

Detecção e modelagem de rampas para órteses ativas usando uma câmera RGB-D

Vitor G. Santos* Luís B. P. Nascimento* Diego S. Pereira*
Daniel H. S. Fernandes* Pablo J. Alsina*
Adelardo A. M. Medeiros* Márcio V. Araújo*
Raquel E. Patino-Escarcina** Dennis Barrios-Aranibar**

* Departamento de Engenharia de Computação e Automação,
Universidade Federal do Rio Grande do Norte, RN, (e-mail:
vitorgaboardi@ufrn.edu.com, lbruno@ufrn.edu.br, pablo@dca.ufrn.br).

** Departamento de Engenharia Elétrica e Eletrônica, Universidad
Católica San Pablo (UCSP), Arequipa, Peru (e-mail:
raquel.patino@gmail.com)

Abstract: This paper presents a ramp detection and modeling algorithm applied in a active orthosis. This method uses depth sensing capacities of the RGB-D cameras to segment different detected planes that integrate the scene and then performs a classification that compares the slope and width of the detected planes with accessibility specifications shown in technical standards. Simulations were made using four different ramps and it proves that the system is able to appropriately classify these structures.

Resumo: Este artigo apresenta um novo algoritmo de detecção e modelagem de rampas transponíveis aplicados a uma órtese ativa. Esse método utiliza uma câmera RGB-D que possui a capacidade de obter informações de profundidade dos objetos que estão no ambiente para segmentar diferentes planos detectados que integram a cena e então realiza a classificação comparando a inclinação e a largura dos planos encontrados com especificações de acessibilidade de normas técnicas. A metodologia foi testada por meio de simulações utilizando quatro rampas com características de inclinação e largura distintas. Foi possível constatar que a estratégia empregada foi capaz de classificar os objetos adequadamente.

Keywords: Ramp detection; Accessibility; Powered orthosis; Segmentation; RGB-D

Palavras-chaves: Detecção de rampas; Acessibilidade; Órtese ativa; Segmentação; RGB-D.

1. INTRODUÇÃO

A utilização de órteses ativas por indivíduos que possuem algum grau de deficiência motora é cada vez mais comum por conta do grau de assistência, independência e qualidade de vida que esses dispositivos podem proporcionar aos seus usuários. Exemplos como Esquenazi et al. (2012) e Arazpour et al. (2015) mostram alguns dispositivos desenvolvidos nos últimos anos que são capazes de auxiliar no processo de caminhada.

A órtese Ortholeg (Gloger et al., 2015) é um dispositivo interessante pois foi desenvolvido a partir do conceito de transparência que, segundo Melo et al. (2017), pode ser definido como a capacidade do dispositivo de fazer com que a experiência da caminhada seja a mais natural possível, tanto para o usuário quanto para as pessoas ao seu redor. Normalmente, em exoesqueletos comuns, o usuário do dispositivo é responsável por definir a melhor estratégia para transpor os obstáculos em seu caminho. Por outro lado, em uma órtese ativa transparente é desejável que exista o planejamento autônomo dos movimentos, garantindo assim um aspecto do conceito da transparência. Para que a órtese possa atender ao critério de transparência mencionado, é necessário que o sistema embarcado da órtese tenha a

capacidade de identificar obstáculos como buracos, escadas e rampas de forma que o sistema de planejamento verifique se existe um caminho seguro que o usuário possa percorrer.

Em Lamônica et al. (2008), fez-se um levantamento sobre a acessibilidade em uma universidade, onde constatou-se que 37,5% das rampas verificadas possuíam inclinações superiores às determinadas nas normas técnicas. Outros trabalhos, como Machado and Palma Lima (2015) e Calado (2006) reforçam o problema de acessibilidade de rampas em outros locais do país. Logo, existe a necessidade do desenvolvimento de um sistema de detecção de rampas que atende aos requisitos mínimos de acessibilidade.

Dentre os algoritmos de detecção de rampas encontrados na literatura, destaca-se que a maioria deles estão voltados para a navegação autônoma de veículos e robôs. Em (Gallo et al., 2008), utiliza-se uma câmera Canesta TOF para reconhecer rampas e meios-fios através da detecção de retalhos de planos usando o CC-RANSAC. Em (Hara et al., 2014), é desenvolvido uma nova ferramenta, chamada de Tohme, que detecta rampas de acesso em meios-fios a partir de imagens extraídas do Google Street View utilizando colaboração coletiva e algoritmos de visão computacional e aprendizado de máquina. Já o trabalho de Hernandez-

Aceituno et al. (2016) apresenta uma abordagem para a navegação autônoma de veículos que inclui um sensor Kinect para localizar áreas intransponíveis através da detecção e isolamento de superfícies não-horizontais, que são consideradas como obstáculos.

Embora as estratégias apresentadas sejam capazes de detectar rampas adequadamente, elas não se aplicariam a uma órtese, visto que essas metodologias não levam em consideração a inclinação, a largura e a posição da rampa em relação ao usuário para saber se é possível que uma pessoa consiga transpor essa estrutura.

Desta forma, este artigo propõe a criação de um algoritmo para reconhecimento de rampas voltado para uma órtese ativa onde classificaremos as rampas como pisáveis ou não. Essa classificação baseia-se na comparação de informações sobre a rampa detectada com especificações em normas técnicas, garantindo assim a segurança do usuário quando ele se deparar com diferentes formas de rampas. Para alcançar esse objetivo, um sensor RGB-D Kinect será colocado na cintura do usuário, fornecendo informações sobre a profundidade dos elementos que constituem o ambiente. Para testar a metodologia proposta, simulações foram realizadas considerando diferentes formas de rampas, onde resultados satisfatórios foram obtidos.

2. CONFIGURAÇÃO DO SISTEMA E AQUISIÇÃO DE DADOS

Existem diversas opções para inserir a câmera em um sistema de navegação vestível. Em Mayol-Cuevas et al. (2009), é feita uma análise extensiva sobre esse tópico. Nesse trabalho, optamos por inserir o sensor Kinect na cintura do usuário, como mostrado na Figura 1. Além disso, escolhemos a orientação da câmera tal que seja possível verificar a existência de obstáculos que estejam entre um a quatro passos do usuário.

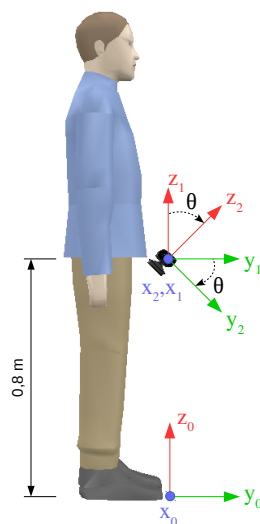


Figura 1. Posição do sensor e sistemas de referência.

Os dados de aquisição do Kinect estão representados por nuvem de pontos, que consiste em um conjunto de pontos no qual cada pixel é definido por coordenadas x , y e z que representam a localização em três dimensões do ponto em relação à câmera. Entretanto, é necessário converter essa

informação, que está no sistema de referência da câmera, em dados orientados que ajudam a conhecer a posição absoluta dos objetos do ambiente em relação ao usuário.

A Figura 1 ilustra esses sistemas de coordenadas, no qual devemos aplicar uma matriz de transformação que leve os pontos do sistema coordenada 2 para o 0, definidos por $[x_2, y_2, z_2]$ e $[x_0, y_0, z_0]$, respectivamente.

As seguintes considerações podem ser feitas referentes a mudança do sistema de coordenadas: o vetor normal de um plano horizontal deve ser paralelo ao eixo z_0 , e a origem do sistema de coordenadas 0 é colocado no chão (altura 0), e se move junto com o usuário. O eixo y_0 sempre está apontado para a frente do usuário. O ângulo de rotação de interesse é aquele correspondente à rotação no eixo x_1 , definido por θ . Os eixos x_1 e x_2 sempre estarão alinhados.

O método de estimação de rampas desenvolvido nesse trabalho é composto de duas partes principais: a segmentação de planos e a detecção de rampas. A Figura 2 mostra o fluxograma que contém as principais etapas do algoritmo, que será especificado nas próximas seções.

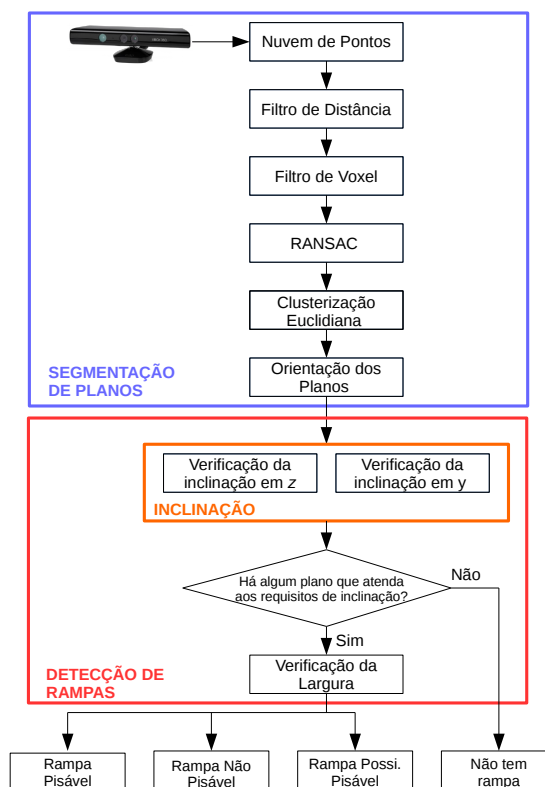


Figura 2. Fluxograma da metodologia proposta.

3. SEGMENTAÇÃO DE PLANOS

De acordo com Gallo et al. (2011), existem três abordagens distintas para a segmentação e estimação de regiões planares em uma imagem a partir de nuvens de pontos. Nesse trabalho, escolhemos uma metodologia que consiste na utilização de um estimador robusto para extrair regiões planares “dominantes” e, a partir delas, verificar a existência de rampas no ambiente.

3.1 Filtro de distância

O modelo utilizado do sensor Kinect possui um campo de visão vertical igual a 57° , fazendo com que a nuvem de pontos gerada possua informações sobre regiões que estão relativamente distantes do usuário. Dessa forma, aplicou-se um filtro de distância que elimina pontos que estão fora de um intervalo de interesse, garantindo maior confiabilidade, visto que pontos distantes tendem a ser mais suscetíveis a ruídos. Nesse trabalho, limitou-se a profundidade do campo de visão em 1,7 metros e a largura em 1,5 metros.

3.2 Filtro de Voxel

A quantidade de informação gerada pela nuvem de pontos é muito grande para ser totalmente utilizada, logo, é necessário realizar uma outra operação de filtragem. Nesse trabalho, implementou-se o filtro de Voxel, um algoritmo utilizado para diminuir a quantidade de amostras de uma nuvem de pontos, ajudando a remover ruídos e a suavizar superfícies. A implementação do filtro de Voxel cria um conjunto de pequenas caixas tridimensionais, chamadas de voxel em toda a nuvem de pontos. Então, em cada voxel, o centróide dos pontos é calculado e a nuvem de pontos resultante é aproximada pelos centróides de cada voxel.

3.3 Detecção de planos com RANSAC

O método de RANSAC (Fischler and Bolles, 1981) voltado para detecção de planos em nuvem de pontos consiste em procurar uma equação do plano que contenha a maior quantidade de pontos possíveis considerando um limiar ϵ . Os pontos que se encaixam na equação do plano são chamados de *inliers* e os que não se encaixam são chamados de *outliers*. O Algoritmo 1 resume esse procedimento.

Algoritmo 1 : Detecção de planos usando RANSAC

- 1: Selecionar aleatoriamente três pontos da nuvem.
- 2: Determinar os parâmetros do plano a partir dos três pontos selecionados.
- 3: Determinar a quantidade de *inliers* que pertencem a esse plano com uma tolerância pré-definida ϵ .
- 4: Se o número de *inliers* é maior que um limiar τ pré-definido, reestimar os parâmetros do modelo usando todos os *inliers* identificados e terminar o programa.
- 5: Se não, repetir os passos 1 até 4 (máximo de N vezes).

O algoritmo descrito encontra apenas um segmento de plano. Portanto, para que seja possível detectar múltiplos segmentos em uma nuvem de pontos, é necessário que o método de RANSAC seja executado repetidas vezes, de modo que a cada plano detectado, retira-se os *inliers* da nuvem de pontos e execute o algoritmo novamente com o restante da nuvem de pontos.

3.4 Clusterização Euclidiana

O método de RANSAC de detectar planos tem a desvantagem de não levar em consideração a distância que os *inliers* possuem entre si. Dessa forma, se existirem objetos distintos que possuam pontos em um mesmo plano, o método de RANSAC classificará esses obstáculos diferentes como se eles pertencessem a uma mesma região.

Aplicando a Clusterização Euclidiana em todos os segmentos de planos detectados, conseguimos separar em *clusters* regiões que estão longe por uma distância mínima (d_{min}), corrigindo o problema descrito. Essa metodologia utilizada é parecida com a desenvolvida em Gallo et al. (2011).

3.5 Orientação dos planos

Uma vez que o estágio de segmentação tenha concluído, é necessário que os planos detectados sejam classificados de acordo com suas orientações e suas posições relativas ao usuário. Dessa forma, para realizar o que foi disposto nesse trabalho, deveremos fazer o cálculo de dois ângulos.

Orientação em relação a z_0 : Inicialmente, calcula-se o ângulo α formado entre o vetor normal ao segmento de plano (N_p) e o vetor unitário normal ao chão que, segundo o sistema de coordenadas 0, é paralelo ao eixo z_0 e representado por $(0, 0, 1)$. A Figura 3 mostra a vista lateral de uma rampa e a representação do ângulo α .

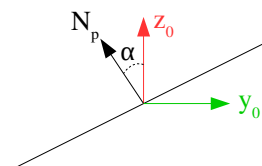


Figura 3. Vista lateral da rampa.

Esse ângulo pode ser facilmente obtido através do produto interno entre esses dois vetores, ou seja:

$$\alpha = \arccos(z_0 \cdot N_p) \quad (1)$$

Além disso, para α próximo de 0° , o segmento de plano é considerado horizontal e para α próximo de 90° , o segmento é classificado como vertical.

Orientação em relação a y_0 : É necessário também realizar o cálculo de β , que representa o ângulo entre a direção para a qual o usuário esteja orientado (y_0) e o plano sagital da rampa, conforme mostra a Figura 4.

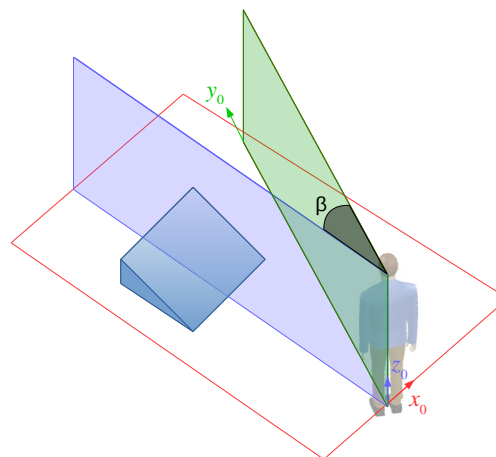


Figura 4. Representação do ângulo β .

O ângulo β será utilizado para verificar se a rampa está posicionada em frente ao usuário, garantindo que o mesmo

não suba na rampa transversalmente. Além disso, nesse trabalho ele é calculado conforme mostra a Equação 2.

$$\beta = \text{acos}(y_0 \cdot N_{p(xy)}) \quad (2)$$

onde $(N_{p(xy)})$ é a projeção no plano xy do vetor normal do segmento de plano e y_0 é o vetor unitário que aponta para a direção onde o usuário está orientado.

4. DETECÇÃO DE RAMPAS

Até o momento mostramos um algoritmo que faz apenas detecção de segmentos de planos e de algumas informações básicas, porém, precisamos realizar um estudo semântico com o objetivo de identificar se essas estruturas têm características de uma rampa e, além disso, verificar se tais rampas podem ser utilizadas por pessoas ou não.

4.1 Rampas pisáveis

A NBR 9050 estabelece critérios e parâmetros técnicos a serem observados quanto ao projeto, construção, instalação e adaptação do meio urbano e rural, e de edificações às condições de acessibilidade (ABNT, 2004). Nesse trabalho, utilizaremos essa norma como base técnica na definição do que será considerado como rampas pisáveis.

Segundo a NBR 9050, rampas acessíveis podem ter uma inclinação máxima de 8,33%, que representa um ângulo α de até 4,76°. Entretanto, quando esse sistema for implementado na prática, haverá erros de medição e aproximações que influenciarão no valor de α de cada segmento de plano detectado. Portanto, consideraremos uma tolerância máxima de 20%, ou seja, as rampas serão consideradas pisáveis (em termos de inclinação) se tiverem uma orientação de até 5,71°. Além disso, com o intuito de diferenciar rampas de planos horizontais, também será definido um limite inferior no valor de α como sendo 3°.

Outro quesito importante a ser verificado é a largura (L) de cada rampa. A NBR 9050 estabelece que a largura mínima admissível para uma rampa deve ser de 1,2 metros. Porém, essa métrica considera a passagem de duas pessoas em uma rampa. Como nosso objetivo é garantir a passagem de uma pessoa e o campo de visão do sensor RGB-D é limitado, decidimos classificar rampas como pisáveis (em termos de largura), se elas possuírem uma largura mínima de 0,7 metros.

É importante levarmos em consideração a possibilidade do usuário não estar exatamente na frente da rampa. Nesse caso, não podemos considerar a rampa como pisável, visto que o usuário não deve caminhar transversalmente sobre a rampa, uma vez que essa situação pode ocasionar acidentes. Porém, ao mesmo tempo, não podemos simplesmente classificar essa estrutura como não-pisável, dado que o usuário pode se posicionar corretamente em frente da rampa e então atravessá-la. Dessa forma, caso o ângulo β seja maior do que 10° e o ângulo α esteja de acordo com os limites descritos nessa seção, a rampa será caracterizada como possivelmente pisável. Caso o valor de β seja menor que 10°, então a rampa estará de frente ao usuário e serão verificados os outros dois parâmetros (α e L). Finalmente, a Tabela 1 resume as informações descritas nessa seção no que diz respeito à classificação de rampas.

Tabela 1. Classificação das rampas com base nos parâmetros α , β e L .

Pisável	Não Pisável	Pos. Pisável
$3^\circ \leq \alpha \leq 5,7^\circ$ e $10^\circ \leq \beta$ e $L \geq 0,7m$	$\alpha > 5,7^\circ$ ou $L < 0,7m$	$3^\circ \leq \alpha \leq 5,7^\circ$ e $\beta > 10^\circ$

4.2 Algoritmo de classificação de rampas

Como descrito na seção anterior, é preciso verificar três parâmetros para realizar a classificação da rampa, sendo eles os ângulos α e β e a largura L .

Inclinações: A partir do cálculo da orientação dos planos é possível determinar se os segmentos de planos detectados podem ser caracterizados como rampas pisáveis em termos de inclinação. Para tal, basta compararmos os ângulos α e β com os limites mostrados na Tabela 1. Se esses valores forem respeitados, verificamos se a rampa possui uma largura adequada ou não.

Largura: Inicialmente, cria-se uma nova nuvem de pontos que irá conter todos os pontos dos planos que foram segmentados. Porém, nesse momento, cada ponto terá um índice que identificará a qual segmento o ponto pertence. Em seguida, calcula-se o centróide (c) de cada segmento de plano (P) detectado que atende aos requisitos de inclinação de rampas pisáveis mostrados na Tabela 1.

O próximo passo é estabelecer uma região de interesse (RDI) para cada segmento de plano P dentro da nova nuvem de pontos que seja em torno do respectivo centróide. Essa RDI armazena todos os pontos da nuvem que estejam dentro de limites nas coordenadas x , y e z , definidos por $[x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}]$. No nosso algoritmo, esses valores são definidos da seguinte forma:

$$\begin{aligned} x_{min} &= x_{cen} - 0,35m & y_{min} &= y_{Pmin} & z_{min} &= -2m \\ x_{max} &= x_{cen} + 0,35m & y_{max} &= y_{Pmax} & z_{max} &= +2m \end{aligned}$$

No qual x_{cen} é a coordenada x do centróide e y_{Pmin} e y_{Pmax} são os limites inferior e superior, respectivamente, na coordenada y do plano P . A Figura 5 ilustra algumas dessas variáveis, onde todos os pontos que possuem a mesma cor pertencem ao mesmo segmento de plano.

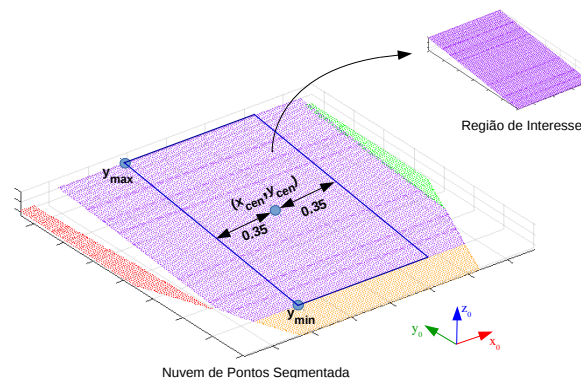


Figura 5. Nova nuvem de pontos formada e RDI.

Em seguida, vamos calcular a porcentagem p de pontos da RDI que pertencem ao segmento plano P , ou seja:

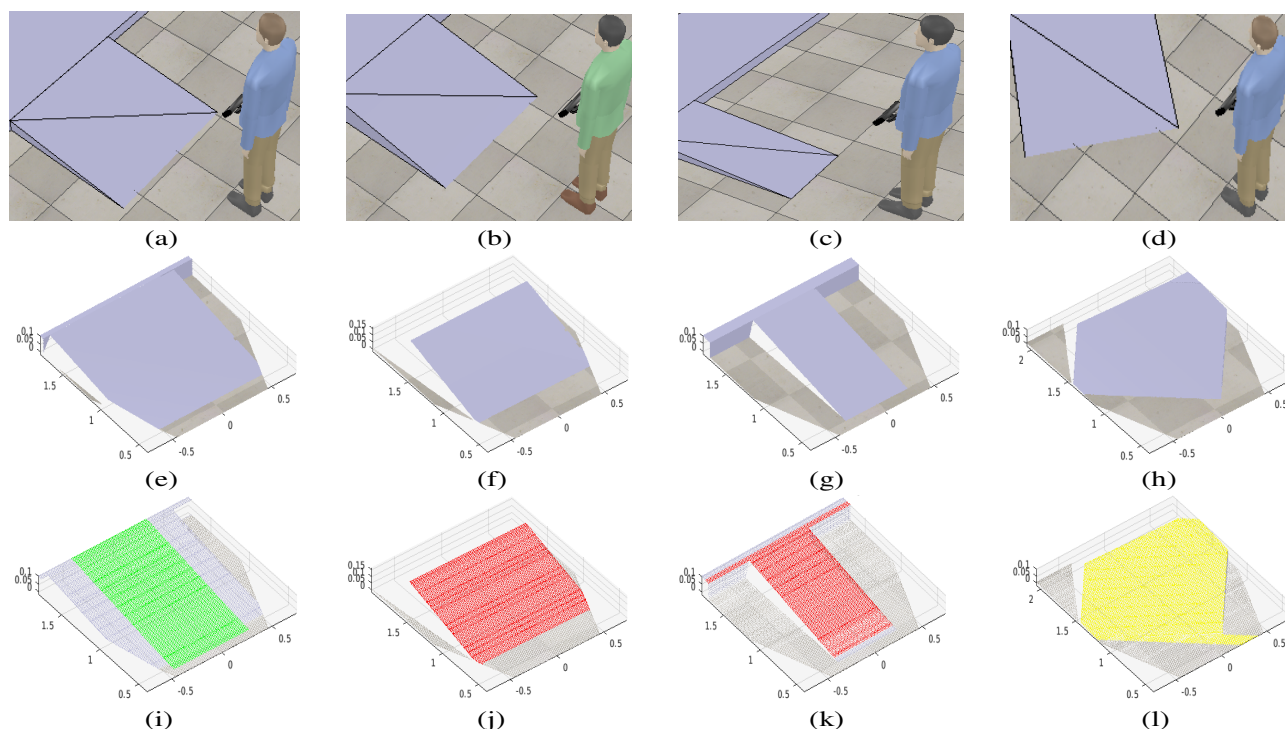


Figura 6. (a-d) Rampas criados no V-REP; (e-h) Nuvem de pontos das rampas; (i-l) Classificação das rampas.

$$p = \frac{QP_{P(RDI)}}{QP_{T(RDI)}}$$

no qual $QP_{P(RDI)}$ representa a quantidade de pontos que pertencem ao segmento de plano P que estão dentro do RDI, e $QP_{T(RDI)}$ representa a quantidade total de pontos da região de interesse.

Se o valor de p for maior que um determinado limiar σ , a maioria dos pontos do RDI pertencem ao mesmo segmento de plano P e, portanto, ele é considerado acessível no quesito de largura, visto que a largura total do RDI verificado sempre é igual a 0,7 metros (0,35 + 0,35).

No caso da Figura 5, o valor de p será igual a 1, que é o máximo possível. Logo, dizemos que essa rampa é pisável, dado que seguiu todas as restrições de inclinação e largura.

5. RESULTADOS

A fim de demonstrar o algoritmo de detecção de rampas desenvolvido, algumas simulações foram implementadas utilizando o software *Virtual Robot Experimentation Platform* (V-REP) (Rohmer et al., 2013). Além disso, utilizou-se o modelo do sensor Kinect feito por Lyall Randell e ligeiramente modificado por Nicola Piccinelli.

Os parâmetros da segmentação de planos foram definidos como segue: filtro de Voxel com tamanho 0,01 m × 0,01 m × 0,01 m, $\epsilon = 0,005$ m, $\tau = 200$ inliers, $N = 400$ e $d_{min} = 8$ cm. Na detecção de rampas, utilizou-se um limiar $\sigma = 0,95$. Além disso, a inclinação do sensor Kinect foi considerada igual a $\theta = -44^\circ$.

No V-REP, construiu-se quatro rampas que possuem as características de acordo com a Tabela 2. As rampas 1,

Tabela 2. Parâmetros das rampas simuladas no V-REP

	α ($^\circ$)	β ($^\circ$)	L (m)
Rampa 1	4,76	0	1,2
Rampa 2	9,46	0	1,2
Rampa 3	4,76	0	0,6
Rampa 4	4,76	-40	1,2

2, 3 e 4 estão representadas nas Figuras 6a, 6b, 6c e 6d, respectivamente. Além disso, as Figuras 6e-6h mostram a nuvem de pontos gerada a partir do sensor Kinect.

Analisando a Tabela 2 e comparando com os parâmetros necessários de cada classificação fornecida na Tabela 1, podemos fazer as seguintes observações:

- A rampa 1 possui uma inclinação α adequada, está localizada em frente ao usuário e tem uma largura de 1,2 metros. Portanto, deve ser classificada como pisável.
- A rampa 2 não possui uma inclinação α adequada, logo ela deve ser classificada como não pisável.
- A rampa 3 tem um ângulo α adequado e está na frente do usuário, porém possui uma largura pequena, logo deve ser classificada como não pisável.
- A rampa 4 tem valor satisfatório de α , porém está de lado do usuário visto que o ângulo β é de 40° . Dessa forma, não é analisado o valor de L e a mesma deve ser classificada como possivelmente pisável.

As Figuras 6i-6l mostram o resultado da classificação das rampas, no qual rampas pisáveis são mostradas de cor verde, rampas não pisáveis são mostradas de cor vermelha e, por fim, rampas possivelmente pisáveis são mostradas de cor amarela. Comparando os resultados com

as observações feitas, percebe-se que em todos os casos o algoritmo conseguiu classificar adequadamente as rampas.

É interessante ressaltar outros resultados sobre o programa desenvolvido: as Figuras 6k e 6l mostram que alguns pontos do chão foram considerados como pertencendo ao segmento de plano da rampa. Isso ocorreu pois a transição entre a rampa e o chão acontece de maneira suave, logo o algoritmo de RANSAC considera que esses pontos pertencem ao mesmo plano. Além disso, uma pequena variação da componente z do vetor normal ao plano gera uma alta variação no valor de α . Isso acontece devido a alta sensibilidade da função cosseno, que é utilizada na Equação 1, em valores perto de um. Dessa forma, é necessário que o algoritmo tenha uma boa precisão nos valores dos coeficientes do vetor normal ao plano.

Esses problemas não afetaram a classificação nos resultados simulados, porém podem ser relevantes ao aplicar o algoritmo no mundo real, visto que os dados são mais ruidosos.

6. CONCLUSÃO

Nesse trabalho apresentamos um módulo de percepção visual voltado para pessoas que utilizam órteses ativas transparentes e que pode ter aplicações em outras áreas, como assistência para pessoas com problemas visuais e casos de navegação autônoma. A principal contribuição dessa pesquisa é o desenvolvimento de um algoritmo de detecção e modelagem de rampas que obedecem critérios de acessibilidade baseado em normas técnicas. As simulações realizadas em um ambiente virtual mostram a eficácia do programa desenvolvido, visto que foi possível detectar e classificar rampas corretamente.

O sistema desenvolvido possui algumas desvantagens, como a possibilidade de detecção de pontos que não pertencem ao segmento de plano e a necessidade de uma alta precisão de algumas variáveis. Dessa forma, a próxima etapa a ser desenvolvida consiste na utilização da metodologia proposta em ambientes reais com o propósito de validar o que foi apresentado nas simulações. Além disso, outros trabalhos futuros envolvem a incorporação de um sensor de Unidade de Medida Inercial (IMU) para verificar a orientação do sensor RGB-D e a adição de detecção de outros obstáculos comuns, como degraus e escadas.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS

- ABNT (2004). *NBR 9050: acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos*. ABNT.
- Arazpour, M., Hutchins, S.W., and Ahmadi Bani, M. (2015). The efficacy of powered orthoses on walking in persons with paraplegia. *Prosthetics and orthotics international*, 39(2), 90–99.
- Calado, G.C. (2006). *Acessibilidade no ambiente escolar: reflexões com base no estudo de duas escolas municipais de Natal-RN*. Master's thesis, Universidade Federal do Rio Grande do Norte.
- Esquenazi, A., Talaty, M., Packel, A., and Saulino, M. (2012). The rewalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury. *American journal of physical medicine & rehabilitation*, 91(11), 911–921.
- Fischler, M.A. and Bolles, R.C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Gallo, O., Manduchi, R., and Rafii, A. (2008). Robust curb and ramp detection for safe parking using the canesta tof camera. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1–8. IEEE.
- Gallo, O., Manduchi, R., and Rafii, A. (2011). Cc-ransac: Fitting planes in the presence of multiple surfaces in range data. *Pattern Recognition Letters*, 32(3), 403–410.
- Gloger, M., Alsina, P.J., and Melo, N.B. (2015). Ortholeg 2.0-a new design for a lower limb active orthosis. In *2015 International Symposium on Micro-NanoMechatronics and Human Science (MHS)*, 1–7. IEEE.
- Hara, K., Sun, J., Moore, R., Jacobs, D., and Froehlich, J. (2014). Tohme: detecting curb ramps in google street view using crowdsourcing, computer vision, and machine learning. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 189–204. ACM.
- Hernandez-Aceituno, J., Arnay, R., Toledo, J., and Acosta, L. (2016). Using kinect on an autonomous vehicle for outdoors obstacle detection. *IEEE Sensors Journal*, 16(10), 3603–3610.
- Lamônica, D.A.C., Araújo-Filho, P., Simomelli, S.B.J., Caetano, V.L.S.B., Regina, M.R.R., and Regiani, D.M. (2008). Acessibilidade em ambiente universitário: identificação de barreiras arquitetônicas no campus da usp de bauru. *Revista brasileira de educação especial*, 14(2), 177–188.
- Machado, M.H. and Palma Lima, J. (2015). Avaliação multicritério da acessibilidade de pessoas com mobilidade reduzida: um estudo na região central de itajubá (mg). *Urbe. Revista Brasileira de Gestão Urbana*, 7(3).
- Mayol-Cuevas, W.W., Tordoff, B.J., and Murray, D.W. (2009). On the choice and placement of wearable vision sensors. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(2), 414–425.
- Melo, N.B., Dorea, C.E., Alsina, P.J., Araujo, M.V., and Gloger, M. (2017). Ortholeg 2.0-design of a transparent active orthosis. *IEEE Latin America Transactions*, 15(10), 1869–1874.
- Rohmer, E., Singh, S.P., and Freese, M. (2013). V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1321–1326. IEEE.