Segmentação em Imagens de Profundidade em Ambientes Controlados

Victor Cangelosi de Lima, Maurício Marengoni
Programa de Pós-Graduação em Engenharia Elétrica e Computação
Universidade Presbiteriana Mackenzie
São Paulo, Brasil
Email: cangelosilima@gmail.com, mauricio.marengoni@mackenzie.br

***Abstract*—Assistive technologies combined with computer vision techniques have important relevance to aid the navigation of the visually impaired, allowing their social inclusion and safety. This paper proposes an efficient and precise system for segmentation of depth images, originated from the Kinect sensor. The algorithm can be used to identify obstacles for navigation purpouse. The approach shows the use of graphs for segmentation avoiding costly post processing.**

***Resumo*—Tecnologias Assistivas combinadas às técnicas de Visão Computacional têm relevância importante para auxiliar a navegação dos deficientes visuais, permitindo sua inclusão social e segurança. Este artigo propõe um sistema eficiente e preciso de segmentação de imagens de profundidade, originadas do Kinect. O algoritmo poderá ser utilizado para identificar obstáculos em sistemas de navegação posteriores. A abordagem mostra o uso de grafos para segmentação, evitando pós-processamento custosos.**

I. Introdução

No presente trabalho é apresentada uma nova abordagem para segmentação de imagens de profundidade para detecção de objetos e compreensão do cenário visando auxiliar na navegação de deficientes visuais em ambientes controlados.

Guiar-se por ambientes desconhecidos é a atividade mais básica e desafiadora para um deficiente visual. Seja em áreas privadas, como quartos de hotéis ou banheiros, ou públicas, como corredores e salas de eventos. Em sua maioria, esses espaços não foram desenhados para pessoas com deficiência visual. Simples objetos como mesas, cadeiras e armários tornam-se obstáculos que podem causar acidentes. Graças ao maior poder de processamento e ao barateamento de dispositivos, novas Tecnologias Assistivas associadas à visão computacional estão sendo desenvolvidas com o foco de ajudar na locomoção utilizando diferentes tipos de segmentação, filtros e interfaces com os usuários; alguns exemplos são apresentados em [1].

Em Visão Computacional, a segmentação de imagens é uma das subáreas mais antigas e estudadas [2, p. 269], sendo o principal problema abordado ou fazendo parte do sistema de análise de imagens e de reconhecimento de padrões principalmente no pré-processamento. Para traçar o caminho livre em um sistema de navegação baseado em visão é possível separando objetos à frente que são obstáculos e identificando locais possíveis para locomoção. Através do Kinect e sua representação do ambiente com as imagens RGB-D; sendo o RGB um dos padrões de cor utilizado para representar uma imagem colorida composto pelas intensidades das cores: de vermelho (Red), verde (Green) e azul (Blue). O D (Depth) representa uma imagem em profundidade onde a intensidade de cada pixel representa a distância entre os objetos e o sensor. O sistema proposto mapeará as distâncias e orientações para segmentar uma imagem de profundidade e localizar mesas, cadeiras e outros objetos relevantes em ambientes controlados com acurácia e eficiência, esta alcançada evitando pós-processamentos que são muito comuns em técnicas de segmentação.

Este trabalho está organizado da seguinte forma: a seção II apresenta trabalhos que discutem sobre técnicas de segmentação utilizando imagens de profundidade; a seção III apresenta o desenvolvimento da aplicação, detalhando as técnicas utilizadas e sua forma de funcionamento; a seção IV apresenta os resultados obtidos, analisando os testes realizados; a seção V apresenta as devidas conclusões, elencando as vantagens e desvantagens do sistema, além das propostas de desenvolvimento futuro, como testes com outros filtros e otimizações no algoritmo.

II. Trabalhos relacionados

A aplicação da segmentação no auxílio à navegação pode ser observada no sistema proposto por [3], que notifica o usuário sobre obstáculos através de um colete vibrotátil composto por motores organizados em uma matriz 4x4 que emitem um sinal vibratório de acordo com os objetos identificados na imagem. A segmentação se dá por um processo de simplificação e redução da resolução na imagem de profundidade. A cada iteração quatro pixels vizinhos são analisados e o valor de maior intensidade é transportado para uma matriz menor, e assim sucessivamente até alcançar o tamanho de 4x4. As informações relevantes são mantidas, como obstáculos mais próximos, devido à maior intensidade de cinza de cada pixel que representa a proximidade com o sensor.

No trabalho de [4], temos a utilização de dois métodos comuns para segmentação de imagens de profundidade: transformada de Hough [5] e RANSAC [6]. A transformada de Hough é utilizada como forma de agrupar elementos de superfícies que não são associadas a clusters co-planares. Combinado com o RANSAC, método utilizado na segmentação de nuvens de pontos 3D em regiões planas para buscar pontos que se encaixem no modelo matemático de um plano, buscas iterativas são feitas na imagem validando o vetor normal

de cada ponto da imagem até que todos os pontos sejam relacionados ao plano e todos os planos sejam encontrados.

Já em [7] o objetivo é segmentar somente as superfícies planas. Para obter mais eficiência evita-se o cálculo de vetores normais e a utilização do RANSAC. Nesta abordagem a detecção de bordas 3D é utilizada para delimitar as regiões, buscando por grandes mudanças de intensidade, bordas curvas e encontro de planos; quando todas as bordas são delimitadas, o sistema agrupa todos os segmentos lineares contidos entre duas bordas que se intersectam, assim identificando cada região segmentada.

Em [8] são apresentadas técnicas semelhantes às utilizadas nesse trabalho, e com a mesma finalidade. Como primeiro passo o filtro bilateral é aplicado na imagem de profundidade, e esta é convertida em nuvem de pontos. Levando em consideração que todos os pontos de uma superfícies possuem o mesmo vetor normal, calcula-se os vetores normais a partir da nuvem de pontos e, pixel a pixel, é analisado se o ponto pertence à uma nova região ou à região que contem algum de seus vizinhos onde o vetor normal seja o mesmo.

Diferente do modelo aqui proposto, após a etapa de segmentação dos trabalhos [4], [7] e [8] ocorre o pós-processamento onde é feita a unificação de regiões que estejam fragmentadas apesar de pertencerem ao mesmo plano, esse é um processo computacionalmente custoso, pois todas as regiões vizinhas são validadas entre si para verificar a similaridade e possível unificação.

III. Desenvolvimento do Sistema

O sistema foi desenvolvido em C++ com o auxílio da biblioteca de visão computacional OpenCV [9], e utiliza para validação do algoritmo as imagens RGB-D que fazem parte da base de dados RGB-D Scenes Dataset v.2 [10]. A segmentação da imagem de profundidade é realizada em três etapas.

A primeira delas é de filtragem de ruídos. Na imagem, devido à captura feita pelo Kinect, nota-se que para uma mesma superfície totalmente plana existe o registro de pequenas oscilações nas distâncias, como se a superfície não fosse totalmente uniforme. Assim, para efetuar a suavização da imagem sem perder o detalhe das bordas foi utilizada a técnica de filtro bilateral [11]. Diferente dos filtros espaciais, este filtro considera para o kernel, além da distância, o valor de intensidade dos pixels ao redor do ponto analisado, assim pixels que possuem diferença de intensidade maiores que seus vizinhos ficam com peso menor no cálculo do novo valor de intensidade. A Figura 1a apresenta um exemplo da filtragem na região da borda com ruído, a Figura 1b mostra o gaussiano desconsiderando a região à esquerda quando analisado um pixel na parte superior do degrau e a Figura 1c a resultante da suavização em todos os pixels ao redor de uma borda.

Na segunda parte, é realizado o cálculo dos vetores normais. Será utilizada a seguinte propriedade geométrica para essa tarefa:

- Um plano W é uma superfície bidimensional regrada por dois vetores linearmente independentes [12, p. xiii];

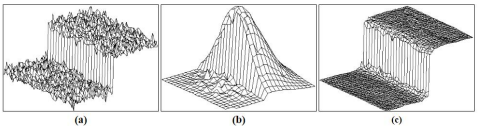
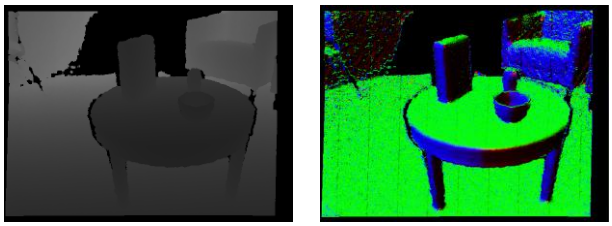


Figura 1. (a) representa a região de borda em uma imagem com ruído. (b) representa o kernel do filtro bilateral considerando os dois pesos ao analisar o pixel na região à direta da borda. (c) resultado final do filtro. Fonte [11].



(a) Imagem original de profundidade

(b) Imagem com detecção de planos

Figura 2. Detecção de planos a partir do cálculo dos vetores normais

Conseguimos obter dois vetores linearmente independentes a partir dos vetores tangencias ao ponto analisado, sendo um horizontal, diferença entre os vizinhos da direta e esquerda, e outro vertical, diferença entre os vizinhos superior e inferior. Para o cálculo do vetor normal é feito o produto vetorial das tangentes, conforme a equação abaixo:

$$\vec{P_{\mathrm{c}}} = (P_{\mathrm{u}} - P_{\mathrm{d}}) \cdot (P_{\mathrm{r}} - P_{\mathrm{l}}) \tag{1}$$

Seja P_{u} o pixel acima, P_{d} pixel abaixo, P_{r} pixel à direita, P_{l} pixel à esquerda e $\vec{P_{\mathrm{c}}}$ o vetor resultante para o pixel atual. Uma matriz de apoio será criada para armazenar os vetores normais calculados, conforme a equação 1, para cada pixel da imagem. Na Figura 2a temos o exemplo da imagem de profundidade original e a imagem segmentada em planos resultante na Figura 2b.

Para a terceira e última parte, faremos a detecção dos planos e segmentação dos objetos na imagem. Para definir se um pixel pertence ao mesmo plano que seus vizinhos verificaremos outra propriedade geométrica:

- Um vetor $\vec{n}$, sendo perpendicular a todos os vetores do plano W, inclusive ao próprio plano, é chamado de vetor normal [13, p. 127];

Seja $P_0 = (x_0, y_0, z_0)$ pertencente ao plano W, $\vec{n} = (a, b, c)$ o vetor normal do plano W, para que o $P = (x, y, z)$ pertença ao plano W, a equação deve ser:

$$\vec{n} \cdot (P - P_0) = 0 \tag{2}$$

Nota-se na figura 2 que planos que estão paralelos possuem a mesma cor devido à orientação do vetor, porém podem não estar na mesma altura. Assim durante a segmentação será necessária a verificação, além da perpendicularidade, da diferença de intensidade entre os pixels, ou seja a distância ou profundidade. Com esse critério eliminamos a possibilidade de planos vizinhos paralelos com alturas diferentes, como o

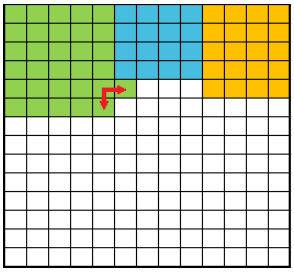
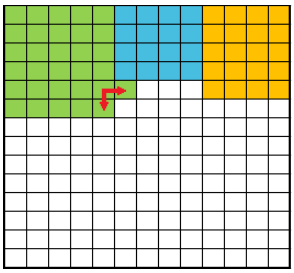


Figura 3. Varredura linha a linha feita no grafo para verificação das similaridades dos vizinhos inferior e a direita

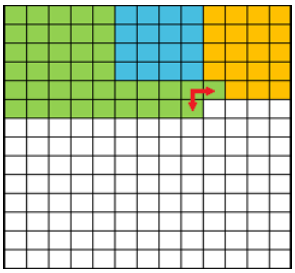
chão e a parte superior da mesa, serem considerados o mesmo objeto.

Para representar cada região segmentada teremos um grafo, onde cada elemento mapeia um pixel na imagem. Como apoio uma nova matriz, com as mesmas dimensões que a imagem original e de vetores normais, é criada e populada com cada elemento dos grafos ainda não conectados. Conforme cada pixel é identificado como parte do mesmo plano que seus vizinhos, estes serão conectados assim expandindo o grafo e região. Uma única varredura é feita pela imagem analisando cada pixel em comparação com seus vizinhos, inferior e à direita, sua diferença de intensidade e perpendicularidade do vetor normal, conforme a equação 2, e, assim, conectando os elementos quando os dois predicados são atendidos, conforme apresentado na Figura 3.

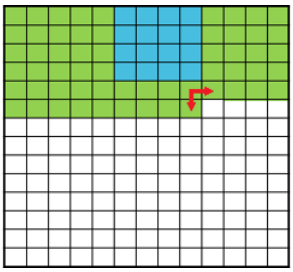
Durante o processo, a mesma região pode estar fragmentada em mais de uma sub-região, devido a alguma outra região estar sobreposta à esta, a Figura 4a mostra as duas regiões, verde e amarela, que apesar de serem iguais ainda não foram unificadas, pois nenhum pixel em comum foi analisado e conectado. Em outras técnicas isso seria resolvido em um processo posterior de fusão de regiões, porém, no algoritmo proposto, isso é solucionado pela simples conexão entre o último elemento observado da região crescente e o primeiro elemento encontrado após o fim da região sobreposta. Conforme a análise de cada pixel prossegue, uma vez que a área sobreposta é totalmente segmentada, o próximo pixel analisado será o pixel em comum com as duas regiões 4b, assim, quando for conectado, teremos a fusão das regiões devido à possibilidade de navegar para todos os elementos do grafo verde e amarelo 4c.



(a) Varredura do grafo verificando similaridades com pixels vizinhos

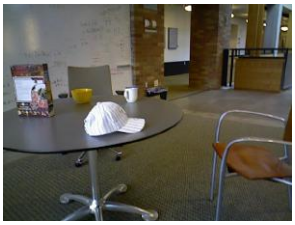


(b) Verificação de similaridade com primeiro pixel da região ainda não conectada

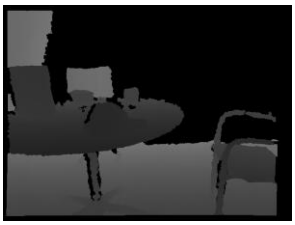


(c) Conexão entre as regiões pelo pixel analisado

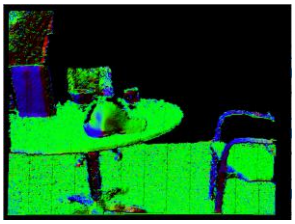
Figura 4. Representação dos grafos durante a análise de similaridade de cada pixel



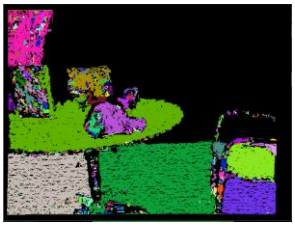
(a) Imagem original



(b) Imagem de profundidade



(c) Detecção dos planos



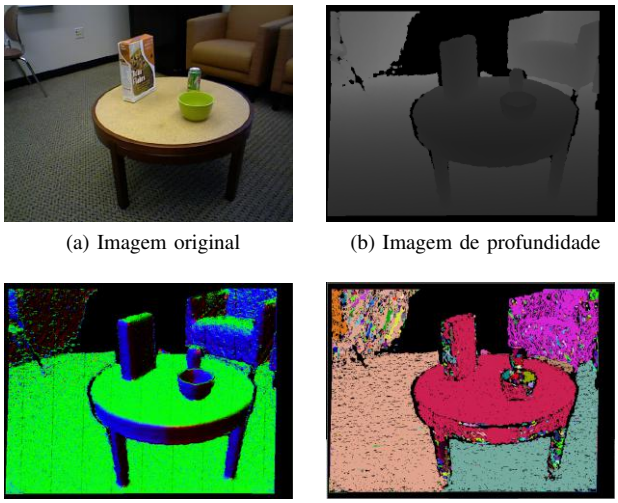
(d) Imagem segmentada

Figura 5. Exemplo de cenário segmentado

IV. Resultados

Para verificar a eficiência e acurácia do sistema foram realizados testes em dois cenários distintos. As Figuras 5a e 6a mostram o cenário trabalhado, cujos ambientes contêm obstáculos como mesas, cadeiras e poltronas em distâncias diferentes.

Inicialmente obtemos as imagens de profundidade 5b e 6b e, após a aplicação do filtro bilateral, calculamos os vetores normais para detecção dos planos, demonstrados nas Figuras 5c e 6c. Essa identificação é importante para a separação das regiões que, apesar de distintas, possuem intensidade de cinza muito próxima por estarem em contato. Para regiões como o chão e os pés das mesas utilizar somente a verificação por diferença dos valores entre os pontos dentro de um limitador seria considerado como a continuação dos mesmos objetos. Porém utilizar isoladamente os vetores normais das superfícies para segmentação pode gerar falhas e considerar planos paralelos como o mesmo objeto. Vemos nas mesmas figuras que regiões que são paralelas, como o chão e a parte superior das mesas, possuem a mesma orientação, mas não a mesma distância com o sensor. Assim sendo necessário a combinação dos dois critérios para análise, vetor normal da



(a) Imagem original (b) Imagem de profundidade

(c) Detecção dos planos (d) Imagem segmentada

Figura 6. Exemplo de cenário segmentado

superfície e diferença entre as intensidades.

Dentre os testes os maiores obstáculos nos cenários, e principais causadores de acidentes, foram separados conforme vemos nas Figuras 5d e 6d. Ao analisarmos as imagens segmentadas vemos que, ainda devido ao ruído, temos itens separados em mais de uma sub-região, como, por exemplo, o chão e os pés das mesas, e para os itens em cima da mesa temos a segmentação do boné, xícara e caixa de cereais e ainda temos objetos menores sendo considerados como parte do mesmo plano como a tigela .

O processamento de imagens com 640x480 leva em média 900 ms, sendo 300 ms para aplicação do filtro, 400ms para cálculo das normais e 200ms utilizando somente CPU e uma única thread, em um computador configurado com processador Intel Core i7 2.80GHz e 16 GB de memória RAM.

Apesar do bom desempenho na fase de segmentação, o processo como um todo ainda não pode ser utilizado em tempo real, que é o esperado em uma aplicação de navegação. O pré-processamento ainda é a parte mais custosa do processo, diferente dos demais métodos de segmentação pesquisados, que tinham como processo mais custoso a segmentação junto à unificação dos planos.

V. Conclusão

O intuito do trabalho foi identificar regiões relevantes para notificação do usuário durante a navegação, prevenindo colisões. Conseguimos identificar obstáculos maiores nos cenários testados, porém, mais testes são necessários, com a utilização de outras bases de dados e outros tipos de sensores para validar objetos menores que estejam no caminho ou desníveis no chão, que também podem ocasionar uma queda.

Os experimentos mostraram método aqui apresentado sem pós-processamentos é eficiente, mas foi penalizado durante os processos de filtragem e cálculo dos vetores normais. Ainda é considerado um sistema offline, e serão necessárias melhorias na performance. Outros filtros devem ser testados, como no trabalho apresentado por [14] a filtragem da imagem ocorre utilizando o filtro bilateral associado à uma segunda imagem para comparação melhorando a eficiência do método, bem como a possibilidade de programar um filtro especificamente para correção de ruídos gerados pelo Kinect. Já para o cálculo dos vetores normais, novas abordagens podem ser adotadas para que objetos em cima da mesa sejam segmentados com mais precisão e utilizados para trabalhos futuros como reconhecimento dos objetos.

O processamento deve ser testado com diferentes tipos de processadores, cujos poderes computacionais sejam menores. Como o objetivo é o de que o sistema seja utilizado em dispositivos móveis para auxílio à navegação, devemos levar em consideração recursos mais limitados, assim testes também devem ser feitos em placas como a plataforma Raspberry Pi [15]

Referências

[1] D. Dakopoulos and N. Bourbakis, "Wearable obstacle avoidance electronic travel aids for blind: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 1, pp. 25–35, jan 2010.

[2] R. Szeliski, *Computer Vision*. Springer London, 2011.

[3] D. Dakopoulos, S. K. Boddhu, and N. Bourbakis, "A 2d vibration array as an assistive device for visually impaired," in *2007 IEEE 7th International Symposium on BioInformatics and BioEngineering*. IEEE, oct 2007.

[4] B. Oehler, J. Stueckler, J. Welle, D. Schulz, and S. Behnke, "Efficient multi-resolution plane segmentation of 3d point clouds," in *Intelligent Robotics and Applications*. Springer Berlin Heidelberg, 2011, pp. 145–156.

[5] P. HOUGH, "Method and means for recognizing complex patterns," US Patent 3,069,654, Dec 18, 1962.

[6] 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, jun 1981.

[7] H. J. Hemmat, E. Bondarev, and P. H. N. de With, "Real-time planar segmentation of depth images: from three-dimensional edges to segmented planes," *Journal of Electronic Imaging*, vol. 24, no. 5, p. 051008, oct 2015.

[8] A. Morar, F. Moldoveanu, L. Petrescu, O. Balan, and A. Moldoveanu, "Time-consistent segmentation of indoor depth video frames," in *2017 40th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, jul 2017.

[9] "Open source computer vision library," [Online; acessado 28-Agosto-2018]. [Online]. Available: https://opencv.org/

[10] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3d scene labeling," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2014.

[11] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. Narosa Publishing House, 1998.

[12] S. Krivoshapko and V. Ivanov, *Encyclopedia of Analytical Surfaces*. Springer International Publishing, 2015.

[13] P. Winterle, *Vetores e Geometria Analitica*. Pearson, 2014.

[14] W. Zhang, B. Deng, J. Zhang, S. Bouaziz, and L. Liu, "Guided mesh normal filtering," *Computer Graphics Forum*, vol. 34, no. 7, pp. 23–34, oct 2015.

[15] "Raspberry pi," [Online; acessado 29-Setembro-2018]. [Online]. Available: https://www.raspberrypi.org/