74929231ms
720p	380067692ms	459799231ms

Tabela II: Tempo de processamento médio para as etapas de pré-processamento e segmentação

A. Qualidade dos resultados

Dado que o problema de segmentação é mal-posto, pretende-se ainda neste projeto realizar uma segmentação manual por vários usuários que devem compor uma base de dados para aferir a qualidade da segmentação.

Os resultados parciais, entretanto, são reportados pelos pesquisadores envolvidos como apresentando um modelo de superfície aquática visualmente próximo do modelo real.

B. Melhorias nos resultados

Como foi dito anteriormente, o trabalho desempenha bem sua função de detecção de superfícies aquáticas, mas por conta do tempo de processamento dos testes, o sistema ainda precisa de algumas otimizações para melhorarmos mais ainda o tempo de processamento, fazendo com que a aplicação seja eficiente em tempo real; o objetivo é atingir uma taxa de 30 quadros por segundo.

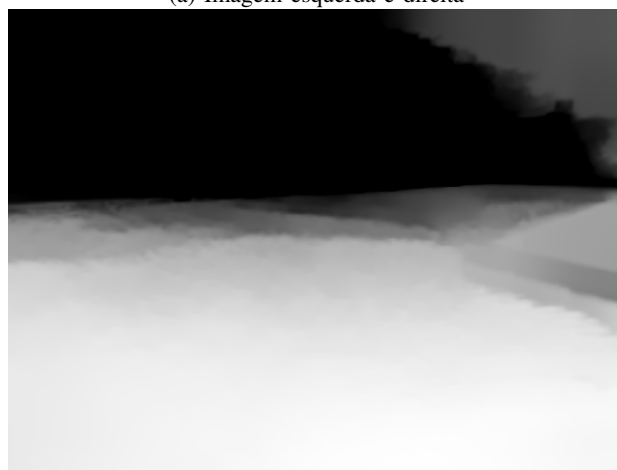
Uma vez que é um dos nossos objetivos, ter uma base de dados, pretende-se realizar um ajuste fino nos parâmetros em função do desempenho do algoritmo nessa base de dados.

VII. CONCLUSÃO

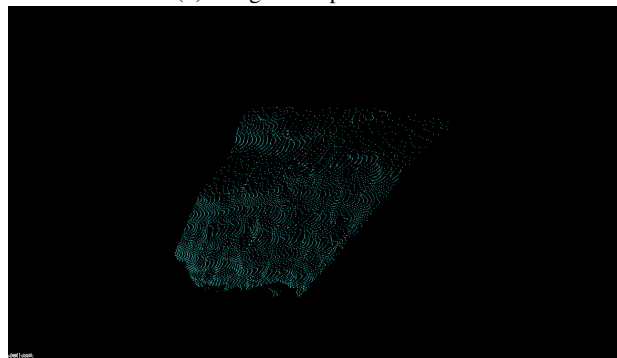
Através do aumento das pesquisas e desenvolvimento envolvendo veículos autônomos, sejam eles veículos aéreos, veículos terrestres ou embarcações, como o barco ROAZ desenvolvido no Brasil [12], cada vez mais vemos aplicações envolvendo visão computacional e melhorias nos métodos existentes de identificação tanto de objetos como de áreas de possível navegação. Esta pesquisa propõe a criação de um sistema de detecção de superfícies aquáticas para ajudar embarcações autônomas.



(a) Imagem esquerda e direita



(b) Imagem de profundidade

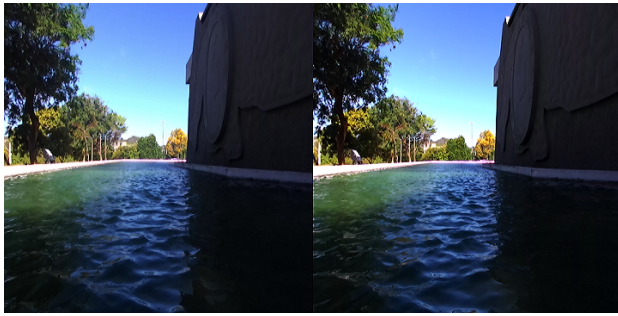


(c) Nuvem de pontos da superfície aquática

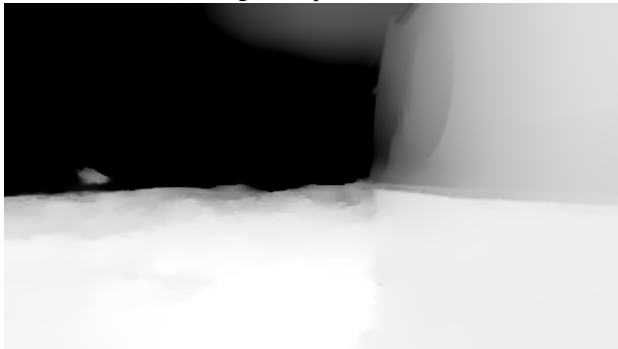
Figura 1: Aquisição e resultados na resolução 480p

Levando em conta o projeto proposto, considerou-se a utilização de um método para detecção de planos com o intuito de usá-lo para detectar superfícies aquáticas. Entre os métodos de detecção de planos mais difundidos pela comunidade de visão computacional estão a detecção de planos utilizando a transformada de Hough, o RANSAC e o método dos mínimos quadrados.

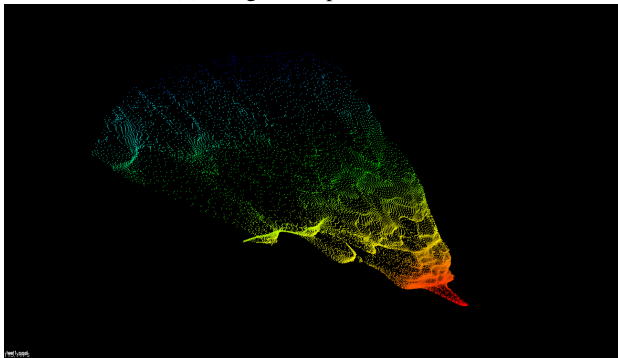
Com isso em mente, escolheu-se o RANSAC como o método núcleo de nossas aplicações por ser geralmente mais eficiente que os outros métodos citados, ser bem genérico, servindo para várias aplicações que necessitam estimação de parâmetros e pela sua intensa utilização pelos pesquisadores de projetos envolvendo a área da visão computacional, espe-



(a) Imagem esquerda e direita



(b) Imagem de profundidade



(c) Nuvem de pontos da superfície aquática

Figura 2: Aquisição e resultados na resolução 720p

cialmente em algoritmos de detecção de planos terrestres.

A partir dos resultados parciais obtidos, pode-se notar que o algoritmo desenvolvido conseguiu cumprir sua função de detecção de superfícies aquáticas, levando em conta que a maior parte dos casos observados conseguiram gerar resultados satisfatórios e sem muitos erros visuais de detecção, inclusive quando havia a presença de paredes nas aquisições, que são consideradas erros de aquisição proposital para testarmos como o pré-processamento lidaria com essa situação.

A. Etapas a seguir

Dando sucessão ao que já foi feito, apresentamos as etapas a serem realizadas nos próximos meses:

- Otimizar o algoritmo do pré-processamento, diminuindo a complexidade do algoritmo para melhorar a eficiência da aplicação como um todo, e utilizando valores mais otimizados para a retirada de dados espúrios;

- Otimizar o algoritmo de detecção de superfícies aquáticas, reduzindo sua complexidade, deixando-o próprio para aplicações em tempo real;
- Desenvolver um algoritmo de detecção de objetos nas superfícies aquáticas e suas respectivas distâncias para ajudar na tomada de decisões de alto risco para as embarcações autônomas;
- Migrar o desenvolvimento de C++ para CUDA, deixando a aplicação mais eficiente nas placas da NVIDIA.

AGRADECIMENTO

À Pró-Reitoria de Pesquisa da UFRN por me proporcionar a oportunidade de atuar na área como bolsista pelo projeto PII12712/2015.

REFERÊNCIAS

- [1] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [2] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," in *Computer graphics forum*, vol. 26, no. 2. Wiley Online Library, 2007, pp. 214–226.
- [3] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter, "The 3d hough transform for plane detection in point clouds: A review and a new accumulator design," *3D Research*, vol. 2, no. 2, pp. 1–13, 2011.
- [4] S. Sangam, "Light detection and ranging," *CMR Institute of Technology*, no. 132, 2012.
- [5] F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer, "Hough-Transform and Extended RANSAC Algorithms for Automatic Detection of 3D Building Roof Planes from Lidar Data," in *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, vol. XXXVI, Espoo, Finland, Sep. 2007, pp. 407–412. [Online]. Available: <https://halshs.archives-ouvertes.fr/halshs-00264843>
- [6] A. Rankin and L. Matthies, "Daytime water detection based on color variation," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 215–221.
- [7] B. Hernan, "Least squares estimation of a plane surface in disparity image space," 2011, [Online; accessed 20-May-2016]. [Online]. Available: http://www.lelaps.de/papers/badino_K9_repX_1.pdf
- [8] eLinux, "Jetson tk1," 2016, [Online; accessed 20-May-2016]. [Online]. Available: http://elinux.org/Jetson_TK1
- [9] Stereolabs, "Zed stereo camera," 2016, [Online; accessed 20-May-2016]. [Online]. Available: <https://www.stereolabs.com/>
- [10] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–4.
- [11] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. "O'Reilly Media, Inc.", 2008.
- [12] H. Ferreira, A. Martins, A. Dias, C. Almeida, J. Almeida, and E. Silva, "Roaz autonomous surface vehicle design and implementation," *Robótica Control, Automação, instrumentação*, 2007.