

Detecção Automática de Linhas de Plantio em Culturas Agrícolas

Matheus Freitas Martins[†], Domingos Sárvio M. Valente[‡], José Augusto M. Nacif[†],

[†]Instituto de Ciências Exatas e Tecnológicas – Universidade Federal de Viçosa, *Campus* Florestal, Minas Gerais, Brasil

[‡]Depto. de Engenharia Agrícola, Universidade Federal de Viçosa, *Campus* Viçosa, Minas Gerais, Brasil

{matheus.f.martins,valente, jnacif}@ufv.br

Resumo—A detecção automática de linhas de plantio em diferentes culturas não é um trabalho simples e exige um esforço para ser realizada manualmente pelo agricultor. A automação deste processo é um bom exemplo de aplicação que tem se tornado promissora com a inserção da tecnologia digital na agricultura para planejar rotas de colheita. Este artigo compara o desempenho de três algoritmos de identificação do trajeto de culturas agrícolas a partir de imagens simuladas por software e de fotografias capturadas por um Veículo Aéreo Não Tripulado (VANT). Os resultados das soluções propostas foram validados pela métrica *Dice Similarity Coefficient* (DSC) por meio da comparação dos resultados com as máscaras dos caminhos idealizados. Os resultados obtiveram um desempenho de até 99,5%.

Abstract—The automatic detection of planting rows in different crops is not a simple job and requires an effort to be performed manually by the farmer. The automation of this process is a good example of an application that has become promising with the insertion of digital technology in agriculture to plan crop routes. This paper compares the performance of three agricultural crop row identification algorithms based on software simulated images and photographs captured by an Unmanned Aerial Vehicle (UAV). The results of the proposed solutions were validated by the *Dice Similarity Coefficient* (DSC) metric by comparing the results with the idealized path masks. The results obtained a performance of up to 99.5%.

I. INTRODUÇÃO

Desde o começo da década de 80, a automação da produção agrícola se tornou um campo amplamente pesquisado no que refere-se à agricultura de precisão [1]. Em destaque, o Brasil é um dos maiores produtores de açúcar e etanol [2]. Assim, a produção agrícola pode ser otimizada por meio da automatização dos processos de colheita, pois essa tarefa requer um alto nível de precisão e demanda a realização de tarefas manuais e repetitivas pelos humanos. Para tal, uma orientação precisa das linhas de cultivos são necessárias para uma máquina agrícola autônoma percorrer eficientemente os caminhos para realizar a colheita. Ademais, as informações da geolocalização das linhas permitem aprimorar a aplicação de insumos e trabalhar com trajetos com falhas.

As coordenadas de uma linha de cultura semeada não necessariamente representam as coordenadas que devem ser consideradas para a extração. O problema justifica-se devido a fatores ambientais que tornam o terreno e/ou a plantação irregulares [2]. Além disso, o sistema deve ser capaz de lidar com inúmeras situações de iluminação, pois algumas condições

afetam diretamente na identificação e processamento [3]. Deste modo, o problema consiste em lidar com cenários nos quais o trajeto da plantação seja linear ou não linear, através de imagens capturadas em baixa ou média altitude por um Veículo Aéreo Não Tripulado (VANT). Destaca-se que este veículo permite um acompanhamento recorrente de baixo custo do trajeto planejado do plantio, pois os caminhos podem sofrer variações e falhas ao longo do tempo.

Este trabalho leva em consideração diversos tipos de situações que são apresentadas ao longo do ciclo de vida de uma cultura, independente do seu tipo de trajeto, seja ele linear ou não linear. Serão apresentadas três diferentes abordagens: a primeira determina um ângulo em comum em relação às linhas de cultivo, a segunda traça retas com a utilização de clusterização de pontos com a técnica *RANdom SAmple Consensus* (RANSAC), e a terceira extrai o esqueleto da segmentação. Assim, foi utilizado neste artigo dois conjuntos de imagens aéreas: (I) fotografias obtidas por um VANT; (II) imagens de culturas genéricas com suas respectivas máscaras desenhadas digitalmente por um especialista. O *dataset* contendo as imagens simuladas foi disponibilizado em um repositório público.¹ Ressalta-se que tal conjunto de culturas aéreas simuladas foi desenvolvido devido à dificuldade de encontrar *datasets* que possuam imagens aéreas de culturas reais.

A literatura apresenta diversas soluções para o problema de identificação de linhas de plantio. Em [2] os autores utilizam uma rede de segmentação semântica para segmentar e classificar as imagens obtidas por um VANT, aperfeiçoando os resultados através da transformação de Radon. Guerrero et al. [3] utilizam técnicas de segmentação de imagens para trabalhar em um sistema de visão a bordo da máquina agrícola que estabelece a orientação, caminho e densidades de ervas daninhas. Em [4] Basso e Freitas utilizam uma abordagem de orientação composta de hardware e software para um VANT com base em técnicas de processamento de imagem. Este trabalho assemelha-se no aspecto de manipular técnicas de processamento digital por meio de imagens aéreas de plantios. Deste modo, o objetivo desta pesquisa é desenvolver um método que:

- i) Lide com falhas e caminhos irregulares;

¹<http://github.com/lesc-ufv/crop-row-detection>

- ii) Convenha para diferentes tipos de culturas;
- iii) Identifique linhas de cultura que possuam características lineares e não lineares.

O restante deste trabalho está organizado da seguinte forma: a Seção II introduz os conceitos mais básicos utilizados neste trabalho; a Seção III apresenta os trabalhos relacionados; a Seção IV apresenta a metodologia com as principais decisões do projeto; a Seção V mostra os principais resultados e comparações; a Seção VI encerra com a conclusão e trabalhos futuros, e por fim apresenta as referências bibliográficas.

II. CONCEITOS INTRODUTÓRIOS

Tendo em vista uma imagem obtida a partir de um VANT ou uma máquina agrícola com uma câmera frontal, é necessário fazer com que o computador a interprete de modo a traçar as rotas adequadas para a realização da colheita de forma autônoma. No entanto, tais imagens não são livres de imperfeições, isso faz com que certos detalhes sejam difíceis de serem extraídos. Dessa maneira, diversas técnicas de processamento digital de imagens podem ser utilizadas para facilitar a extração destas informações. É possível aplicar técnicas de segmentação, tais como remoção de ruídos, aplicação de filtros morfológicos e estabelecimento de regiões de interesse. Assim, uma vez finalizado o pré-processamento, é possível extrair as informações das rotas com base em técnicas de regressão linear e clusterização, por exemplo.

Para realizar a segmentação de uma imagem, algumas técnicas como segmentação Otsu, transformada de Hough, aplicação de filtros morfológicos e de suavização podem ser utilizadas.

A **Segmentação Otsu** é utilizada para realizar o *thresholding* automático das imagens [5]. Dada uma imagem em tons de cinza, o processo de *thresholding* consiste em atribuir coloração branca para os pixels cujo valor é superior ou igual ao valor do limiar (*threshold*), e coloração preta caso contrário. Isso permite destacar regiões de interesse para análises futuras.

Ruídos podem ser definidos como uma variação dos valores nos níveis de cinza dos pixels de uma imagem, causados por erros na transmissão de dados, ou eventuais distorções introduzidas na fase de aquisição de uma imagem em geral [6]. Uma forma de retirar esses ruídos é através do uso de **de filtros de suavização**. Alguns exemplos de filtros de suavização são os filtros Bilateral, Mediana, Gaussiano e Prewitt.

A **Aplicação de filtros morfológicos** utiliza uma máscara ou elemento estruturante para determinar o processo de filtragem geométrica, sendo eficazes para extrair informações das imagens. A morfologia, por sua vez, descreve um conjunto de técnicas de processamento não linear de imagens que trata da forma (estrutura) ou morfologia de objetos. Ela tem como operações básicas os filtros morfológicos Erosão e Dilatação, e seus derivados Abertura e Fechamento [6].

A **Transformada de Hough** é uma técnica utilizada para a extração de características das imagens [7]. Ela se utiliza de um sistema de votação para identificação de objetos, sendo útil para identificação de linhas e estruturas de interesse.

Para que funcione de acordo com os objetivos esperados, é recomendado que a imagem possua boa qualidade e a segmentação seja satisfatória para obter de forma adequada o que se pretende extrair.

Após realizada a segmentação, pode-se aplicar estratégias de regressão linear. No entanto, como em uma mesma imagem podem haver diversos caminhos a serem traçados, isso significa que várias regressões lineares são necessárias considerando-se janelas de avaliação na imagem. Tais janelas podem ser pensadas como *clusters* de pixels em que as regressões lineares podem ser realizadas. A cada *cluster* pode ser aplicado a técnica *RANdom SAMple Consensus* (RANSAC):

De acordo com [8], **RANSAC** é uma técnica para estimar os parâmetros de um modelo matemático a partir de um conjunto de pontos. Em outras palavras, este método é útil para estabelecer a reta que melhor se adapta a um determinado conjunto de pontos, sendo possível estabelecer a rota que uma máquina autônoma deverá tomar. No entanto, este método pode consumir muito tempo de processamento e memória, sendo também dependente das regiões analisadas na imagem, isto é, o subconjunto de pixels da imagem para se analisar a direção do caminho em cada parte da imagem. Este último ponto é uma tarefa relativamente complicada ao tentar pensar em uma solução mais genérica, visto que depende das características da imagem, da segmentação e dos tratamentos realizados. Uma alternativa é considerar o número de casamentos de pixels que a reta gerada consegue realizar em relação à imagem segmentada.

III. TRABALHOS RELACIONADOS

Foi proposto pelo trabalho [2] uma maneira de utilizar imagens obtidas por VANTs para efetuar a segmentação e detecção de linhas da cultura de cana-de-açúcar. Foi estabelecido duas técnicas para tratar do problema, (I) baseado no Algoritmo Genético (AG) e Otsu para produzir imagens binarizadas; (II) Semantic Segmentation Network (SSN) para segmentar e classificar as linhas de plantio. Posteriormente, as linhas segmentadas são transformadas pela transformadora de Radon para reconstruir e melhorar as linhas que se encontram soltas.

Por meio do trabalho desenvolvido por [3], um sistema de visão mecânica foi construído na perspectiva frontal da máquina para a detecção de linhas de cultivo, consegue lidar com terrenos regulares e não regulares. O seu trabalho tem como consideração três objetivos principais: (I) obter uma orientação precisa do ponto de partida para o destino; (II) obter uma orientação precisa do trajeto a ser percorrido; (III) determinar a densidade e camadas de ervas daninhas que há na plantação. Sua metodologia aborda as estratégias de processamento de imagem para detectar as linhas de cultura com base em um ajuste de precisão e extração de dados para determinar densidades de ervas daninhas na plantação de milho.

Um método eficiente de detecção de linhas de cultura que consegue detectar com precisão retas e curvas foi produzido por [1], sua técnica utiliza a programação dinâmica para

combinar evidência de imagem e conhecimento prévio de 281 imagens de diferentes culturas. A abordagem proposta consiste em três etapas: (I) detecção de vegetação; (II) detecção de padrões regulares; (III) determinação de um modelo ótimo de linha de cultura. Sua metodologia consiste na comparação de dois métodos: (I) baseado na transformadora de Hough; (II) regressão linear. Além disso, estabelece e demonstra uma métrica própria que considerou ser superior para avaliação dos métodos de detecção de linhas de cultura.

A concepção de um sistema com orientação embarcada em VANTs foi desenvolvida por [4]. A partir da utilização de técnicas de processamento de imagens, produziu-se dois algoritmos, o primeiro foi denominado de *Crop Row Detection* que tem como objetivo identificar corretamente as linhas de cultivo e o segundo foi denominado de *Line Filter* que é responsável por gerar corretamente os parâmetros de condução.

O trabalho [9] estabelece um método para detectar linhas lineares e não lineares durante o diversificado estágio de crescimento da vegetação da cultura que eventualmente surgem ervas daninhas, considerando espaçamentos regulares e irregulares entre o plantio. Por meio da perspectiva de uma câmera frontal instalada em um trator, identifica-se as linhas de cultivo com base em duas finalidades: (I) orientação precisa e autônoma; (II) tratamentos específicos do local, incluindo remoção de ervas daninhas. Assim sua técnica consiste nas fases de (I) segmentação da imagem; (II) identificação dos pontos de partida para determinar o início das linhas de cultivo; (III) detecção de linhas de cultivo.

Em [10] os autores propuseram o conceito de desenvolver um algoritmo que trabalha com a ideia de ser uma proposta genérica para identificar linhas de plantio. Sua proposta consiste em duas etapas: (I) agrupar hierarquicamente os pontos para subdividi-los em linhas; (II) correção de imprecisões das linhas.

Este trabalho apresenta comparações entre três diferentes abordagens, sendo elas: (I) Baseado na hipótese de que as linhas de cultivo são paralelas, calcula-se o ângulo médio para os traçados; (II) Técnica RANSAC aplicada à clusterização de pontos; (III) Esqueletização das segmentações.

IV. METODOLOGIA

Para a elaboração deste trabalho, as tecnologias adotadas foram *Python 3.8*, *OpenCV*, *Scikit-image* e *Scikit-learn*. As três diferentes abordagens desenvolvidas consistem na aplicação de filtros e manipulações de cores para realizar a segmentação da imagem. As avaliações das segmentações para cada uma das diferentes técnicas são descritas a seguir:

1) Se vale da análise dos tons de cinza para identificar os possíveis pontos que fazem parte de cada rota na cultura. A Figura 2(a) mostra a análise dos tons de cinza da imagem segmentada considerando-se o centro da imagem da esquerda para a direita. A cada pico representado no gráfico existe a possibilidade do pixel em questão fazer parte da rota a ser traçada. A Figura 2(b) mostra os pontos marcados pelo algoritmo, por meio das análises dos tons de cinza para algumas alturas da imagem, que foram utilizados para o

cálculo do ângulo médio, enquanto a Figura 3(a) mostra o traçado das retas considerando-se tais pontos. A estratégia para traçar tais retas é apresentada no Algoritmo 1 e consiste em:

- i) A partir da imagem de interesse (*img*) e de um parâmetro de distância de salto (d), encontra os pontos analisando-se os tons de cinza da imagem. O valor para o parâmetro d obtido empiricamente foi de 37 durante os testes (linhas: 2-15);
- ii) Para cada par de pontos em alturas distintas, calcula-se o ângulo formado pela reta gerada ao ligar ambos os pontos. Para cada ângulo obtido, armazena-os em um vetor e remove os valores outliers (linhas: 16-24);
- iii) Por fim, calcula-se a média geral dos pontos e o desvio padrão (linhas: 25-36). Os pontos ligados em cada rota para o traçado das retas definitivas são aqueles mais próximos da média geral.

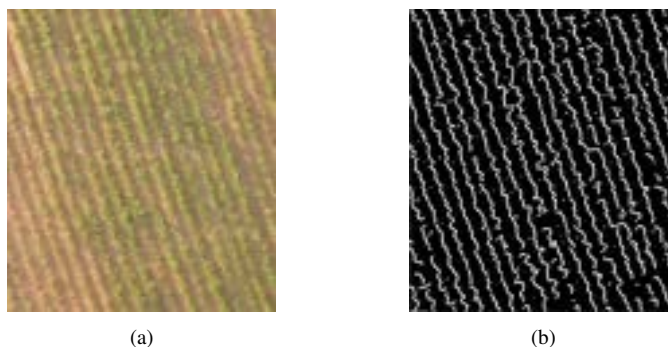


Fig. 1: Realizando a segmentação. (a) Recorte de uma fotografia obtida por um VANT de uma plantação de uma cultura de cana-de-açúcar. Resolução utilizada: 603 x 510 pixels. (b) Segmentação da Figura (a).

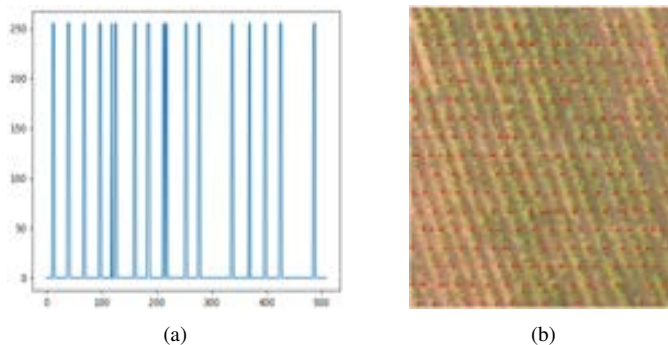


Fig. 2: Identificação de pontos. (a) Gráfico das variações de tons de cinza em relação a linha central horizontal. (b) Pontos marcados de vermelho por meio das análises dos tons de cinza da Figura (a).

Algorithm 1: Análise de Ângulos

Input: img, d **Output:** None**Start**

```
1  gray_img = Aplicação das seguintes operações sobre
   img: [Filtro Bilateral (Kernel 7x7, Filtro Mediana
   (Kernel 5x5), Conversão de imagem BGR para Gray,
   Filtro Gaussiano (Kernel 3x3), Filtro adaptativo
   gaussiano para thresholding (Limiar igual à média
   dos pixels da imagem), Filtro Prewitt (Kernel 3x3),
   Filtro Mediana (Kernel 3x3), Filtro binário para
   thresholding (Limiar inferior 100, Limiar superior
   255)];
2  mp ← Matriz vazia de pontos de cada linha
   separados por  $d$  pixels de distância;
3  h ← altura de gray_img;
4  w ← largura de gray_img;
5  index_array ← -1;
6  For  $y \leftarrow 0; y < h; y \leftarrow y + d$  do
7      Adicione um vetor vazio a mp;
8      index_array ← index_array + 1;
9      For  $x \leftarrow 0; x < w; x \leftarrow x + 1$  do
10          $x0 \leftarrow x$ ;
11         While  $x < w$  && gray_img[y][x] == 255
12             do
13                  $x \leftarrow x + 1$ ;
14             End
15             If  $x \neq x0$  then
16                  $xm \leftarrow (x + x0) // 2$ ;
17                 Adicione  $xm$  a mp[index_array];
18             End
19         End
20     End
21     al ← Lista vazia de ângulos entre pontos em alturas
        consecutivas;
22     For  $i \leftarrow 0; i < index\_array + 1; i \leftarrow i + 1$  do
23          $j\_max \leftarrow$  Tamanho do vetor mp[i];
24         For  $j \leftarrow 0; j < j\_max; j \leftarrow j + 1$  do
25              $p0 \leftarrow (mp[i][j][0], mp[i][j][1])$ ;
26              $p1 \leftarrow$  Ponto mais próximo a  $p0$  à distância
                vertical  $d$ ;
27              $\theta \leftarrow$  Ângulo da reta que liga  $p0$  a  $p1$ ;
28             Adicione  $\theta$  a al;
29         End
30     End
31     al ← Remova outliers de al;
32      $m \leftarrow$  ângulo médio de al;
33      $std \leftarrow$  desvio padrão de al;
34     For  $i \leftarrow 0; i < index\_array + 1; i \leftarrow i + 1$  do
35          $j\_max \leftarrow$  Tamanho do vetor mp[i];
36         For  $j \leftarrow 0; j < j\_max; j \leftarrow j + 1$  do
37              $p0 \leftarrow (mp[i][j][0], mp[i][j][1])$ ;
38              $x1 \leftarrow h - 1$ ;
39              $y1 \leftarrow p0[0] + (p0[1] - x1)/m$ ;
40              $p1 \leftarrow (y1, x1)$ ;
41             If Reta não foi traçada para rota then
42                  $p0 \leftarrow (p1[0] + p1[1]/m, 0)$ ;
43                 Trace a reta de  $p0$  a  $p1$  se o ângulo  $\alpha$ 
                    entre esses pontos respeitar
                     $m - std \leq \alpha \leq m + std$ ;
44             End
45         End
46     End
47 End
```

2) Se vale da construção de *clusters* de pontos e da aplicação

do algoritmo RANSAC sobre estes *clusters*. A construção de *clusters* é mostrada no Algoritmo 2 e consiste em definir janelas de avaliação de pontos da imagem segmentada, no qual:

- i) A construção foi fundamentada em um conjunto de parâmetros específicos que devem ser definidos para cada imagem a ser analisada. Os parâmetros são: imagem de interesse (*img*), a largura e altura do *kernel* de busca (*hks* e *vks*) respectivamente, o comprimento de um *kernel* quadrado auxiliar para tratar os limites das imagens (*sks*), o número mínimo de pontos que devem se encontrar dentro do *kernel* para este ser válido como um *cluster* (*ms*), cor em tons de cinza mínimo para ser considerado ponto de um *cluster* (*t*) e quantidade de pixels que a janela deve saltar para a avaliação de novos *kernels* (*jz*). Os valores para os parâmetros obtidos empiricamente são mostrados na Tabela I.

TABELA I: Parâmetros utilizados no Algoritmo 2

hks	vks	sks	jz	t	ms
35	100	8	35	110	1300

- ii) Com os *clusters* montados (linhas: 2-17), o algoritmo RANSAC pode ser aplicado para determinar as equações das retas de cada *cluster* (linha: 18).

Algorithm 2: Clusterização de Pontos + RANSAC

Input: $img, hks, vks, sks, jz, t, ms$ **Output:** None**Start**

```
1 gray_img = Aplicação das seguintes operações sobre
img: [Filtro Bilateral (Kernel 7x7, Cor Sigma 7, Cor
Espacial 7, 5 repetições), Conversão de imagem
BGR para Gray, Filtro Gaussiano (Kernel 3x3),
Filtro adaptativo gaussiano para thresholding (Limiar
igual à média dos pixels da imagem), Filtro Prewitt
(Kernel 3x3), Aplicar Fechamento (3 iterações de
Erosão e 5 iterações de Dilatação)];
2 cluster_id ← 0;
3 res ← Dicionário de clusters vazio;
4 h ← altura de gray_img;
5 w ← largura de gray_img;
6 hs ← [h//8, 4 * h//8, 7 * h//8];
7 For h in hs do
8   y ← 0;
9   While y < w do
10    For k in [-sks, sks] do
11     If gray_img[h2 + k][y] > t then
12      cluster ← Vetor vazio;
13      rh ← Vetor de valores inteiros entre
max(0, y - hks) e
min(w, y + hks + 1);
14      rv ← Vetor de valores inteiros entre
(h2 - vts + k) e
(h2 + vks + 1 + k);
15      Adicione todo pixel de gray_img
que seja maior que o valor de t
percorrendo o retângulo de largura
rh e altura rv;
16      Adicione cluster em res se seu
tamanho for superior ou igual a ms;
17      Se o vetor foi adicionado a res,
incremente cluster_id em uma
unidade y ← y + jz;
18      End
19    End
20  End
21 End
22 Aplique o algoritmo RANSAC para todos os clusters
de pontos em res e trace as retas encontradas pelo
algoritmo;
23 End
```

3) Realiza a esqueletização da segmentação. Esse processo está descrito no Algoritmo 3. Seu mecanismo se dá pela captura dos pontos médios das rotas presentes na segmentação (linhas: 10-12) em uma avaliação de cada linha da imagem. Por depender única e exclusivamente da segmentação, é necessário que a segmentação seja excelente para um bom desempenho na aplicação desta técnica.

Algorithm 3: Esqueletização

Input: img **Output:** None**Start**

```
1 gray_img = Aplicação das seguintes operações sobre
img: [Conversão de imagem BGR para Gray,
Inversão de cores (bitwise_not), Median Blur (Kernel
39x39), Filtro binário e OTSU para thresholding
(Limiar inferior 100, Limiar superior 255)];
2 h ← altura de gray_img;
3 w ← largura de gray_img;
4 For y ← 0; y < h; y ← y + 1 do
5   gray_img[y][w - 1] ← 255;
6   For y ← 1; y < h; y ← y + 1 do
7     p ← 'x': 0, 'y': y;
8     cor ← (0, 0, 0);
9     If gray_img[p['y']][p['x']] == 255! =
gray_img[y][x] then
10      cor ← (255, 255, 255);
11      End
12      xk ← x - (x - p['x'])//2;
x1 ← (xk, y);
Marque o ponto x1 na figura como parte da
rota;
13   End
14 End
```

Em relação à métrica de desempenho, os resultados das imagens processadas foram comparados com máscaras que podem ser entendidas como gabaritos que possuem o traçado idealizado das linhas. Estas máscaras foram geradas digitalmente por um especialista abrangendo todos os trajetos dos traçados dos plantios. Para esta comparação usamos o *Dice Similarity Coefficient* (DSC) [11], um coeficiente de similaridade de segmentação que mede a semelhança entre duas imagens sobrepostas que é definido por:

$$DSC = \frac{2TP}{2TP + FP + FN} \quad (1)$$

em que:

- TP (True Positive) - Corresponde ao número de pontos corretamente identificados que fazem parte da rota da máscara;
- FP (False Positive) - Corresponde ao número de pontos identificados a fazerem parte da rota mas que não o fazem na máscara de avaliação;
- FN (False Negative) - Corresponde ao número de pontos marcados como não pertencentes à rota mas que deveriam ser marcados de acordo com a máscara de avaliação.

V. RESULTADOS

Nesta seção são apresentados os resultados das execuções e as respectivas observações entre elas, que são mostradas nas Tabelas II, III e IV. Para tal, foi utilizado a imagem 1(a) para os algoritmos 1 e 2 e imagem 5(a) para o algoritmo 3. Para ser possível estabelecer uma comparação entre os diferentes

resultados, máscaras foram utilizadas para a validação da métrica DSC.

Ao executar para o Algoritmo 1, obteve-se o resultado apresentado na Figura 3(a). O gabarito está representado pela cor vermelha e o caminho resultante combinado com o gabarito está representado pela cor azul na Figura 3(b). O valor de DSC obtido nesta execução foi de 74,4%.

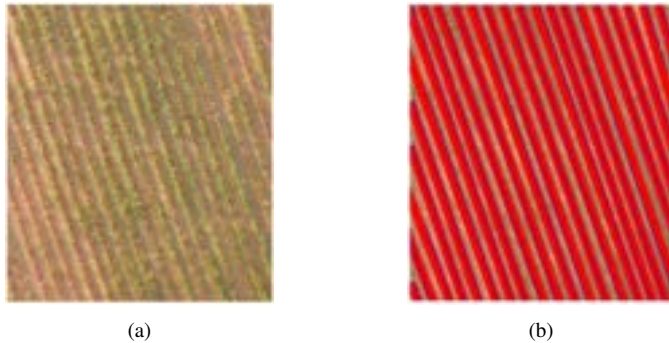


Fig. 3: Resultado da Análise de Ângulos. (a) Linhas traçadas pelo algoritmo. (b) Resultado da figura (a) sobre a máscara.

TABELA II: Resultado do primeiro algoritmo calculado pelo DSC.

Alg.	Fig.	TP (%)	FP (%)	FN (%)	DSC (%)
1	1(a)	59,2	5,37	35,38	74,3

Ao executar para o Algoritmo 2, nota-se que na Figura 4(a) mais de uma regressão linear foi delineada em alguns trajetos das linhas da plantação, porém na Figura 4(b) foi considerado para a validação mais de uma linha caso a linha resultante exceda o limite delimitado pela máscara, o que proporcionou um desempenho de 91,7%.

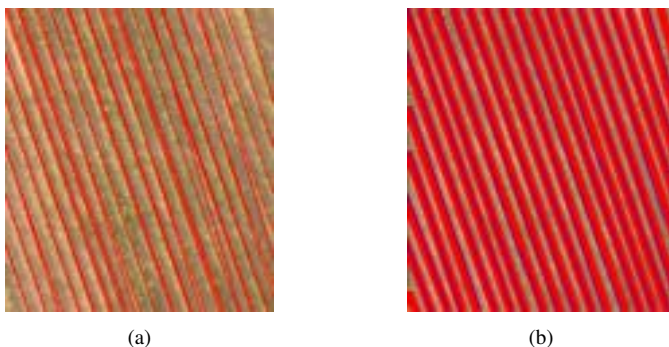


Fig. 4: Resultado da Clusterização de Pontos + RANSAC. (a) Linhas traçadas pelo algoritmo. (b) Resultado da Figura (a) sobre a máscara.

TABELA III: Resultado do segundo algoritmo calculado pelo DSC.

Alg.	Fig.	TP (%)	FP (%)	FN (%)	DSC (%)
2	5(a)	84,7	13,98	1,32	91,7

Ao executar para o Algoritmo 3, identifica-se que o traçado conseguiu identificar a plantação de forma eficiente para praticamente todo o caminho, obteve 99,5% de desempenho. Evidencia-se que tal resultado possui relação com a utilização de uma imagem de boa qualidade, ou seja, não possui interferência dos fatores ambientais. Além disso, uma correção no código pode fazer este algoritmo aumentar seu potencial para alcançar 100% de abrangência na métrica.

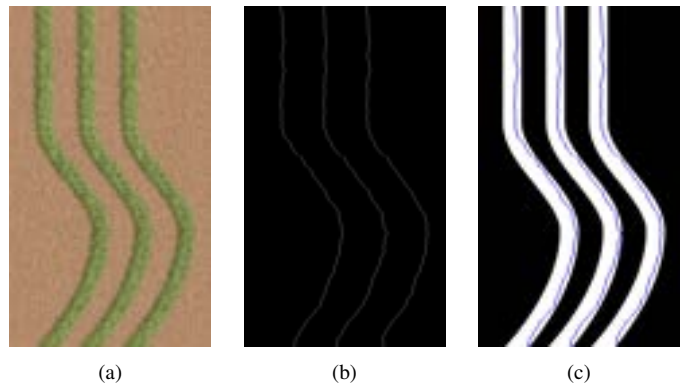


Fig. 5: Resultado da Esqueletização. (a) Imagem gerada por software. Resolução utilizada: 853 x 480 pixels. (b) Imagem resultante da Esqueletização aplicada em (a). (c) Resultado da Figura (b) sobre a máscara.

TABELA IV: Resultado do terceiro algoritmo calculado pelo DSC.

Alg.	Fig.	TP (%)	FP (%)	FN (%)	DSC (%)
3	5(a)	99,18	0,35	0,46	99,5

VI. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou o uso de três diferentes abordagens para a identificação de linhas de culturas: (I) Baseado na hipótese de que as linhas de cultivo são paralelas, calcula-se o ângulo médio para os traçados; (II) Técnica RANSAC aplicada à clusterização de pontos; (III) Esqueletização das segmentações.

O primeiro algoritmo apresentou bom desempenho na identificação de linhas de cultura em imagens cujas segmentações apresentavam falhas. No entanto, por se basear na hipótese que as linhas de cultivo são paralelas, isso a limita para o traçado quando a cultura apresenta curvas.

O segundo algoritmo também apresentou bom desempenho para os mesmos cenários considerados pela abordagem anterior. No entanto, possui uma forte dependência na configuração de parâmetros para efetuar o processo de clusterização de pontos necessários para a aplicação do algoritmo RANSAC. Nota-se que é possível melhorar a técnica realizando a combinação de linhas que se intersectam para formar um caminho contínuo.

Em comparação com os resultados obtidos, o terceiro algoritmo aparentemente demonstrou ser superior ao realizar a esqueletização das segmentações. Ele serve tanto para cenários

não lineares quanto lineares, porém por depender de um bom resultado na segmentação, é necessário que imagens de boa qualidade sejam utilizadas. No entanto, vislumbra-se a possibilidade dessa técnica ser aprimorada ao utilizar uma rede neural convolucional para efetuar a segmentação.

Futuramente, pretende-se estender a funcionalidade do primeiro algoritmo para tratar de cenários curvados, estabelecendo janelas de avaliação nas imagens de modo a traçar pequenas retas dentro de curvas. Além disso, pretende-se aperfeiçoar o terceiro algoritmo ao analisar a realização de segmentações com apoio de uma rede neural convolucional.

REFERÊNCIAS

- [1] I. Vidović, R. Cupec, and Ž. Hocenski, "Crop row detection by global energy minimization," *Pattern Recognition*, vol. 55, pp. 68–86, 2016.
- [2] R. d SILVA, "Detection of sugarcane crop rows from uav images using semantic segmentation and radon transform," 2020.
- [3] J. M. Guerrero, J. J. Ruz, and G. Pajares, "Crop rows and weeds detection in maize fields applying a computer vision system based on geometry," *Computers and electronics in agriculture*, vol. 142, pp. 461–472, 2017.
- [4] M. Basso and E. Pignaton de Freitas, "A uav guidance system using crop row detection and line follower algorithms," *Journal of Intelligent & Robotic Systems*, vol. 97, no. 3, pp. 605–621, 2020.
- [5] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [6] J. I. Miranda, "Processamento de imagens digitais: Prática usando java tm," *Campinas: Embrapa Informática Agropecuária*, 2006.
- [7] P. V. Hough, "Method and means for recognizing complex patterns," *US patent*, vol. 3, no. 6, 1962.
- [8] 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.
- [9] I. D. García-Santillán, M. Montalvo, J. M. Guerrero, and G. Pajares, "Automatic detection of curved and straight crop rows from images in maize fields," *Biosystems Engineering*, vol. 156, pp. 61–79, 2017.
- [10] I. R. Souza, M. C. Escarpinati, and D. D. Abdala, "A curve completion algorithm for agricultural planning," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 284–291.
- [11] T. A. Sorensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons," *Biol. Skar.*, vol. 5, pp. 1–34, 1948.