Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-16 (2016) 231–236

Spatial Map Generation from Low Cost Ground Vehicle Mounted Monocular Camera

Stephen Cossell * **Mark Whitty** * **Scarlett Liu** * **Julie Tang** *

* {s.cossell, m.whitty, sisi.liu, julie.tang}@unsw.edu.au
School of Mechanical and Manufacturing Engineering,
The University of New South Wales, Australia

Abstract: This paper presents a method for generating a spatial map of a particular plant or environmental property of a vineyard block based on low cost camera technology and existing vineyard vehicles. Such properties can range from leaf area, per vine bunch count or bare-wire detection. The paper provides a low cost ground vehicle based solution that does not rely on live GPS position recording. Rather, the relative estimated motion between video frames is used to localize each sensor reading within the bounds of each row. Row end locations are derived from post-processed GPS recorded locations of the perimeter of a block with an aerial photograph. This paper uses the proportion of leaf colored pixels in a video frame as a token example of measuring the relative growth of vines during the shoots stage.

Keywords: computer vision, viticulture, spatial mapping, proximal sensing

1. INTRODUCTION

The generation of spatial maps of certain plant properties within a vineyard is vital to management practices employed by vineyard managers. It is one of the key tools to assist in the practice of *precision viticulture* (Cook and Bramley (1998)). For example, a vine vigor map (Hall et al. (2001)) shows a vineyard manager the relative health of vines spatially. A vineyard manager may also want to know not only the percentage of the block that consists of bare wire, but also the approximate location of such non-productive regions. Yield prediction is also a vital practice that has recently become well adopted from the precision agriculture community. Jarrett et al. (2014) mention that a spatial measure of yield is vital to maintain a desired standard of fruit quality. Liu et al. (2015) also presented work that detected the proportion of shoots within a vineyard block to provide an early season yield forecast.

Many spatial data collection and mapping technologies exist, but the majority require an aerial vehicle or satellite imagery. Hall et al. (2003) produced a method of mapping pixel locations in aerial photography to individual vines within a given row. Johnson et al. (2003) and Johnson et al. (1996) presented a method to convert geo-referenced satellite images to a leaf area index to spatially map plant growth and specifically mention that manual ground based measurements are not suitable for vineyard managers with large scale sites. Johnson et al. (2003) make the assumption that manual measurements are performed destructively on foot, whereas the proposed method can capture chosen sensor data at 35 minutes per hectare on average.

A small amount of research is available on proximal ground based sensor systems. Rubio and Más (2013) presented a system for estimating vine vigor using an over the row IR sensor to detect the proportion of reflectance of leaves. This uses a similar technique of indirect vigor measurement as the Plant Cell Density (PCD) map. Their method used an on board GPS unit to approximate location, but could only generate a spatial map with $25m^2$ grid cell resolution. The method provided in this paper is capable of localization errors in the order of $2m$–$3m$ without no other sensor than a single camera.

The main objective of this research is to introduce a method for producing spatially self-consistent maps with low-cost hardware. Only a georeferenced block outline and single camera are required, as opposed to expensive and complicated positioning solutions such as GPS, IMUs, or wheel encoders. The example used in this paper is to detect and map the size of vine canopy throughout the block using a single low-cost camera, which relates indirectly to vine health and vine balance.

This paper begins by detailing the low cost method of localizing sensor measurements. Assumptions and limitations are discussed before results are presented using a proxy measurement for leaf area — the proportion of *leaf colored* or green pixels within video frames themselves.

2. METHOD

A spatial map is generated using a low cost portable video camera, an existing tractor or vehicle, and a pre-defined driving pattern. This technique can be used to generate spatial maps of any sensor reading, however for this paper, the amount of leaves in a video frame is estimated during the shoot stage of phonological growth using a basic measurement of the amount of *leaf colored* pixels in each

⋆ Funding and resources provided by Jarrett's of Orange, Treasury Wine Estates and Wine Australia through project DPI 1401.

10.1016/j.ifacol.2016.10.043

Fig. 1. Mounting configuration of the GoPro camera.

frame. This section outlines the experimental equipment, collection process and processing required to turn video footage into a proper spatial map.

2.1 Video Recording Equipment and Procedure

A GoPro Hero 3+ camera was mounted on the side of a vehicle as shown in Figure 1. The camera was mounted horizontally facing the cordon with the aim to capture the majority of the immediate vine as centered in the camera's vertical field of view. Video was captured at 30 frames per second and the Medium (**M**) field of view option was selected, giving a horizontal and vertical angle of view of 94.4^o and 72.2^o, respectively (GoPro (accessed 2016)). The fish-eye distortion of captured frames was removed as the initial step of any video frame analysis.

Video recording was completed on a per row basis. For each row of the predetermined driving pattern, a vehicle was parked just before the beginning of a pair of rows. Video recording was started and confirmed to be running before the vehicle was driven the length of the row trying to maintain 10km/h. The vehicle was driven past the end of the last post before stopping. Video recording was then stopped while the vehicle was stationary before the vehicle was driven to just outside the next pair of rows in the driving pattern, and the process repeated.

2.2 Driving Patterns

Three different driving patterns were developed to assist in mapping a particular video file to the row, with the driving direction and side of vine being recorded. These are referred to as:

- single-pass
- double-pass single-sensor
- double-pass double-sensor

The *single-pass* driving pattern is used under conditions where a general spread of sensor readings were required across the block, but not every side of every vine required recording. In this driving pattern a vehicle starts at a predefined zero end, for example the Northern end of a block, and drives between rows 1 and 2. The vehicle then

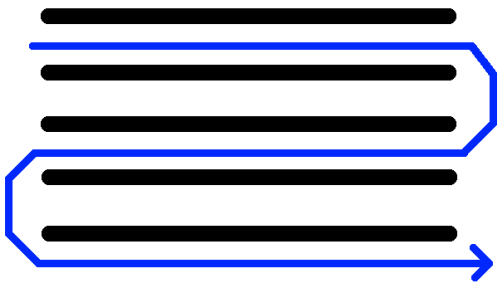

Fig. 2. The *single-pass* driving pattern.

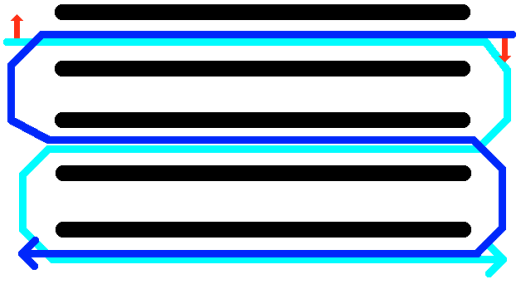

Fig. 3. The *double-pass single-sensor* driving pattern. The lighter path shows the first pass, while the darker path shows the second pass. Red arrows indicate the direction of the sensor for each pass if it were mounted on the left side of the vehicle.

records between rows 3 and 4 driving in the opposite direction, and so on. Figure 2 demonstrates the row order and direction for the *single-pass* driving pattern.

The *double-pass single-sensor* driving pattern is used under conditions where only a single sensor is available on one side of the vehicle, with at least one side of every vine being required to be recorded. Here the *single-pass* driving pattern is used as a first pass of the block. Then a modified *single-pass* pattern is used for a second pass, where every pair of rows is recorded in the opposite direction to the initial pass. This pattern ensures that a sensor mounted on a particular side of the vehicle is able to sense at least one side of every vine. Figure 3 demonstrates the row order and direction for the *double-pass single-sensor* driving pattern.

The *double-pass double-sensor* driving pattern is used under conditions where two sensors are available, one on each side of the vehicle and both sides of every vine are required to be recorded. Here the *single-pass* driving pattern was again used for a first pass of the block. Then as a second pass, the *single-pass* driving pattern is used starting from the same predefined zero end of the block, but outside of row 1. That is, the first few pairs of rows recorded on the second pass are outside row 1, rows 2 and 3, then rows 4 and 5. With each driving pattern, if a single row remains unrecorded after all other pairs of rows have been recorded then the remaining row is recorded from the outside, so as to maintain the same progression of row numbers and directions for sensors on a particular side of the vehicle. Figure 4 demonstrates the row order

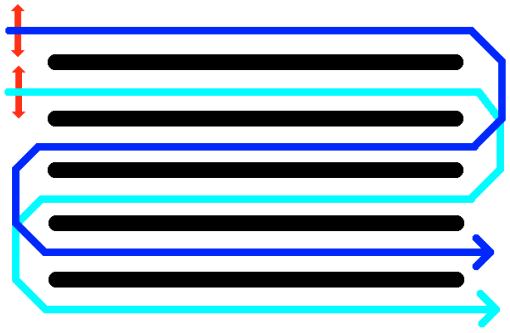

Fig. 4. The *double-pass double-sensor* driving pattern. The lighter path shows the first pass, while the darker path shows the second pass. Red arrows indicate sensor locations.

and direction for the *double-pass double-sensor* driving pattern.

Predefined driving patterns are important when working with off-the-shelf cameras, as it can be difficult to associate any contextual data with a video while it is being recorded, as was the case with the GoPro cameras used in these experiments. When an entire block has been recorded, the resulting video file names comprise a list of automatically incremented indices. If two cameras are used simultaneously on the left and right sides of a vehicle then the operator is left with two groups of such ascendingly named folders of video files. It is desirable for spatial mapping to associate each video file with the row number, the direction of travel (either North to South or South to North in the case of blocks studies in this paper) and the side of the vine being recorded (either East or West).

Given a row number, the following formulae derive key parameters associated with a particular video file for all three driving patterns.

$$D = \begin{cases} \texttt{NtoS} & \text{if } r \bmod 4 \in \{1,2\} \wedge p = 1 \\ \texttt{NtoS} & \text{if } r \bmod 4 \in \{0,3\} \wedge p = 2 \wedge \text{DP-SS} \\ \texttt{NtoS} & \text{if } r \bmod 4 \in \{0,1\} \wedge p = 2 \wedge \text{DP-DS} \\ \texttt{StoN} & \text{otherwise} \end{cases} \quad (1)$$

$$VS = \begin{cases} \texttt{WestSide} & \text{if } r \bmod 2 = 1 \wedge p = 1 \\ \texttt{WestSide} & \text{if } r \bmod 2 = 1 \wedge p = 2 \wedge \text{DP-SS} \\ \texttt{WestSide} & \text{if } r \bmod 2 = 0 \wedge p = 2 \wedge \text{DP-DS} \\ \texttt{EastSide} & \text{otherwise} \end{cases} \quad (2)$$

$$CS = \begin{cases} \texttt{Left} & \text{if } r \bmod 4 \in \{0,1\} \wedge p = 1 \\ \texttt{Left} & \text{if } r \bmod 4 \in \{2,3\} \wedge p = 2 \wedge \text{DP-SS} \\ \texttt{Left} & \text{if } r \bmod 4 \in \{0,3\} \wedge p = 2 \wedge \text{DP-DS} \\ \texttt{Right} & \text{otherwise} \end{cases} \quad (3)$$

$$VN = \begin{cases} \left\lfloor \dfrac{r+1}{2} \right\rfloor & \text{if } p = 1 \\ \left\lfloor \dfrac{R}{2} \right\rfloor + 1 + \left\lfloor \dfrac{r}{2} \right\rfloor & \text{if } p = 2 \end{cases} \quad (4)$$

where:

- **D** is the direction of travel [1],
- **VS** is the vine side,
- **CS** is the camera side,
- **VN** is the video number [2],
- **r** is the row number,
- **R** is the total number of rows in the block,
- **p** is the pass number [3] with $p \in \{1, 2\}$,
- **DP-SS** represents the double-pass single-sensor driving pattern, and
- **DP-DS** represents the double-pass double-sensor driving pattern.

For each row number in a block, the correct raw video file can be accessed via the camera side and video number. Once the raw video file is found, it can then be renamed using the row number, direction and vine side.

2.3 Localization with low cost monocular camera

To be able to generate an accurate spatially self-consistent map, each frame of each video must be localized. This process involves four steps.

(1) The latitude and longitude of end posts of each row are derived offline.
(2) The frames within each video where the start and end posts appear most centered are detected.
(3) The instantaneous velocities while recording video along the row are estimated.
(4) The frame number is mapped to a location via a relative velocity adjusted linear interpolation between end post frames and locations.

To generate the locations of every end post in the given study block, the GPS measured locations of key posts that comprised the perimeter of each block are used [4]. These locations are then strung together to comprise a polygon outline of the block. Next, the polygon is overlaid on top of a georeferenced aerial photograph [5] of the vineyard block and pixels outside of the perimeter are discarded as shown in Figure 5. The orientation of rows is then calculated using Hough line detection via a Canny edge filter on the remaining pixels of the aerial photo (Figure 6). A consensus orientation is then calculated as the average of the 1st and 2nd tertiles [6] of all orientations of significant lines detected. For the study block used in this paper, a histogram of integer angles representing each of the detected intervals is shown in Figure 7. Here 15^o was chosen as the consensus orientation as the 1st and 2nd tertiles were both 15^o.

[1] This assumes that the zero end of the block is the Northern end. Directions here can switch if a block is defined with the Southern end being the starting end of rows.
[2] The video number is the index of a particular video file in a folder when files are listed in ascending order. For example, a video number of 1 would equate to the first file when listed in ascending order.
[3] The first pass of the double-pass patterns and the single-pass pattern are given $p = 1$, otherwise the second pass of a double-pass pattern is given $p = 2$.
[4] It is common for vineyard managers to have the GPS locations of key corner posts in each of their blocks, but not the locations of every start and end post.
[5] Courtesy Google Maps 2015
[6] 3-quantile

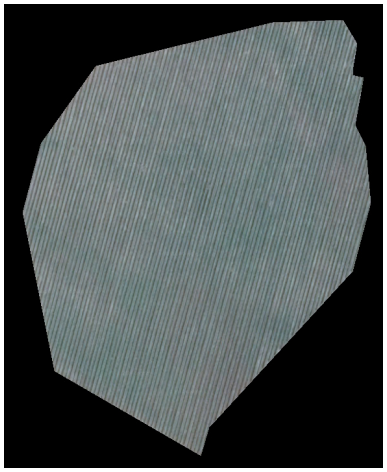

Fig. 5. Aerial photograph of vineyard block cut out from georeferenced perimeter polygon.

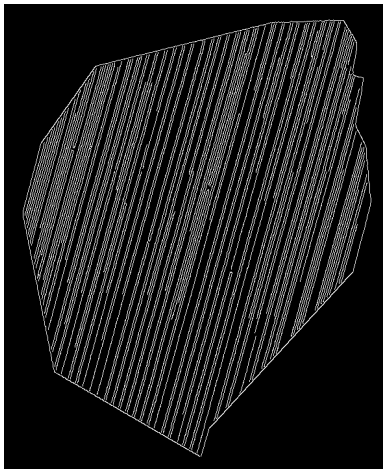

Fig. 6. Canny edge filter applied to aerial photograph.

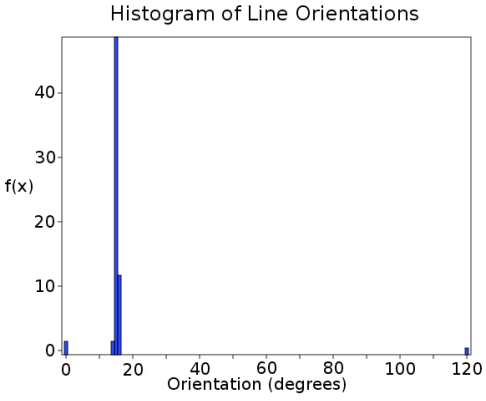

Fig. 7. Histogram of detected line orientations in degrees from the horizontal.

The first and last rows are then detected by comparing the orientation of all sides of the polygon to the consensus orientation, with the two closest matches kept. Figure 8 shows the calculated orientations of each polygon interval along side ticks in the direction of the consensus angle. A latitude-longitude point situated somewhere along each row is then calculated by linearly interpolating between an end post on the first row and an end post on the last

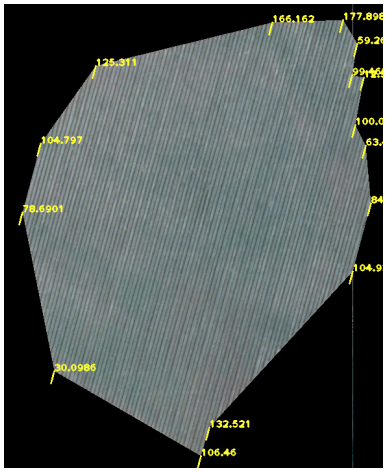

Fig. 8. Slope angles of each perimeter polygon interval.

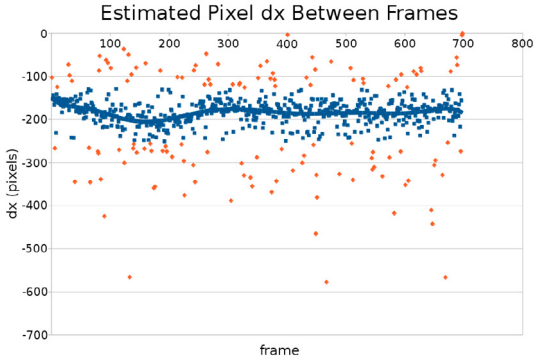

Fig. 9. Distribution of estimated pixel differences between frames. Red diamonds are estimated dx values outside the $\mu \pm \sigma$ range. Blue squares are within $\mu \pm \sigma$, with the heavy blue line representing the 10th degree polynomial approximation. Given dx values in this figure are negative as objects in the camera's view are moving from left to right for this particular recording.

row, with increment size relative to the known number of rows in that block. Each row point is then combined with the consensus orientation to detect any points of intersection with the row line and the intervals of the perimeter polygon. These points of intersection are then given as the row start and end post locations. It should be noted that this method of row end location calculation assumes straight rows.

Each row video is then processed to detect both the start and end post frames and estimate the instantaneous velocities across the video. This was required because a constant velocity cannot be assumed due to variation in the driving conditions along a row (as shown in Figure 9. First, the distance travelled between subsequent frames dx is approximated by matching dominant blobs in adjacent frames of the video. Next, the mean μ and standard deviation σ of all dx values were calculated and any dx values situated outside the range $\mu \pm \sigma$ were culled. A 10th degree polynomial is then fitted to the remaining points and this curve is used as the approximation of the distance travelled between frames along a row. Figure 9 shows the components used to approximate movement along a particular row.

Given the 10th degree polynomial approximation, corrected dx values between all pairs of frames are then recalculated based on corresponding polynomial values. The linear interpolation variable t_i is then calculated for frame i based on the sum of all corrected dx values up to frame i, normalized relative to the sum of all corrected dx values for the row, as given by Equation 5.

$$t_i = \frac{\sum_{p=1}^{i} dx_p}{\sum dx} \tag{5}$$

The corrected approximation of distance travelled along a row d at frame i can then be calculated by Equation 6.

$$d_i = t_i \times \text{length of row} \tag{6}$$

Given a distance d along a row and the derived latitude-longitude locations of start and end posts, the approximate location of a frame can be calculated using the haversine formula (Veness (2011)). Sensor readings associated via time with a given frame and known video frame rate can then be assigned a latitude-longitude location.

2.4 Map Generation

Once sensor readings have been mapped to an approximate latitude-longitude location via the video and frame number, results can be collected for each frame and graphed spatially. Figure 10 shows the measurement of *leaf colored* pixels in video frames measured across the block. For this particular application of the method, *leaf colored* pixels are defined as pixels that satisfy all of the following requirements in HSV color space.

- $80^o \leq H \leq 120^o$
- $S > 20\%$
- $V > 50\%$

Blue colors represent the lowest counts, with reds representing the highest. Figure 11 shows the same values grouped into three equally sized tertiles[7], with maroon representing the lowest $\frac{1}{3}$ of values and yellow representing the highest $\frac{1}{3}$ of values. Values are derived throughout the spatial map using inverse distance weighted approximation with power parameter $p = 2$.

3. RESULTS

To properly quantify the accuracy of location estimation within a vineyard row, two methods were used to predict post locations within a selected row. As a baseline, the distance between every post in row 21 of block B4 was measured to the nearest centimeter, with the center of each post used as the location. This particular row contains 46 posts (45 panels) and is 310.4m long.

The first method of location estimation used linear interpolation between start and end post frames, f_s and f_e respectively, with the interpolation variable t at frame i given by Equation 7.

[7] 3-quantile

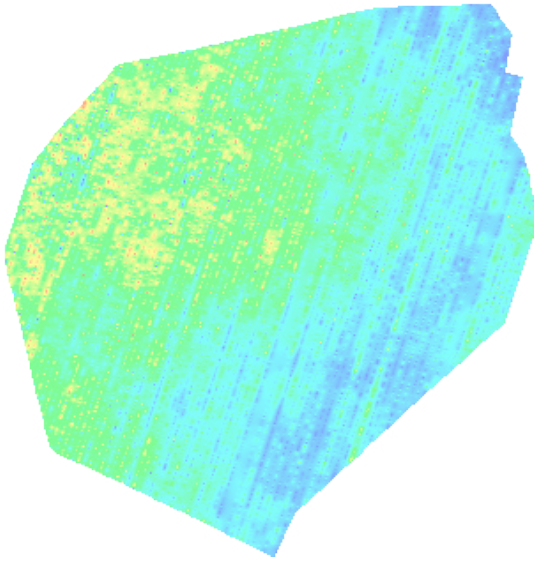

Fig. 10. Spatial map of approximated shoot count for block 40A. Colors show a linear mapping from blue at low levels to red at high levels.

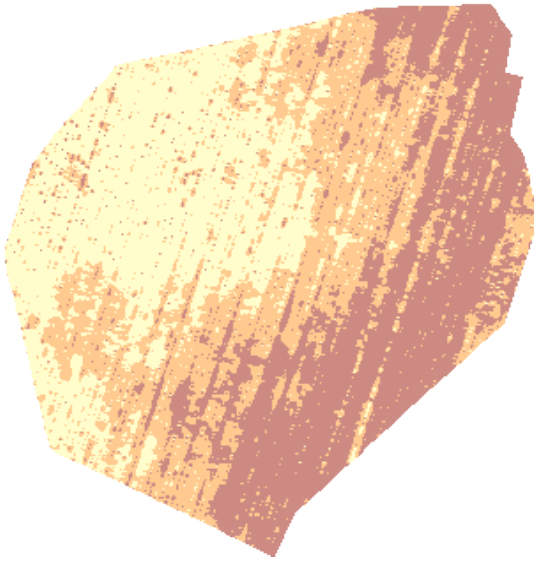

Fig. 11. Spatial map of approximated shoot counts for block 40A grouped into low (brown), medium (orange) and high (yellow) bands.

$$t_i = \frac{f_i - f_s}{f_e - f_s} \tag{7}$$

The second method used the velocity corrected interpolation variable t_i from Equation 5. For each method, all 46 post locations within the baselined row were estimated, with the errors of estimation relative to ground measurements given in Figure 12. Although the velocity corrected approach provided a more accurate estimation of location, it is hypothesized that a more robust feature matching algorithm would greatly improve the inter-frame dx estimation values over major blob matching used in this paper. This would provide lower estimate errors relative to reality. Figure 13 demonstrates spatially how the two methods estimate corresponding post locations relative to real measurements, with absolute errors quantified through the use

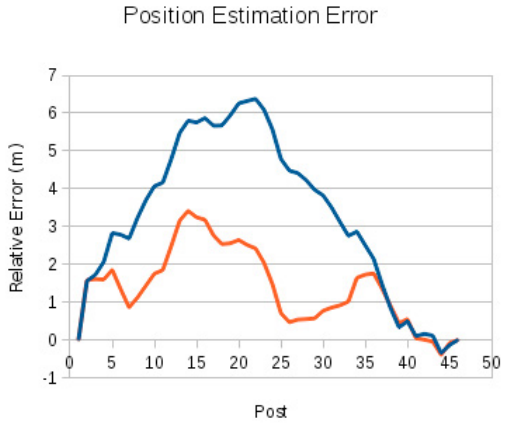

Fig. 12. Comparison of the error in estimating post locations along a recorded row. The blue line represents location errors based on naïve linear interpolation via the frame number of the video, while the red line represents velocity corrected estimates of post location.

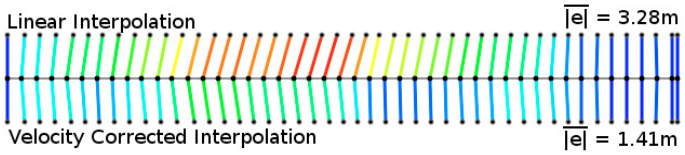

Fig. 13. Comparison of the linear and velocity corrected post locations for block 12, row 21 relative to real post locations. The center line shows real post locations, with linear approximations above and velocity corrected below the center line, respectively. The color spectrum is used to convey the absolute magnitude of the error with red representing the maximum error of 6.4m and blue representing zero error.

of the color spectrum. The mean absolute error between linear and velocity corrected methods is 3.28m and 1.41m, respectively.

4. CONCLUSION

This paper has presented a novel method for producing spatially self-consistent maps with low-cost hardware using only a a georeferenced block outline and a single camera. The camera is mounted on the side of a vehicle to measure the distance travelled along each row while recording other sensory data. Each frame of the recorded video can be associated with a particular row and distance along that row, which can be mapped to an approximate latitude-longitude location through an improved velocity-corrected interpolation. A spatial map of the sensor readings can then be generated to give farmers an improved view of critical plant and environment properties. The authors believe that localization within a row can be further improved by both a more robust inter-frame feature matching approach, as well as automatic post detection within frames. Where posts in the block exist and are spaced evenly, their location could be used to assist in the localization, as shown by Nuske et al. (2011).

ACKNOWLEDGEMENTS

The authors would like to thank Jarrett's of Orange and Treasury Wine Estates for resources and facilities provided for data capture, financial support from Wine Australia through project DPI 1401. The authors would also like to thank Gregory Dunn, Paul Petrie, Angus Davidson and Catherine Wotton for assistance with data collection and Stephen Lin for his assistance with manually labelling data.

REFERENCES

Cook, S. and Bramley, R. (1998). Precision agriculture-opportunities, benefits and pitfalls of site-specific crop management in australia. *Animal Production Science*, 38(7), 753–763.

GoPro (accessed 2016). Field of view information. `https://gopro.com/support/articles/hero3-field-of-view-fov-information`.

Hall, A., Louis, J., and Lamb, D. (2003). Characterising and mapping vineyard canopy using high-spatial-resolution aerial multispectral images. *Computers and Geociences*, 29(7), 813–822.

Hall, A., Louis, J., and Lamb, D. (2001). A method for extracting detailed information from high resolution multispectral images of vineyards. In *Proceedings of the 6th International Conference on Geocomputation*, 24–26.

Jarrett, J. et al. (2014). You need to get to know your GLO. *Australian and New Zealand Grapegrower and Winemaker*, (609), 28.

Johnson, L., Lobitz, B., Armstrong, R., Baldy, R., Weber, E., De Benedictis, J., Bosch, D., et al. (1996). Airborne imaging aids vineyard canopy evaluation. *California Agriculture*, 50(4), 14–18.

Johnson, L., Roczen, D., Youkhana, S., Nemani, R., and Bosch, D. (2003). Mapping vineyard leaf area with ultispectral satellite imagery. *Computers and electronics in argiculture*, 38(1), 33–44.

Liu, S., Tang, J., Cossell, S., and Whitty, M. (2015). Detection of shoots in vineyards by unsupervised learning with over the row computer vision system. In *Australasian Conference on Robotics and Automation, Proceedings of*. Canberra, Australia.

Nuske, S., Achar, S., Gupta, K., Narasimhan, S., and Singh, S. (2011). Visual yield estimation in vineyards: Experiments with different varietals and calibration procedures. Technical Report CMU-RI-TR-11-39, Robotics Institute, Carnegie Mellon University, USA.

Rubio, V.S. and Más, F.R. (2013). Proximal sensing mapping method to generate field maps in vineyards. *Agricultural Engineering International: CIGR Journal*, 15(2), 47–59.

Veness, C. (2011). Calculate distance and bearing between two latitude/longitude points using haversine formula in javascript. *Moveable Type Scripts*.