The International Society of Precision Agriculture presents the
15th International Conference on
Precision Agriculture
26–29 JUNE 2022
Minneapolis Marriott City Center | Minneapolis, Minnesota USA

Scaling up window-based regression for crop-row detection

Guy E. Hokanson[1], Anne M. Denton[1], and J. Paulo Flores[2]

[1] Department of Computer Science, North Dakota State University, Fargo, ND

[2] Department of Agricultural and Biosystems Engineering, North Dakota State Univ., Fargo, ND

A paper from the Proceedings of the
15th International Conference on Precision Agriculture
June 26-29, 2022
Minneapolis, Minnesota, United States

Abstract.

Crop-row detection is a central element of weed detection and agricultural image processing tasks. With the increased availability of high-resolution imagery, a precise locating of crop rows is becoming practical in the sense that the necessary data are commonly available. However, conventional image processing techniques often fail to scale up to the data volumes and processing time expectations. We present an approach that computes regression lines over large windows that surround pixels. Pixels for which the excess green value exceeds a threshold are viewed as being part of plants. Those pixels are then processed as if they were data points in a statistical analysis. Once the regression parameters of slope and offset have been computed, the central positions of the crop rows are inferred. Note that the resolution of the imagery is preserved in the process, because sliding window techniques use every possible sub-window of the image, i.e., there is one window associated with every pixel in the image. Our approach is highly computationally efficient due to an aggregation strategy that scales logarithmically in the window size. Aggregates of spatial coordinates x, x2, y, and xy are initially computed over sliding windows of size 2x2, and then the resulting windows are aggregated iteratively to generate all possible sub-windows of size 4x4, 8x8, 16x16 etc. In our example analysis, 64x64-sized windows capture individual crop rows without overlapping with multiple rows. Computing aggregates for windows of size 64x64 only requires 6 basic passes of the data. As a result, the algorithm is intrinsically designed to allow being deployed in a real-time processing context.

Keywords.

Crop-row detection, weed detection, spatial regression, sliding-window analysis

Introduction

Precision agriculture researchers at NDSU have been working to develop a site-specific weed control approach to control weeds in corn with a post-emergence herbicide application, when the corn vegetative stage is between V4 and V6. The approach consists of using unmanned aerial systems (UAS) imagery to map the weeds in the corn field and use that map as the basis to create an herbicide prescription map. One of the core ideas of that approach is to identify and remove the corn plants from the imagery, and the remaining vegetation fraction is considered weeds. The goal is to identify the corn rows, that optimally would go through the centroid of a corn plant and create a buffer around that row. The next step is then to remove all the vegetation (corn plants) located under the buffered area. Currently the research team uses a Pixel Intensity Projection (Sapkota, 2022) algorithm to identify the rows. Although that algorithm seems to be promising and computationally efficient, it in many cases fails to identify the center of the corn rows, which ultimately results in tips of corn plants to be identified as weeds.

The goal of the presented work is to allow for variations in the crop row location. Fig. 1 shows an exaggerated representation of the problem. The left and the right side both show three crop rows that are intentionally displayed as very uneven. The left panel shows the conventional modeling of crop rows as straight lines for the entire width of the image under consideration. Crop plants are assumed to be within a fixed-width strip around the line that delineates the center of the crop row. When the straight line is a poor approximation of the crop centers, the plants may be closer to the edge of the strip than assumed. Consequently, it may appear as if the space between crop rows has substantial weeds because part of the crop foliage is mistakenly labeled as weeds. The red ellipses highlight problem areas in this analysis. When crop rows are delineated accurately, and strips are created with regard to the actual plant centers, this problem can be mitigated. Auto-steering in tractors, which has become more prevalent in recent years, somewhat reduces the severity of this problem but does fully solve it. Not only are there farmers who do not use autosteering. More importantly, even if the path of the tractor were a perfect straight line, the seeding implement may swing from side to side.

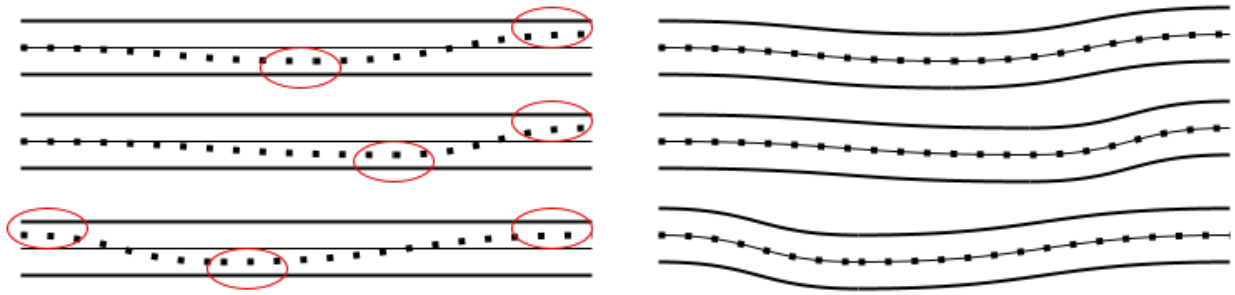

Fig. 1: Schematic representation of the problem of creating strips around imprecisely planted crop rows. When the delineation algorithm is based on an assumption of perfectly straight lines (left panel) crop foliage may be mistaken for weeds. The right panel shows that a more accurate delineation of the crop rows and strips can mitigate this problem.

Limitations to the accuracy of identifying corn rows are not limited to the Pixel Intensity Projection algorithm discussed in the first paragraph. Most other conventional approaches also assume that crop rows are perfect straight lines. When a Hough Line Transform (HLT) is used (Bakker et al. 2008), the predicted crop row locations may be at an angle with regard to the orientation of the image, but they are still typically assumed to be characterized by one straight line. Approaches also exist that consider multiple crop rows together and determine clusters of lines in Hough space to address noise issue (Leemans and Destain 2006).

Researchers have addressed the limitations of straight lines in a variety of ways. Soares et al. (2018) use a tiling approach to allow for locally different row directions. Chen et al. (2021) explore Hough space in a way that allows composing wavy paths with the goal of guiding robots. Bah et al. (2017) developed an approach for crop row prediction in which a skeleton is first extracted to

characterize crop rows, and a Hough transform is later applied. The skeleton construction bears similarities to our regression approach, but our approach is designed to allow scalability at a more fundamental level.

The proposed approach is based on a fast computation of additive measures over sliding windows that has been useful in other contexts, such as regression between different bands (Chavan et al. 2016) and the computation of topographic measures (Gomes et al. 2019). It is possible because there is much overlap between the computations needed for adjacent windows. By aggregating windows successively from 2x2 to 4x4 to 8x8 etc. we can perform the spatial regression task in a time that is comparable to the computation of the Hough transform. Details are discussed in the Experiments Section.

Concepts

Attempting to produce a good fit of crop rows is difficult for many reasons. One problem is that the data are noisy, and weeds may interfere with the fitting process. Another is that crop rows have to be inferred from imagery of the leaves, and the location of plant centers cannot be determined directly through remote sensing. We use images that are collected from unmanned air systems, and which have raster point sizes of approximately 1 cm. In this analysis, Excess Green Indicator (ExG) is used as measure for the presence of a living plant (Woebbecke 1995) and a threshold is then applied that determines the presence of a raster point.

$$ExG = (2 * G) - (R + B) \qquad (1)$$

We use the full statistical power of line-fitting using linear regression and apply it over sliding windows. Each window contributes one data point to the reconstructed image of the plant locations. Because of the nature of linear regression, the approach assumes that rows cannot be vertical. If angles with regard to the horizontal axis are very large, the image has to be rotated by 90 degrees to satisfy the expectations of the regression. Within each window, raster points are fit by an equation of the form:

$$y = a * x + b \qquad (2)$$

The center of the window is assumed to be the origin of the coordinate system. That means that b corresponds to the vertical distance between the regression line and the center of the window at the mid-point in the x dimension. Provided a relatively small shift (no more than 3 pixels in the examples), a point is added to the output image at the shifted location ($x=0$ and $y=b$ in the reference system of the window). Fig. 2 shows the process for an 8x8 window.

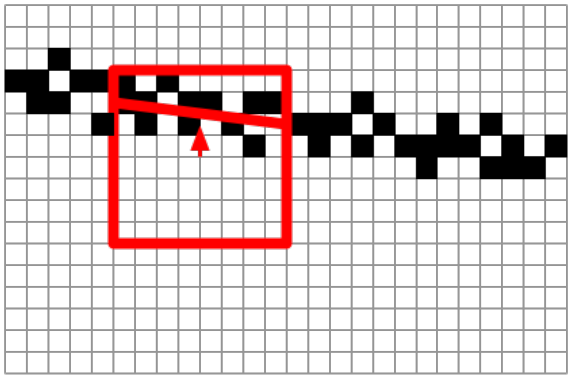

Fig 2: Schematic representation of the window-based regression process. For each window, the offset of the regression line is computed, and the window then contributes to the regression line at a location that is shifted accordingly. The red arrow represents the offset b.

It would be computationally prohibitive to compute the regression coefficients by explicitly extracting coordinates from each selected raster point and aggregating the necessary sums and sums of squares of *x* and *y* values as it would be done for the standard solution of regression

problems. Since the possible *x* and *y* values are located on a grid, we were able to use a formalism that we developed for curvature computations in digital elevation models (Gomes et al. 2019). While the details of the necessary coefficients for spatial regression are different from curvature computations, both depend on aggregates of the form <*zx*>, <*zy*>, <*zxx*>, and <*zxy*>, where in the spatial regression case *z* is 1 when the raster point has a value of ExG that exceeds the threshold and 0 otherwise. x and *y* are coordinates in the reference system of the sliding window.

A key observation for making the computation of the aggregates computationally feasible is the additivity of all aggregates. The aggregate of a window of size 4x4 can be computed from the four aggregates of the constituent 2x2 windows. We start by aggregating all 2x2 windows and then, in a second iteration aggregate 4x4 windows from the results of specifically the four 2x2 windows that fill the 4x4 window. For mathematical details on this aggregation process, please refer to (Gomes et al. 2019).

Experiments

For the experiments, we used two images of corn plants with 1 cm resolution from the Carrington Research Extension Center, Carrington, ND. They were captured with a DJI Matrice hexacopter, flown at 350 ft. AGL (above ground level), and equipped with Sony Alpha 7R II camera (sensor size 7952 × 5304 pixels, 42.4 megapixels). The images were taken in North Dakota on June 14, 2021. Details about the image acquisition and processing can be found in (Sapkota and Flores 2022). We used 1024 x 1024 sized sections for inspection, but the analysis scales to larger images, as can be seen in the performance analysis. We first considered an image that is largely weed-free but has some irregularities due to differences in row spacing and a tractor traversing the field diagonally as seen in Fig. 3. The left panel represents the ExG applied to the RGB bands of the original image, with a threshold of 50 units. Note that we used the original ExG definition, which is also used by the QGIS software and does not include a band-level normalization. For that reason, thresholds are in the original units, for which each band varied between 0 and 255.

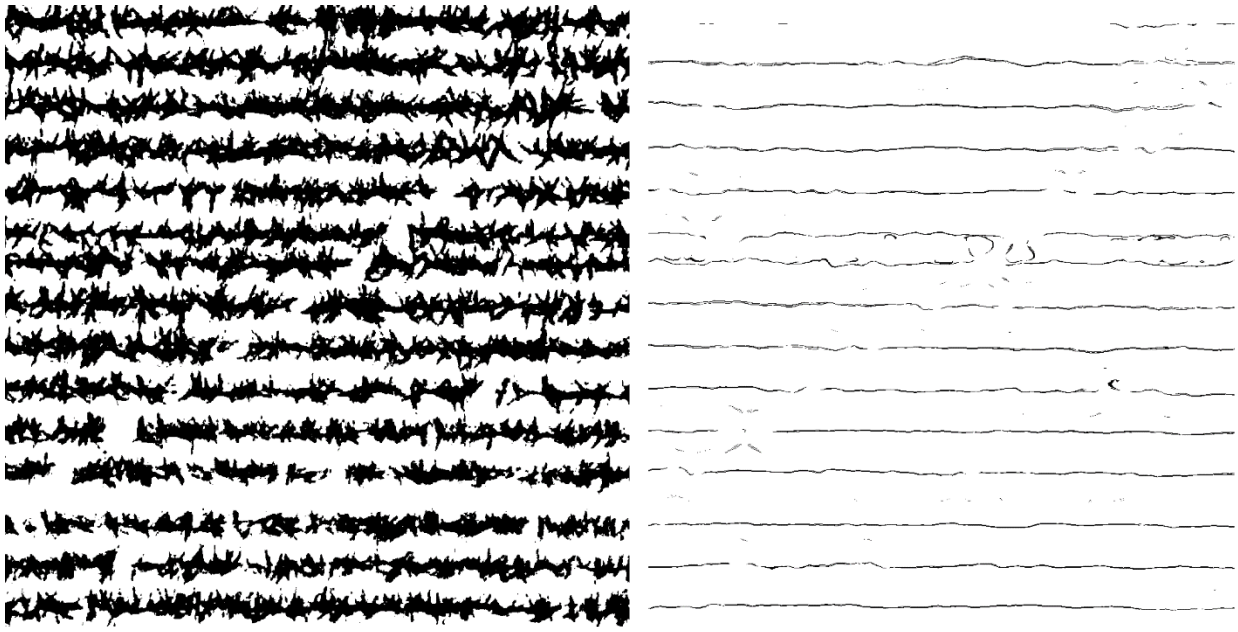

Fig 3: ExG (left panel) and window-based regression result (right panel) for a 1024x1024 section of a corn field with a raster point size of 1 cm with threshold 50.

Processing consists of two passes over the data. In the first pass, linear regression over the window is performed and the shifted center for the window is recorded, provided it satisfies the filter requirements. Images with a regression center ≥ 4 raster points from the window center are filtered out, and so are windows with $N \Sigma y^2 > 20(\Sigma y)^2$ where N is the sum of non-zero raster points and the summations over *y* and y^2 are likewise taken over non-zero raster points. After identifying the shifted centers, a second smoothing step is performed, using a 4x4 sliding window. Only those

results are reported, for which at least 3 of the 16 raster points in any given 4x4 window are non-zero. This final smoothing step is included for eliminating noise.

The right panel in Fig. 3 shows the line fit, with every raster point representing one window of size 64 x 64 raster points. Because of the sliding window evaluation, a frame of 32 raster points is unavailable along the edges of the image in the right panel. This is unavoidable, because no complete 64 x 64 windows are available for analysis in this region. Any window-based algorithm has this problem although some GIS algorithms impute estimates to maintain an image that is of equal size as the original. We did not do that because of the impact on the quality of the results in those regions. One consequence of the lost frame is that the top crop row that is visible in the left panel of Fig. 3 is largely invisible in the right one.

It can be seen in the right panel of Fig. 3 that the crop rows are reconstructed with few gaps and are represented by relatively straight lines of a width much smaller width than the size of the plants themselves. This is what was expected by the regression process and smoothing. While a larger smoothing window would help with noise reduction, it would also result in thicker output lines. Note that the resulting lines have a relatively consistent width, and the shapes of the plants are no longer visible. In some cases, close to the diagonal tractor path, the window-based regression picked up lines in a different direction than the actual crop rows, which can be expected, considering that the window size is smaller than the separation between rows. For future work, we are considering the use of elongated windows that are substantially larger parallel to the rows than perpendicular to them.

To test the algorithm under more realistic conditions, we considered the image in Fig. 4, with substantial weed coverage. Some of the weeds, especially in the bottom and top left are centered between the crop rows, while a smaller number gets close to the crop rows themselves, especially in regions with small stand count.

Fig 4: UAV image of corn rows with resolution of 1 cm. Besides the crop rows, weeds can be identified in the top left, center, and bottom right.

The result of computing the ExG indicator together with the window-based regression result can be seen in Fig. 5. Overall, crop rows are still easily recognizable and represented through lines of regression points that stretch across much of the width of the image. In some places where weeds appeared clustered in the space between rows, the window-based regression algorithm picked them up in ways that look similar to the crop rows themselves. In other cases, where weeds approach the crop rows, the regression process uses both crop and weed plants together. Similar problems were observed by Bah et al. (2017) who show that when a local representation of crops is constructed, it is affected by the weeds themselves. Further processing would have to exclude those systematically.

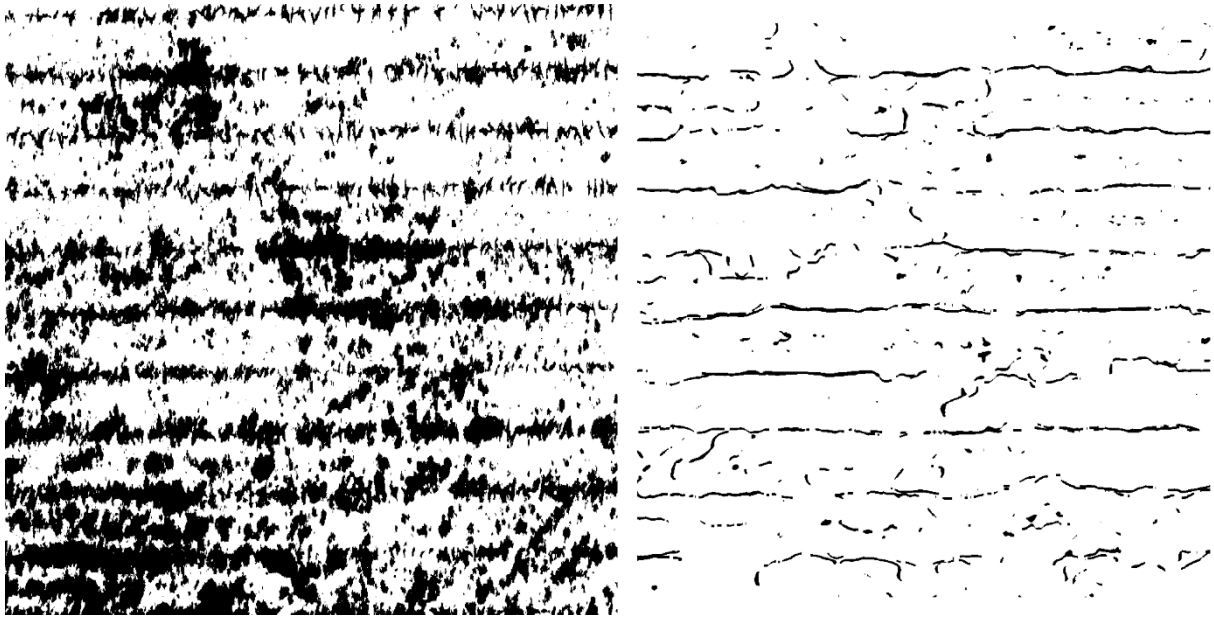

Fig. 5: ExG (left panel) and window-based regression result (right panel) for a 1024x1024 section of a corn field with a raster point size of 1 cm with threshold 15.

Since the goal of the algorithm is to extract crop rows rather than the weeds, we redid the analysis with a larger threshold that excludes more of the weeds. Fig. 6 shows the result. The left panel is substantially lighter than in Fig. 6. Yet, the lines are still largely represented in the right panel. The patches of randomly distributed weeds are now reduced in frequency as expected. Unfortunately, extra rows due to the weeds that are clustered around the centers between crop rows are still visible.

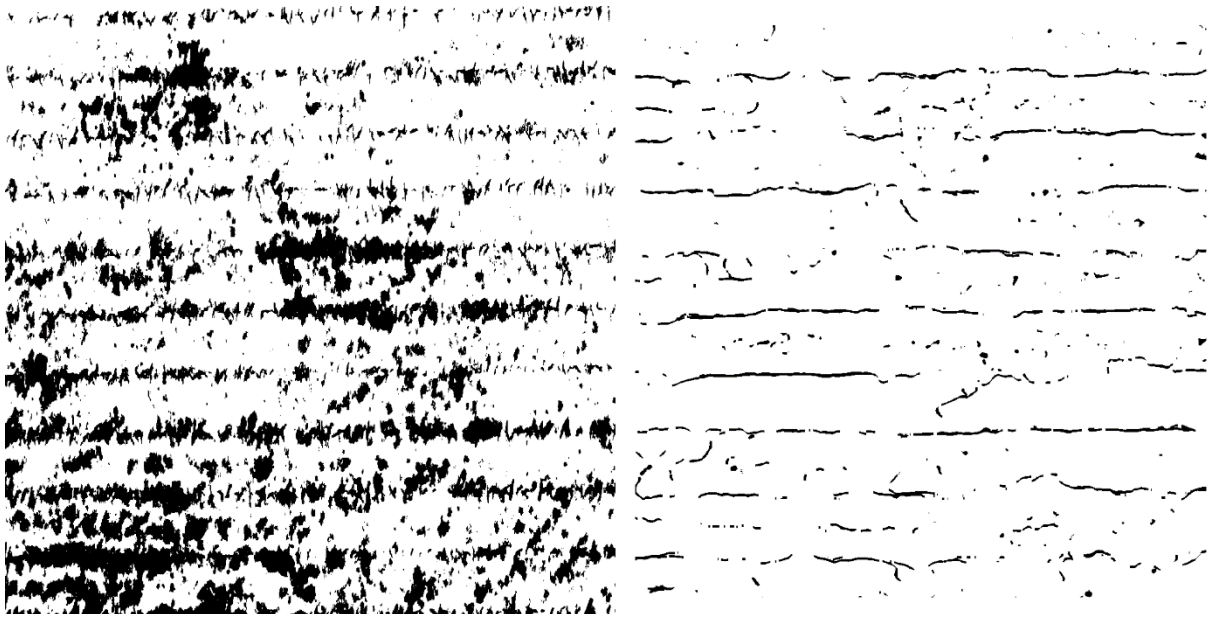

Fig. 6: ExG (left panel) and window-based regression result (right panel) for a 1024x1024 section of a corn field with a raster point size of 1 cm with threshold 25.

For comparison, we used the HLT algorithm in the OpenCV Python library (OpenCV 2022). The result can be seen in Fig. 7. For some rows, there is a broad spectrum of lines that can fit the existing data, and in some cases the lines are at an angle with the actual crop rows. Instead of capturing the curved paths of the crop rows that were visible in the right panel if Fig. 6 this approach simply returns several alternative lines.

Fig. 7 HLT applied directly to thresholded ExG image of the field, using threshold 25 as in Fig. 6.

Although the ultimate intent of our work is to develop an algorithm that represents crop rows through curves that do not have to be straight lines, we also experimented with following a process similar to (Bah 2017), and applied HLT to the output of the window-based regression algorithm. The result can be found in Fig. 8. All rows are now recognized correctly, and there are fewer lines returned for each crop row. As such, the window-based regression algorithm succeeds as some of the goals mentioned in (Bah 2017). However, there are still typically multiple fit lines shown, some of which diverge especially towards the edges of the image. Going forward, these lines may help distinguish results associated with crop rows from ones that are not, as we are pursuing the goal of using a model of crop rows that is more flexible than straight lines in the future.

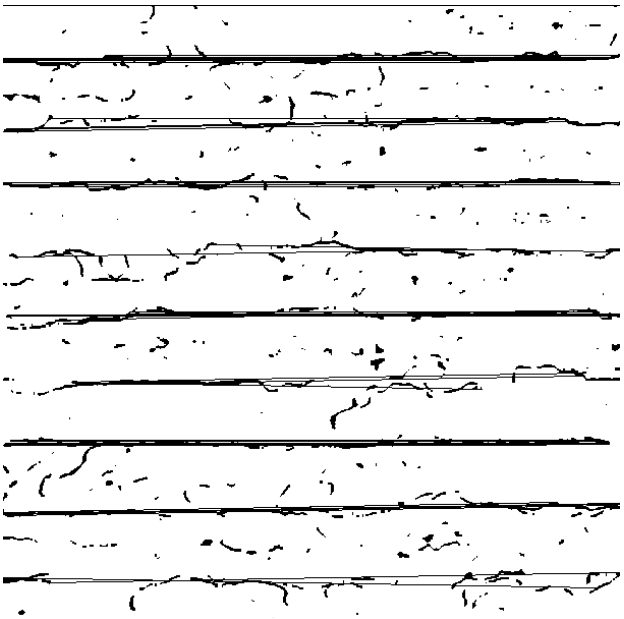

Fig. 8 HLT applied to the output of the window-based regression algorithm, right panel in Fig. 6.

Finally, we tested the speed of the presented window-based regression algorithm. Fig. 9 shows the result for the 1024 x 1024- sized images considered in the evaluation. At the window size of 64 x 64 that is used throughout the evaluation, the runtime is approximately 2 s. For an image of the same size, the HLT code takes 0.27 seconds, but only returns results for the entire image. The window-based evaluation therefore has a relatively small impact on the runtime. Fig. 9 shows that the total runtime increases by not even a factor of 2 between window sizes of 4 x 4 and 64 x

64. The decrease in runtime for the largest window size is a result of the reduced output size due to the smaller number of complete sub-windows within the image.

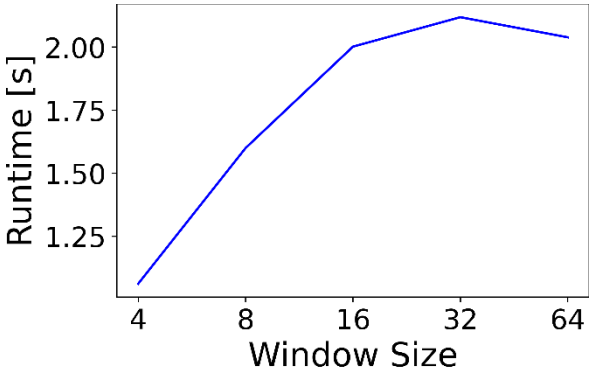

Fig. 9 Runtime for an image of size 1024 x 1024. The actual window size throughout the evaluation is 64 x 64. Other window sizes are included to demonstrate the scaling.

Conclusions

In summary, we have shown the promise of using spatial window-based regression over sliding windows towards the crop row detection in corn fields. With the goal of reducing errors in the identification of weeds, we work towards a model of crop rows that attempts to infer the centers of the rows. We do so by using linear regression over those raster points that exceed a threshold in the ExG value. The regression is done efficiently using an aggregation strategy that allows simultaneously computing all window-based aggregates with a small number of passes over the image, specifically as many as the binary logarithm of the window size. We show that the resulting window-based regression result follows the crop rows well with relatively few gaps, but also still captures weeds somewhat. One alternative for distinguishing crop- and weed-related regression results would be to draw on existing techniques of using the aid of a Hough line transform. In summary, we have shown the value of using a systematic window-based regression approach towards identifying crop rows in UAV imagery towards the goal of weed identification.

References

Bah, M. D., Hafiane, A., & Canals, R. (2017). Weeds detection in UAV imagery using SLIC and the hough transform. 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), 1–6.

Bakker, T., Wouters, H., van Asselt, K., Bontsema, J., Tang, L., Müller, J., & van Straten, G. (2008). A vision based row detection system for sugar beet. Computers and Electronics in Agriculture, 60(1), 87–95.

Chavan, H., Denton, A.M., Franzen, D., Nowatzki, D. (2016) Window-based regression analysis of field data. Proc. of the International Conference for Precision Agriculture

Chen, J., Qiang, H., Wu, J., Xu, G., & Wang, Z. (2021). Navigation path extraction for greenhouse cucumber-picking robots using the prediction-point Hough transform. Computers and Electronics in Agriculture, 180, 105911.

Gomes, R., Denton, A. and Franzen, D.W. (2019). Quantifying Efficiency of Sliding-Window Based Aggregation Technique by Using Predictive Modeling on Landform Attributes Derived from DEM and NDVI. ISPRS Int. J. Geo-Information 8(4): 196.

Leemans, V., & Destain, M.-F. (2006). Line cluster detection using a variant of the Hough transform for culture row localisation. Image and Vision Computing, 24(5), 541–550.

OpenCV (2022). Open Source Computer Vision. https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html

Sapkota, R., & Flores, P. (2022). UAS Imagery and Computer Vision for Site-Specific Weed Control in Corn. arXiv preprint arXiv:2204.12417v1.

Soares, G. A., Abdala, D. D., & Escarpinati, M. C. (2018a). Plantation Rows Identification by Means of Image Tiling and Hough Transform. VISIGRAPP (4: VISAPP), 453–459.

Woebbecke, D. M., Meyer, G. E., Von Bargen, K., & Mortensen, D. A. (1995). Color indices for weed identification under various soil, residue, and lighting conditions. Transactions of the ASAE, 38(1), 259-269.