

MEDIÇÃO AUTOMÁTICA DE ALTURA DE PASTAGEM USANDO VISÃO COMPUTACIONAL

ASSUNÇÃO^{1*}, Talles Bezerra de, BALICO², Leandro Nelinho, BENDAHAN³, Amaury Burlamaqui.

¹ Universidade Federal de Roraima. (tallesbezerra25@gmail.com)

² Professor Ciência da Computação Universidade Federal de Roraima. (leandro.balico@ufr.br)

³ Pesquisador Embrapa Roraima. (amaury.bendahan@embrapa.br)

Palavras Chave: Amazônia, manejo de pastagem, segmentação de imagens, visão computacional.

INTRODUÇÃO

A degradação das pastagens tem sido apontada como um dos principais problemas das atividades pecuárias na Amazônia, trazendo consequência prejudiciais ao meio ambiente e a economicidade das atividades (DIAS-FILHO, 2011).

Um dos principais motivos dessa degradação é o manejo inadequado das pastagens (BENDAHAN, 2015; DIAS_FILHO, 2017). Dentro do manejo, a altura do pasto, para entrada e saída dos animais, é o ponto chave ao sucesso de um manejo adequado (COSTA; QUEIROZ, 2013).

Se de um lado, sabe-se que as pesquisas já estabeleceram essas alturas para as mais diversas forrageiras utilizadas na região amazônica, de outro lado, observa-se a pouca adesão dessa tecnologia pelos produtores.

Nas atividades de seu dia-a-dia, os produtores, usam réguas marcadas (COSTA; QUEIROZ, 2013) com as alturas de entrada e saída dos animais e percorrem, periodicamente, os pastos para essa verificação, o que pode ser um dos muitos motivos para a pouca adesão dessa ferramenta.

Com o avanço das tecnologias informáticas e o aumento do uso de drones no setor agropecuário, uma nova perspectiva se abre para que a altura dos pastos seja mais utilizado por pecuaristas como indicador de manejo.

Nesse contexto, esse trabalho teve o objetivo de testar um algoritmo que por meio de imagens captadas por sensor portado por um drone consiga medir a altura do pasto de maneira automática.

MATERIAL E MÉTODOS

A metodologia foi dividida em 6 etapas. A primeira foi a coleta das imagens, a segunda foi o tratamento da imagem para a identificação da barra, a terceira foi a de identificação das bordas da barra, a quarta a identificação dos pontos extremos das retas de largura e altura, na quinta etapa foi do cálculo da altura e largura da barra e na etapa final foi a de testes.

Na primeira etapa, de coleta das imagens, utilizou-se o drone multirrotor da marca DJI modelo Phantom 4 Advanced, equipado com sensor CMOS RGB FC6310, 8.8, 5472x3648 de 20 megapixel, acoplado a um gimbal de três eixos para estabilização (DJI, 2020). Nessa etapa utilizou-se uma barra de ferro, pintada de vermelho, fincada ao chão com altura de 1,5 m e largura de 6,2 cm. A imagem foi tirada a uma distância cerca de 5 m e altura entre 1,5 e 2 m. Na imagem utilizada neste trabalho, colocou-se uma caixa de cor cinza de 30 cm de altura, para se obter a medida exata da parte da barra que a imagem captou (Figura 1).



Figura 1: Exemplo de imagem para teste.

Nas etapas 2 a 6 todos os procedimentos foram realizados utilizando as funções do OpenCV descritas em cada etapa abaixo (INTEL, 2020).

Na segunda etapa, de tratamento da imagem, que teve o objetivo de facilitar a identificação da barra, converteu-se a imagem original para o sistema de cores HSV (Hue, Saturation e Value) (SURTI, 2020) e utilizando o intervalo de tons vermelhos em HSV, criou-se um filtro mostrando apenas os pixels que possuem a cor dentro deste intervalo, sendo aplicado na imagem original e destacando a barra vermelha.

Na etapa três, de identificação das bordas da barra, a imagem foi convertida para tons de cinza (Grayscale) e levemente borrada (BLUR) dilatando os pixels com o objetivo remover ruídos na imagem. Em seguida foi aplicado o detector de bordas Canny Edge Detector, criando uma imagem apenas das bordas. Após, foi aplicado o detector de contornos findContours, que encontra formas na imagem e guarda em uma lista os pixels que formam o contorno dessa forma. Os pixels do contorno foram novamente marcados na imagem drawContours (ROSEBROCK, 2016). Esses dois processos foram executados duas vezes seguidas.

Na etapa quatro, primeiramente definiu-se os limites das barras marcando os quatros vértices (Pontos amarelos da Figura 2). Passou-se, em seguida, ao cálculo dos quatro pontos médios entre os pontos localizados nos vértices (Pontos cinza da Figura 2). Após a marcação dos pontos médios acima descrito, calculou-se os pontos médios entre os pontos limites e o pontos médios calculados anteriormente (Pontos azuis da Figura 2). Finalizando essa etapa, construiu-se o algoritmo de rasterização de linha de Bresenham, na linguagem Python (FOUNDATION, 2020), para traçar linhas ligando os pontos médios, obtendo-se três linhas horizontais e três verticais ligando as bordas, direita e esquerda e superior e inferior, respectivamente. (Figura 2). No momento em que o algoritmo traça as retas, ele verifica se o novo ponto da reta faz parte da borda, dessa maneira os pontos da borda (Pontos pretos da Figura 2) ficam mais justos, melhorando a precisão das medidas.

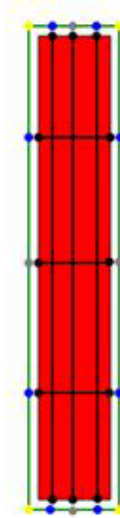


Figura 2: Ilustração com os pontos marcando os vértices (amarelos), os pontos médios (cinza), Processo de escolha do pontos

Na etapa seguinte, calculando a distância euclidiana entre os novos pontos da borda, obteve-se as medidas das três larguras e das três altura, em seguida foi calculada as médias das larguras e das alturas, chamadas largura em pixels (L_{px}) e altura em pixels (A_{px}), respectivamente.

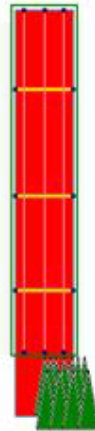


Figura 3: Ilustração da medição da largura e altura em pixels

Utilizando regra de três simples, calculou-se a proporção entre pixels e centímetros das medidas utilizando a fórmula da figura abaixo.

$$\frac{L_{px}}{A_{px}} = \frac{L_{cm}}{A_{cm}}$$

$$A_{cm} = \frac{L_{cm} * A_{px}}{L_{px}}$$

Figura 4: Equação utilizada para estimar as medidas.

Onde:

L_{px} : Largura em pixels;

L_{cm} : Largura em centímetros;

A_{px} : Altura em pixels;

Acm: Altura em centímetros.

A etapa final foi de teste, utilizou-se uma barra vermelha de 6,2 cm de largura e o seu topo está a 150 cm de altura em relação ao solo, na frente da barra, há uma caixa com 30 cm de altura.

RESULTADOS E DISCUSSÃO

Na Figura 5, observa-se o resultado do tratamento da imagem quando foi aplicado o filtro de tons vermelhos na imagem, resultando em uma nova apenas com os pixels vermelhos destacando a barra.

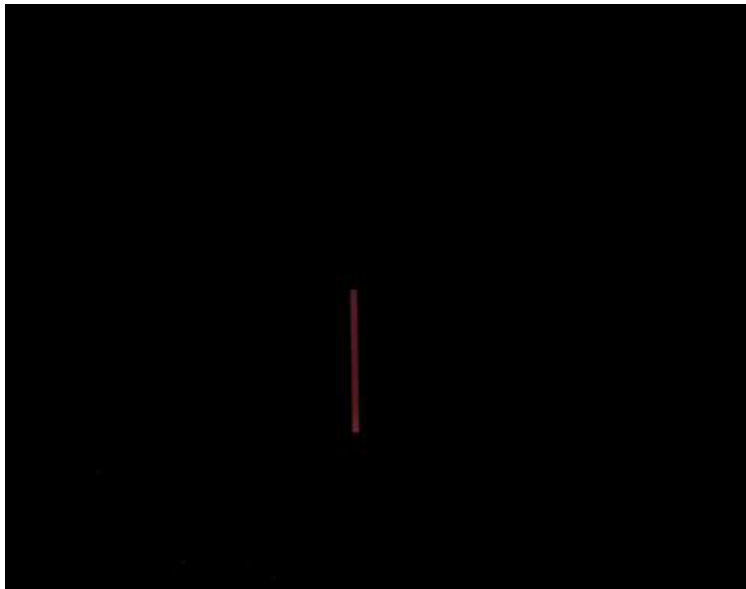


Figura 5: Imagem apenas com os pixels vermelhos.

Resultado com a definição das bordas após a conversão para tons cinza e utilização do detector de bordas Canny Edge Detector e marcando-as com drawContours (Figura 6).

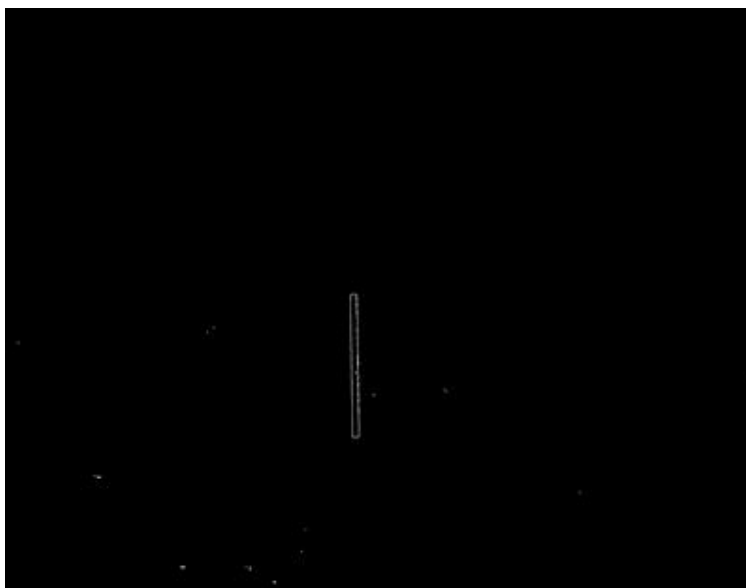


Figura 6: Contornos da barra marcados

Os resultados dos testes, após a aplicação da metodologia, podem ser observados na Figura 7 a e b, respectivamente, para o teste 1 e Teste 2.

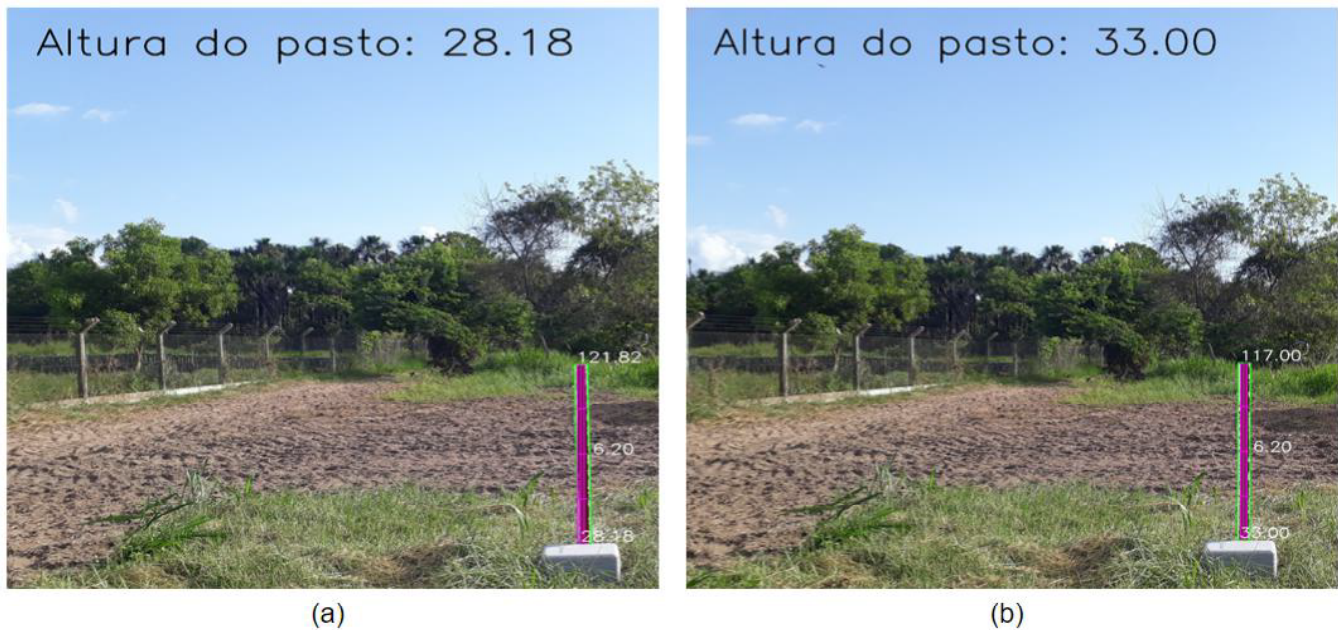


Figura 7: Resultados do teste 1 (a) e teste 2 (b).

Os pontos brancos mostram o resultado de marcação do contorno da barra. As linhas verdes mostram o resultado dos processos de identificação da barra. As linhas rosas mostram a largura e altura calculada da barra.

No primeiro teste (Figura 7a) obteve-se a altura de 121,82 cm da altura correspondente a distância entre o topo de barra e o topo da caixa e no teste 2 (Figura 7b), essa medida foi de 117 cm. Em relação à altura da caixa, no teste 1 a altura foi de 28,18 (Figura 7a) e no teste 2 foi de 33 cm (Figura 7b). As medidas reais da altura da barra foram de 120 cm e da altura da caixa foi 30 cm, medidas intermediárias às calculadas nos dois testes.

CONCLUSÕES

Os testes do algoritmo obtiveram resultados próximos das medidas reais dos objetos, com as medidas da caixa variando entre 2 a 4 cm para mais ou para menos. Foi observado que as variáveis iluminação, distância e ângulo de captura da imagem podem estar influenciando na precisão das estimativas das medidas de altura. Há necessidade de aprimoramento do algoritmo e aprofundar os testes com o objetivo de estabelecer um protocolo de captura de imagens a fim de minimizar o efeito das variáveis iluminação, distância e ângulo de captura, na precisão das estimativas das medidas de altura.

Contudo, a metodologia de medição apresentada demonstra ter potencial para que uma aplicação seja disponibilizada para auxiliar produtores rurais no processo de manejo de pastagens.

AGRADECIMENTOS

Agradecimento a Universidade Federal de Roraima e ao departamento de Ciência da computação, A Embrapa Roraima, e ao Fundo Amazônia, por financiar parte dessa pesquisa, por meio do Projeto Estratégias para o desenvolvimento sustentável da agricultura familiar com enfoque em sistemas agroflorestais e recuperação de pastagens degradadas na região amazônica (AMAPEC).

COSTA, J. A. A. DA; QUEIROZ, H. P. DE. **Régua de Manejo de Pastagens**, 2013. Disponível em: <<https://www.infoteca.cnptia.embrapa.br/bitstream/doc/970497/1/COT125.pdf>>. Acesso em: 24 set. 2020.

DIAS-FILHO, M. **Degradação de pastagens: processos, causas e estratégias de recuperação**. 4. ed. Belém, PA: Do Autor, 2011.

DIAS_FILHO, M. **Degradação de pastagens: o que é e como evitar**. Brasília,DF: Embrapa, 2017.

DJI. **DJI Phantom 4 Advanced**. Disponível em: <<https://www.dji.com/br/phantom-4-adv>>. Acesso em: 24 set. 2020.

FOUNDATION. **Python.org**. Disponível em: <<https://www.python.org/>>. Acesso em: 28 set. 2020.

INTEL. **OpenCV**. Disponível em: <<https://opencv.org/>>. Acesso em: 28 set. 2020.

ROSEBROCK, A. **Measuring size of objects in an image with OpenCV**, 2016. Disponível em: <<https://www.pyimagesearch.com/2016/03/28/measuring-size-of-objects-in-an-image-with-opencv/>>. Acesso em: 24 set. 2020.

SURTI, R. **Thresholding Operations using inRange**. Disponível em: <https://docs.opencv.org/master/da/d97/tutorial_threshold_inRange.html>. Acesso em: 24 set. 2020.