

# A Vision-Based Robust Grape Berry Counting Algorithm for Fast Calibration-free Bunch Weight Estimation in the Field

Scarlett Liu<sup>a,b</sup>, Xiangdong Zeng<sup>a</sup>, Mark Whitty<sup>b,\*</sup>

<sup>a</sup>*School of Traffic and Transportation Engineering, Central South University,  
Changsha, Hunan 410075, China*

<sup>b</sup>*School of Mechanical and Manufacturing Engineering, UNSW Sydney, NSW 2052,  
Australia*

---

## Abstract

1 Counting the number of berries per bunch is a key component of many yield  
2 estimation processes but is exceptionally tedious for farmers to complete.  
3 Recent work into image processing in viticulture has produced methods for  
4 berry counting, however these require some degree of manual intervention  
5 or need calibration to manual counts for different bunch architectures.

6 Therefore, this paper introduces a fast and robust calibration-free algo-  
7 rithm for berry counting for winegrapes to aid yield estimation. The algo-  
8 rithm was tested on 529 images collected in the field at multiple vineyards  
9 at different maturity stages and achieved an accuracy of approximately 89%  
10 per bunch. As it would mostly likely be used to obtain an average value  
11 across a block, the low bias of this method resulted in an average accuracy  
12 of 99% and was shown to be robust from pea-sized to harvest and between  
13 both red and green bunches.

14 Taking only 0.1 to 1 second per image to process and requiring only a  
15 smartphone and small backing board to capture, the algorithm can readily  
16 be applied to images which are captured in the field by farmers. This allowed

17 bunch weights to be estimated to within 92% accuracy and assisted larger  
18 scale yield estimation processes to achieve accuracies of between 3% and  
19 16%. The robustness of the method lays the foundation for fast fruit-set  
20 ratio determination and more detailed bunch architecture studies in-vivo on  
21 a large scale.

*Keywords:* grape bunch, bunch reconstruction, image processing, berry counting, bunch weight, yield estimation

---

22 **1. Introduction**

23 Automating yield component analysis is vital for improving yield esti-  
24 mation in viticulture since the current manual approaches can not meet the  
25 requirements of fast measurement and large sampling scale to secure the  
26 accuracy of grape production forecasting. The small sample size and lack of  
27 objectivity in interpreting the state of vine development also leads to poor  
28 accuracy in yield estimation in the wine industry. State-of-the-art manual  
29 sampling immediately prior to harvest result in errors from 3 to 30% [1]  
30 (Table 6.9), which anecdotally matches industry experience. Subsequently,  
31 wineries are forced to bear the cost of suboptimal tank space allocation,  
32 oak barrel purchases and contract adjustments as well as undertake the  
33 challenging task of managing harvest logistics within a decreasing harvest  
34 window. Hence, researchers in viticulture have been seeking solutions from  
35 image processing and computer vision to accelerate crop yield forecasting.  
36 Nuske *et al.* [2] presented image processing methods which were able to

---

\*Corresponding author

Email addresses: [scarlett.liu.academic@gmail.com](mailto:scarlett.liu.academic@gmail.com) (Scarlett Liu),  
[eastward219@gmail.com](mailto:eastward219@gmail.com) (Xiangdong Zeng), [m.whitty@unsw.edu.au](mailto:m.whitty@unsw.edu.au) (Mark Whitty)  
Preprint submitted to *Computers and Electronics in Agriculture* May 26, 2020

37 generate unbiased estimates notably smaller than manual estimates.

38

39 Image processing has also been applied for grape bunch phenotyping  
40 in the context of breeding programs, and such phenotyping includes three  
41 quantitative methods of analysis [3]; the number of components, overall  
42 morphology of components and overall length of components, with compo-  
43 nents being berries, rachis internodes and other internodes *etc.* The number  
44 of berries remains stable after fruit set and it has vital impact on final yield  
45 for a bunch (Martin *et al.* 2003 [4]). Besides that, the ratio between bunch  
46 size and the number of berries per bunch is one of many factors governing  
47 the quality of the fruit at harvest. Given the number of berries per bunch  
48 and bunch weight are critical parameters for early forecasts of production  
49 we mainly focus on the yield component of berry number and its contribu-  
50 tion to bunch yield estimation in this paper.

51

52 Currently, berry counting and bunch weighing across the grape growing  
53 season are accomplished by tedious and labour intensive manual measure-  
54 ments. To expedite this, two main approaches have been used: image based  
55 (2D, RGB images) berry counting and 3D point cloud sensor-based berry  
56 counting (by laser or RGB-D camera).

57

58 Under the category of 3D point cloud sensor-based methods, initial work  
59 was conducted by Florian and Volker [3] who presented a fully-automated  
60 sensor-based 3D reconstruction approach to phenotyping grape bunches.  
61 The proposed approach is able to generate a comprehensive bunch struc-  
62 ture based on the 3D point cloud by iteratively optimizing parameters which

63 define the bunch structure. As to berry number estimation, their approach  
64 was shown to achieve 12.35% error (see Table 1 in that work [3]). Later on,  
65 the same research group extended this work by developing software called  
66 "3D-Bunch-Tool" based on new lightweight 3D scanner [5] which can be uti-  
67 lized in the field. That software achieved 78.83% accuracy with  $R^2 = 0.95$   
68 ((see Table 2 in that work[5]) on lab-based berry counting and the process  
69 of scanning in the lab took approximately 1 minute. Field based scanning,  
70 observing only one side of a bunch meant approximately 50% of berries were  
71 observed, and these were correlated to the total number of berries from a  
72 360deg scan with an  $R^2$  value of 0.83; the actual error in terms of berry  
73 count was not presented in the paper.

74

75 The intricacy of the 3D scanning approach and cost of the sensors has  
76 meant considerable focus has been given to the imagery based solution in  
77 the field. Liu *et al.* [6], Diago *et al.* [7] and Ivorra *et al.* [8] showed  
78 how yield component analysis could lead to more efficient forecasts using  
79 image processing. Kicherer *et al.* [9] presented the Berry Analysis Tool  
80 (BAT) for counting berry number, diameter and volume, which is reliant  
81 on destemming a bunch and arranging berries on a perforated metal plate  
82 in laboratory conditions. Grosssetete *et al.* [10] and later Diago *et al.* [7]  
83 processed RGB images to count berries using a photo of one side of a bunch,  
84 obtaining  $R^2$  values of 0.92 and 0.82 respectively between the real and de-  
85 tected number of berries. The work presented by Diago *et al.* [7] was tested  
86 with a dataset of ten images for each of seven varieties, with  $R^2$  values  
87 varying from 0.62 to 0.95 with an average of 0.82 across the seven cultivars.  
88 Aquino *et al.* [11] developed an algorithm to detect visible berries from a

89 single bunch photo in the field, achieving  $F1$  score = 0.89 based on their  
90 best parameter settings. As for actual berry estimation, that paper showed  
91 results of  $R^2 = 0.75$  and an accuracy of 84.35% between estimated berries  
92 and actual berries. This work was extended into an app [12] which was  
93 not available online at the time of writing. However, the image processing  
94 algorithms proposed by Grosssetete *et al.* [10] and Aquino *et al.* [11] rely  
95 on a specular reflection at a single point on each berry and are not robust  
96 following veraison since the surface of the berries may become matte and  
97 in some cases shrivelled.

98

99 Besides estimating the number of berries by processing single 2D im-  
100 ages, 3D bunch reconstruction has also been achieved using stereo imagery  
101 [13]. There, two 3D bunch models were built with substantial manual input  
102 and the method achieved  $R^2 = 0.797$  using a point model and  $R^2 = 0.778$   
103 against a CAD model. This customized stereo camera arrangement also has  
104 a natural minimum range and limits applicability to *ex-vivo*<sup>1</sup> analysis and  
105 maneuvering such a setup within a sprawling canopy is impractical.

106

107 Commercial mobile solutions have became the main objective which is  
108 challenging the robustness of existing image processing algorithms [11]. A  
109 common approach relies on a backing board with contrasting color to the  
110 bunch. As for actual berry counting, the most common approach is to esti-  
111 mate the occluded berries based on the detected number of visible berries  
112 [11]. This needs calibration and varies between cultivars and lighting con-

---

<sup>1</sup> *in-vivo* (also referred to as in-field) is used in this paper to refer to experimentation done without removing bunches from the shoot while *ex-vivo* means the condition that bunches are detached from the shoot

113 ditions since most existing algorithms are sensitive to illumination changes.  
114 Cultivar dependency of this calibration has not been investigated to date  
115 given the tedious data collection procedure.

116

117 In order to relieve the burden of building calibrations for each cultivar,  
118 Liu *et al.* [14] proposed a novel approach to count berries from a single  
119 image. Their method is limited to red grapes and can only deal with con-  
120 ical or cylindrically shaped bunches because the reconstruction procedure  
121 only follows the main branch of the bunch. A range of berry radii needs to  
122 be manual defined using their approach. The tested images were collected  
123 under laboratory conditions.

124

125 Hence with the assistance of a backing board and under the condition  
126 that the bunch can be segmented out from a high contrast backing board  
127 by mature bunch segmentation solutions [15, 16, 17] this paper focuses in  
128 particular on the robust berry counting solutions for both red and green  
129 bunches and its contribution to bunch weight estimation.

130

131 For direct comparison with other approaches, we provide the benchmark  
132 of collected bunch images and related metrics. The datasets are published<sup>2</sup>  
133 on the Smart Robotic Viticulture group’s website<sup>3</sup>.

134

135 In the remainder of this paper, Section 2 presents a field-robust algo-  
136 rithm for *in-field* berry counting based on a single RGB image, catering

---

<sup>2</sup><http://www.robotics.unsw.edu.au/srv/datasets.html>

<sup>3</sup><http://www.robotics.unsw.edu.au/srv>

137 for red and green bunches with a range of berry diameters. Section 3 de-  
138 scribes the experimental data and procedures used to validate the algorithm.  
139 Section 4 shows the accuracy of the results, as well as evaluating the robust-  
140 ness of the algorithm to differences in development stage, the contribution  
141 of berry counting to bunch weight estimation, and the possibility of yield  
142 estimation by image-based berry counting. Section 5 then draws conclu-  
143 sions and makes recommendations for future work.

144

## 145 2. Methodology

146 In general, berry counting is divided into three steps, Region of Interest  
147 (ROI) extraction, visible berry detection and actual berry count estimation,  
148 which are demonstrated in Figure 1. We propose a novel algorithm for 3D  
149 bunch reconstruction based on a single image for fast berry counting in vine-  
150 yards. According to the flowchart in Figure 1, the proposed approach starts  
151 with sub-bunch detection then processes each sub-bunch <sup>4</sup> before calculat-  
152 ing the bunch sparsity factor (mentioned in Section 2.4) and reconstructing  
153 the 3D bunch model (explained in Section 2.3).

154

155 The proposed berry counting procedure is divided into two parallel steps  
156 once the sub-bunches are defined by color segmentation: on one hand, ob-  
157 taining the initial 3D bunch model to get a berry number (see Figure 2);  
158 on the other hand, calculating the sparsity factor based on the difference of  
159 two color channels. The final berry number is obtained by combining this

---

<sup>4</sup>A sub-bunch refers to the separate sections of the bunches which are visually disconnected in the image other than by rachis structure

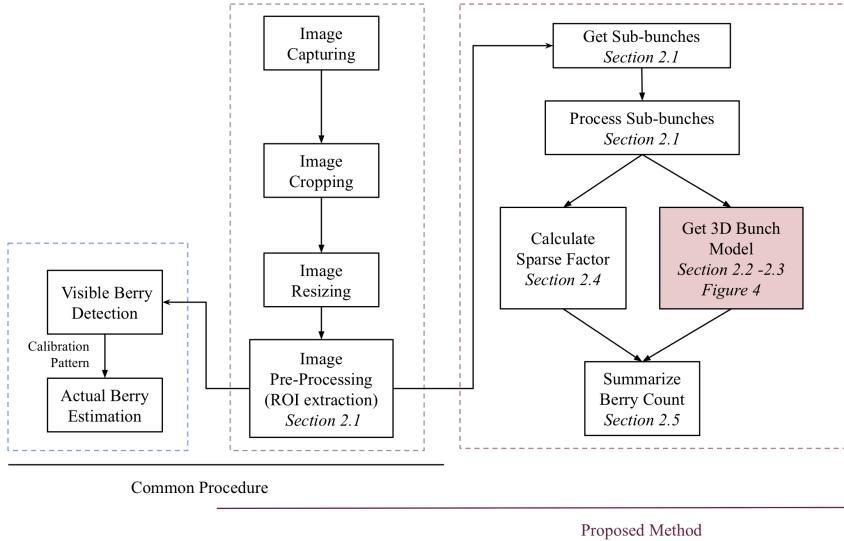


Figure 1: The proposed fast berry counting approach by 3D bunch reconstruction

160 initial berry number and the sparsity factor. Figure 2 gives an overview  
 161 and Section 2.3 and Section 2.4 give detailed explanations of these steps.  
 162 For convenience, the list below provides the explanation of major variables  
 163 referred to in this paper:

164

165 **Nomenclature**

166  $\sigma$  w/l, Bunch width / Bunch length

167  $a_r$  Sum of areas in pixels calculated by Red channel from RGB color

168  $a_s$  Sum of areas in pixels calculated by Saturation channel from HSV  
169 color space

170  $a_v$  Sum of areas in pixels calculated by Value channel from HSV color  
171 space

172  $b_e; b_{g_i}$  Estimated berries at the bunch edge; Estimated berries in each group

173  $B_x$  Estimated Sub-bunch

174  $bw_e$  Binary image of  $B_x$ 's edge

175  $C$  Histogram bin counts

176  $E$  The value of bin edge

177  $e_p, e_a$  The percentage error and absolute error, see Equations 3 and 4

178  $f$  Sparsity factor of a bunch or sub-bunch

179  $g_i$  Groups of berries at the bunch edge

180  $I_g$  Grayscale image of  $B_x$

181  $id$  Index

182  $k$  Curvature for each pixel

183  $L$  The set of edge line segments

<sup>184</sup>  $n_e, n_a$  Estimated and actual berry number per bunch/sub-bunch

<sup>185</sup>  $n_i$  Initial berry number per bunch or sub-bunch

<sup>186</sup>  $P$  Points extract from  $L$

<sup>187</sup>  $Po$  Fitted polygon

<sup>188</sup>  $R; R^2$  Radius of berries; R-square value, the goodness of fit

<sup>189</sup>  $r, c$  Row and column numbers of an image

<sup>190</sup>  $s$  Step size for 3D position adjusting of a berry along a track

<sup>191</sup>  $t$  Tolerance of berry overlapping

<sup>192</sup>  $tmp$  Temporary count

<sup>193</sup>  $v$  The metric value of detected berries by Hough Transform

<sup>194</sup>  $w_b$  Weight of berry

<sup>195</sup>  $w_{B_e}, w_{B_a}$  Estimated bunch weight and actual bunch weight

<sup>196</sup>  $x, y, z$  Coordinates of a berry

197    *2.1. Sub-bunch Segmentation*

198    A backing board is used during image collection to aid segmentation and  
199    is recommended to be of contrasting color with the berries. In this work, a  
200    black backing board was utilized for green bunches while a white backing  
201    board was used for red bunches. The backing board provides good contrast  
202    which leads to better bunch identification.

203

204    Alternatively, bunch detection (Figure 2b) can be done by the methods  
205    described by Luo *et al.* [15, 16] or Perez-Zavala *et al.* [17] amongst many  
206    existing approaches. In this paper ROI extraction (Figure 2c) is conducted  
207    by Otsu's method [18], which is commonly adopted for bunch segmentation  
208    [14, 11]. The non-connected ROIs are treated as sub-bunches and labelled  
209    as  $B_x$ , and for each  $n_i$  and  $f$  are calculated individually.

210

211    *2.2. Robust Berry Detection*

212    A vital step in berry detection is that of initializing the range of ra-  
213    dius of berries for applying the Hough Transform. Approaches presented by  
214    Mirbod *et al.* [19] and Dahal *et al.* [20] could be applied on bunches with  
215    a high-reflection spot (from artificial lighting or under lab conditions), but  
216    this was not possible for the field-based datasets in this paper. For a more  
217    practical usage, we applied Sobel edge detection followed by several basic  
218    morphological operations [21], which includes removing small objects from  
219    binary image (in this case those less than 100 pixels in area) and connection  
220    checking between segmented lines (in this case those that have less than a  
221    5 pixel gap), to extract the edges of visible berries (Figure 3b).

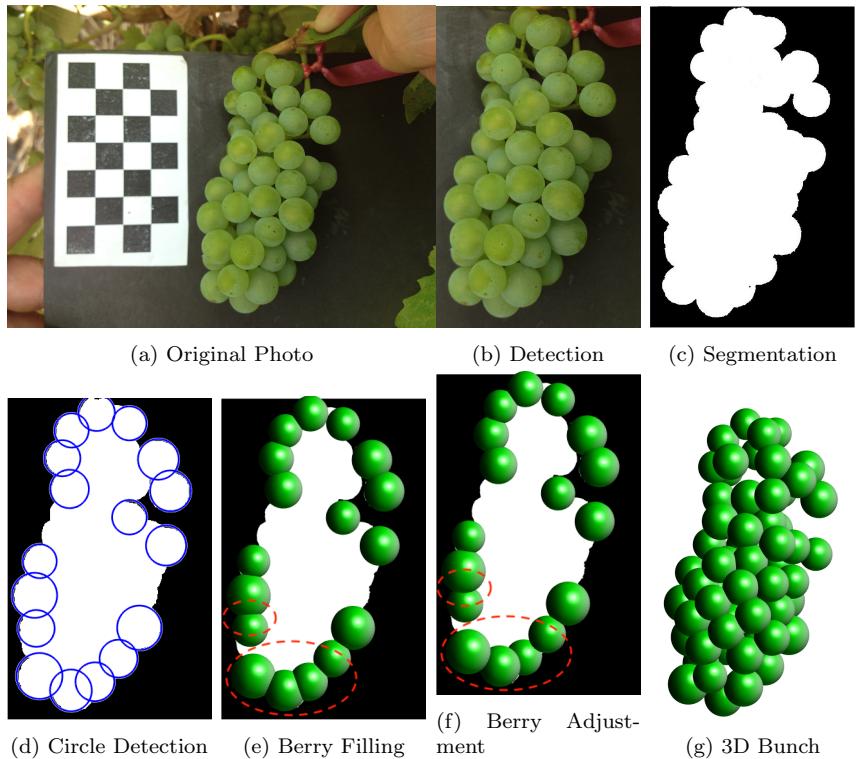


Figure 2: Steps in the detailed bunch reconstruction process, illustrated with an *in-vivo* image of a Chardonnay bunch. The original photo (a) has been cropped (b) and segmented from the background (c). From this segmentation outline, berries on the exterior are fitted (d) where the outline has appropriate curvature. Where these berries overlap (e), they are adjusted forward or backwards until there are no collisions (f). Then the remaining berries are placed inside the hull formed from the segmented outline as per Figure 4.

222

223    We then propose a new algorithm (Algorithm 1) to estimate the ini-  
224    tial range of berry radii for edge berry and internal berry detection. Part  
225    curvature calculation is demonstrated in Figure 3c; showing that a large  
226    proportion of the edge points ( $k$ ) have similar radii. This guides the choice  
227    of the initial berry radius range as described in Algorithm 1.

228

---

**Algorithm 1:** Estimation of initial berry radius range

---

**Data:**  $I_g$  of  $B_x$

**Result:** Radius range for berry detection,  $R$

```
1  $bw_e \leftarrow$  edge detection by Sobel [21] ;                    // see Figure 3b
2  $L_i \leftarrow bw_e$  ;                                                    // The set of edge line segments
3  $tmp \leftarrow 0$ ;
4 for  $i \leftarrow 1$  to findNum( $L_i$ ) do
5      $P_j \leftarrow$  discrete( $L_i$ );
6     for  $j \leftarrow 2$  to findNum( $P_j$ ) - 1 do
7        $tmp \leftarrow tmp + 1$ ;
8        $Po_j \leftarrow fit(P_{j-1}, P_j, P_{j+1})$ ;
9        $k_{tmp} \leftarrow cur(Po_j)$  ;                                    // Curvature from the fitted
                                                                          polygon
10      end
11 end
12  $k \leftarrow k(\text{abs}(k) > 0 \text{ AND } \text{abs}(k) < Inf)$ ;            // Filter curvature
13  $R \leftarrow 1./k$ ;                                                    // Calculate radii for each edge pixel
14  $[C, E] \leftarrow histcounts(R)$ ;                                    // Histogram of radii
15  $id \leftarrow \text{FindMax}(C)$ ;                                            // Find the bin with maximum counts
16  $R \leftarrow \text{Round}([E(id - 1), E(id + 1)])$ ;                    // Specify a range of
                                                                                  radii from bin widths
```

---

229    2.3. Sub-bunch Reconstruction

230    Figure 4 shows the process of how each sub-bunch,  $B_x$ , is reconstructed  
231    to give an initial berry number.

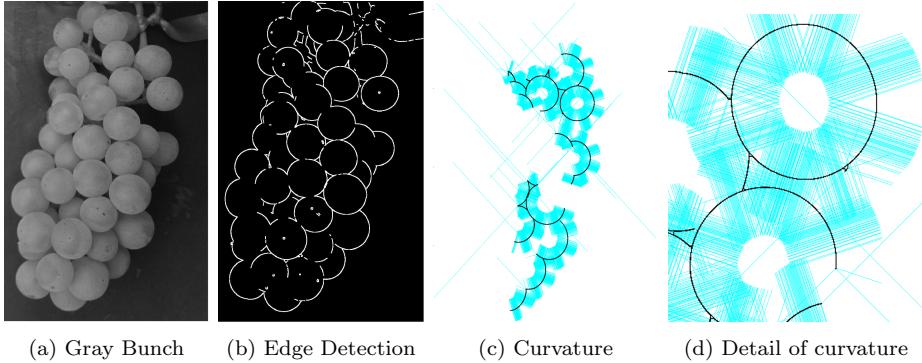


Figure 3: Estimation of initial berry radius range. Cyan lines indicate the magnitude and direction of the normal of each edge pixel.

232

233       The berries at the edges are detected by the Hough transform [21](Figure  
 234      2d), aided by the algorithm in Section 2.2. Those berries are defined as  $b_e$   
 235      and the diameter and location of each one are used to place correspond-  
 236      ing spheres in a plane parallel to the image sensor as shown in Figure 2e.  
 237      Neighbourhood searching [22] using a threshold of the largest estimated  
 238      berry radius is applied for grouping the overlapping berries in our proposed  
 239      edge berry adjusting algorithm 2. For each group, berries are sorted based  
 240      on the likelihood of being a circle, as calculated by the Hough transform.  
 241      The berry with largest likelihood is moved closer the camera (along the z-  
 242      axis) while the berry with smallest metric is moved further from the camera  
 243      until there is no overlap (Figure 2f). The overlap has an additional toler-  
 244      ance,  $t$ , added as a proportion of the berry radius.

245

246       Once the edge berries have been positioned, the  $B_x$  is divided into two  
 247      parts if the ratio between the width and length of a bunch is larger than  
 248      3 : 4. A watershed point of the divided bunch is formed 3/4 of the way

---

**Algorithm 2:** Berry adjustment at bunch edges

---

**input** : A binary image  $bw_e$  of bunch with size  $r \times c$ , step size  $s$ , tolerance  $t$

**output:** 3D location  $(X_e, Y_e, Z_e)$  of detected berries at a bunch edge

```
1  $bw_e \leftarrow$  bunch segmentation;
2  $b_e[x, y, R, v] \leftarrow$  hough( $bw_e$ );
3  $g_i \leftarrow b_e$ ; // Nearest Neighbor Search[22]
4 for  $i \leftarrow 1$  to findNum( $g_i$ ) do
5    $b_{g_i} \leftarrow$  sort( $g_i(v)$ );
6    $v_m \leftarrow$  findMedian( $g_i(v)$ );
7    $b_{g_{i_m}} \leftarrow$  Find( $g_i(v) = v_m$ );
8   for  $j \leftarrow 1$  to findNum( $b_{g_i}$ ) do
9     down  $\leftarrow$  moving berry backward along the z-axis by  $s$ ;
10    up  $\leftarrow$  moving berry forward along the z-axis by  $s$ ;
11    stay  $\leftarrow$  don't move;
12    keep  $\leftarrow$  save current  $b_{g_{i_j}}(x, y, z)$  in a stack  $(X_e, Y_e, Z_e)$ ;
13    if  $b_{g_{i_j}}(v) < b_{g_{i_m}}(v)$  then
14      while  $b_{g_{i_j}}(x, y, z)$  collides with  $(X_e, Y_e, Z_e)$  by  $t$  do
15        down ( $b_{g_{i_j}}(x, y, z)$ )
16        keep  $b_{g_{i_j}}(x, y, z)$ 
17    else if  $b_{g_{i_j}}(v) > b_{g_{i_m}}(v)$  then
18      while  $b_{g_{i_j}}(x, y, z)$  collides with  $(X_e, Y_e, Z_e)$  by  $t$  do
19        up ( $b_{g_{i_j}}(x, y, z)$ )
20        keep  $b_{g_{i_j}}(x, y, z)$ 
21    else
22      stay ( $b_{g_{i_j}}(x, y, z)$ )
```

---

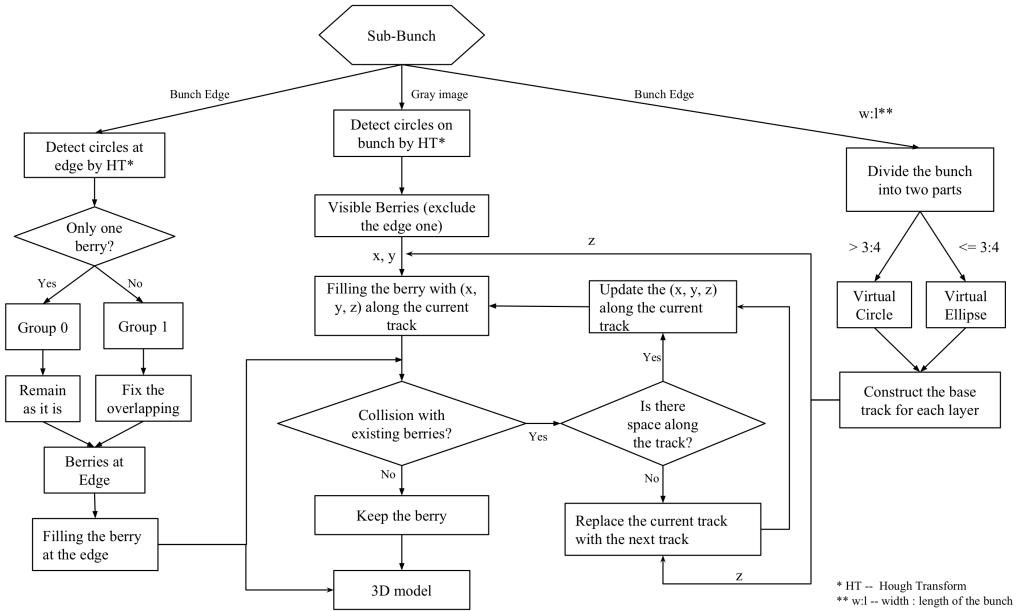


Figure 4: Flowchart of the proposed 3D bunch reconstruction algorithm based on a single sub-bunch

249 down the length of the longitudinal axis. This value has been defined em-  
 250 pirically, based on the available data. Then, virtual tracks are formed to  
 251 place berries by finding the first and last pixel of a section through the width  
 252 of the  $B_x$ . This section is revolved about the longitudinal axis through its  
 253 middle, forming the virtual track. If the  $B_x$  has been divided into two parts,  
 254 the upper part is revolved as an ellipse whereas the lower part is revolved  
 255 as a circle. This provides the  $z$  value for each berry candidate. The video  
 256 attached to this article simulates this process.

257

258 The next step is detection of all visible berries from the gray image of the  
 259  $B_x$ , again using the Hough transform. Berries at the edges are subtracted  
 260 and the remaining berries are placed where they are detected. Their  $z$  po-  
 261 sition is provided by virtual track generated in the proposed algorithm to

262 form the three dimensional position for each berry.

263

264 The next step is to fill non-visible or interior berries in a shell of tracks  
265 that is generated based on image processing. The radius for each populated  
266 berry is chosen according to the distribution of radii detected from all vis-  
267 ible berries and assumes the berries are all spherical. Starting at the top  
268 detected berry and moving around the circumference of each virtual track,  
269 placement of a new berry is attempted at regular intervals (1 degree is used  
270 in this work) and is considered successful if no collision with any existing  
271 sphere is detected.

272

273 The berry placement moves down the bunch by a step size defined as  $s$   
274 (chosen to be two pixels in this work) and repeats the placement attempts  
275 on the track below until the bottom of the bunch is reached and the model  
276 is complete. The number of berries placed is tallied to form the Initial Berry  
277 Number ( $n_i$ ).

278

#### 279 *2.4. Sparsity Factor Calculation*

280 The initial 3D model is built based on the assumption that the processed  
281 bunch is healthy and compact. When berries fill the convex hull estimated  
282 from a single image there are no concerns about the ‘empty space’. This  
283 means that berries are tightly packed in the initial 3D bunch model and  
284 does not allow for any of the rachis structure, resulting in an overestimate  
285 of berry number. An extended sparsity factor ( $f$ ) is proposed to process  
286 both red and green grapes according to the visible proportion of berries

287 within the bunch area — a proxy for bunch compactness:

$$f = \begin{cases} (a_r - a_s)/a_r & \text{red} \\ (a_s - a_v)/a_s & \text{green} \end{cases} \quad (1)$$

288

289

### 290 2.5. Berry Count Estimation

The estimation of the number of berries is improved through the application of the extended sparsity factor according to the following formula:

$$n_e = (1 - f) \times n_i \quad (2)$$

291 where  $n_e$  is the final estimate of the berry number per sub-bunch. This is  
292 then tallied across the sub-bunches to give the final number of berries in  
293 the entire bunch. Both of the equations above were determined empirically,  
294 and found to be appropriate for all the datasets tested.

295

## 296 3. Data Scope and Experimental Design

297 In total, 529 bunch images from two cultivars were tested and the de-  
298 tails of each dataset are illustrated in Table 1 including whether *in-vivo* or  
299 *ex-vivo*<sup>1</sup> and which model of smartphone was used. All bunches were pho-  
300 tographed in the field (whether *in-vivo* or *ex-vivo*) without artificial lighting,  
301 replicating end-user usage, the only requirement being the use of a back-  
302 ing board to aid segmentation. The proposed 3D reconstruction algorithm

303 and sparsity factor calculation was implemented in Matlab (R2018b, Math-  
304 works, MA, USA) and a PC with Intel Core i5-6500 and 16 GB RAM was  
305 used to process all the images to obtain the final estimates of the number  
306 of berries ( $n_e$ ).

307 The proposed method was validated in two aspects regarding berry num-  
308 ber and one aspect regarding bunch weight as well as in application to yield  
309 estimation:

- 310     • A1, the accuracy of the berry counting algorithm for bunches of dif-  
311         ferent colours
- 312     • A2, the robustness of the algorithm to different development stages of  
313         bunches
- 314     • A3, the accuracy of bunch weight estimation, derived from fast berry  
315         counting
- 316     • A4, the accuracy of yield estimation from bunch weight estimates

Table 1: Details of datasets D1 to D8 used in this paper

Dataset	Num. of images	<i>in/ex-vivo</i> <sup>1</sup>	Cultivar	Color	Development (E-L) Stage [23]	Location	Date Imaged	Smart-phone	Resolution [pixels]	Aims
D1	94	<i>ex-vivo</i>	Chardonnay	green / light yellow	38 Harvest	Clare Valley, SA, Australia	28/02/2017	LG G3	4160*2340	A1, A3, A4
D2	73		Shiraz	red / light yellow			28/03/2017			
D3	86		Chardonnay	green / light yellow		Orange, NSW Australia	02/03/2017			
D4	44	<i>in-vivo</i>	Shiraz	red	31 Berries pea-sized 33 Berries still hard and green 34 Berries begin to soften	Clare Valley, SA, Australia	10/02/2015	iPhone 5	2448*3264	A1, A2
D5	56			green		Orange, NSW Australia	23/02/2015		1536*2048	
D6	63		Chardonnay	green		Clare Valley, SA, Australia	30/12/2014	iPhone 4S	2448*3264	
D7	56			green			29/12/2014			
D8	57			green			05/01/2015			

317        Each photographed bunch was deconstructed and the number of berries  
318        counted manually with shrivelled or substantially smaller berries being ex-  
319        cluded from the count.

320        For comparison of the bunch weight, this was calculated using the estimated  
321        berry number and average berry weight:  $w_{B_e} = n_e * w_b$  and then compared  
322        with the measured bunch weight. In this dataset, the average berry weight  
323        was calculated by weighing five berries individually from each bunch; in  
324        practice a number of alternate methods are available to determine average  
325        berry weight.

326        With reference to the nomenclature section above, the evaluation indicators  
327        used in this paper are:

- 328        • The  $R^2$  value, based on a linear correlation between the actual and  
329        estimated number of berries, which can be used to reflect the goodness  
330        of fit between these two groups of numbers.
- 328        • The percentage error  $e_p(\%)$ , which is defined as:

$$e_p(\%) = \begin{cases} (n_e - n_a)/n_a * 100 & \text{berry counting} \\ (w_{B_e} - w_{B_a})/w_{B_a} * 100 & \text{bunch weight estimation} \end{cases} \quad (3)$$

331        In many cases, the user would desire an average berry count per bunch  
332        over a number of samples in the block, so this metric is used as an  
333        estimate of the bias of the average value and thus the robustness of  
334        the approach in practice.

- The percentage of the absolute error  $e_a(\%)$ , which is defined as:

$$e_a(\%) = \begin{cases} |n_e - n_a|/n_a * 100 & \text{berry counting} \\ |w_{B_e} - w_{B_a}|/w_{B_a} * 100 & \text{bunch weight estimation} \end{cases} \quad (4)$$

335        This measure is best suited for determining the accuracy of the method  
 336        for a single image as errors are not balanced out when averaged over  
 337        a large number of bunches.

338        **4. Experimental Results and Discussion**

339        Using the computer described above, each image took 0.1 seconds to  
 340        be processed, without any code optimisation. Qualitatively, manual obser-  
 341        vations of the real and reconstructed bunches matched quite well, as the  
 342        proposed method fits berries around the outer profile of the bunch, similar  
 343        to common bunch architectures. Figure 5 demonstrates some 3D bunch  
 344        models reconstructed by the proposed method. Note the variation in the  
 345        illumination conditions and bunch structure that exists among the origi-  
 346        nal photos and thereby the robustness of this method to handling both  
 347        symmetric and non-symmetric bunches. It should be emphasised that the  
 348        reconstruction is an estimate and the berries may appear smaller due to  
 349        the shading used. The results clearly show (Figure 6) the number of berries  
 350        calculated is reasonably accurate. The exact positioning of berries not de-  
 351        tected from one view is only an estimate.

352

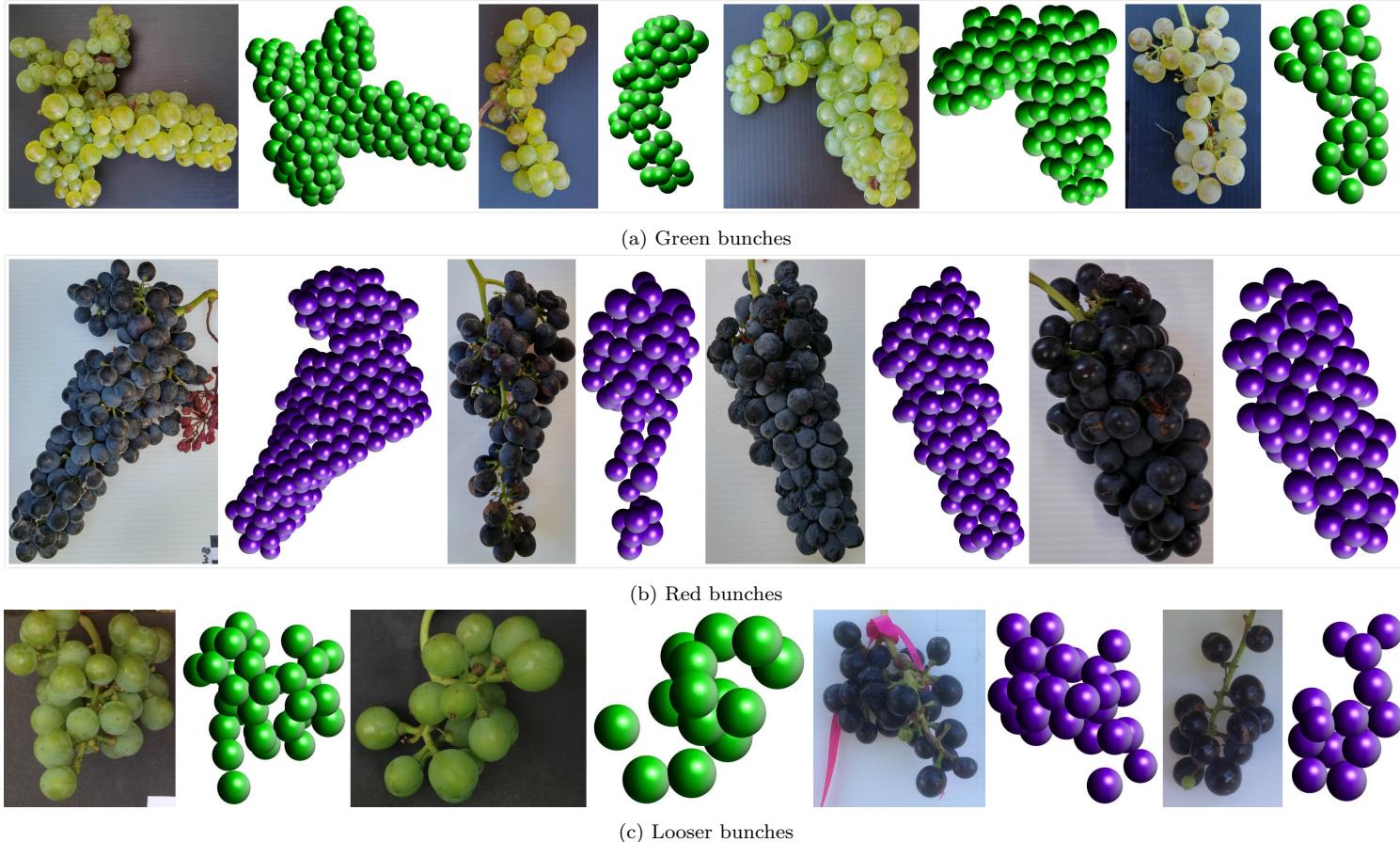


Figure 5: Reconstructed 3D models of bunches for both green and red bunches. Images were collected in the field by smartphones and the variation in lighting conditions and bunch architecture can be seen.

353    4.1. Accuracy of Berry Counting (A1)

354    Figure 6 represents the quantitative relationship between estimated and  
355    actual berry numbers found from the processed images of both red and  
356    green cultivars. Eight datasets (Table 1) were tested and the detailed re-  
357    sults are illustrated in Table 2.  $R^2$  values varied from 0.82 to 0.95 (average  
358    0.91), along with the percentage of absolute error  $10.2\% \sim 15.26\%$  (aver-  
359    age 11.85%) and the percentage of error  $-6.0\% \sim 8.16\%$  (average 3.1%),  
360    indicating the good performance of the proposed fast berry counting al-  
361    gorithm. A slight decrease in the performance of the algorithm was seen  
362    in-vivo (datasets D4-D5) as opposed to ex-vivo (datasets D1-D3), most  
363    likely due to more variable illumination and a smaller datasets more prone  
364    to outliers.

365

366    Table 2 compares the proposed method with five state of the art methods  
367    presented in the literature. The methods proposed by Diago *et al.* [7] and  
368    Aquino *et al.* [11] require calibration of the relationship between visible and  
369    non-visible berries, which varies between cultivars and development stage.  
370    The approaches presented by Herrero-Huerta *et al.* [13], Schöler *et al.* [3]  
371    and Rist *et al.* [5] need human interaction with software or at least the  
372    tuning of parameters. Alternatively, the method presented in this paper is  
373    not-cultivar or development stage specific and requires no human interac-  
374    tion once the image has been acquired. Furthermore, the proposed method  
375    works accurately with both red (Shiraz) and green (Chardonnay and pre-  
376    veraison Shiraz) cultivars and is able to achieve  $R^2$  values on par with the  
377    existing work without the drawbacks mentioned above. In addition, direct  
378    comparison of the accuracy ( $e_p(\%)$  and  $e_a(\%)$ ) shows that the accuracy is

379 at least as high as the existing work as shown in Table 2. Results pre-  
380 sented in the original papers are directly provided for comparison, as the  
381 corresponding datasets have not been made available<sup>5</sup>.

---

<sup>5</sup>Our data: <http://www.robotics.unsw.edu.au/srv/datasets.html>

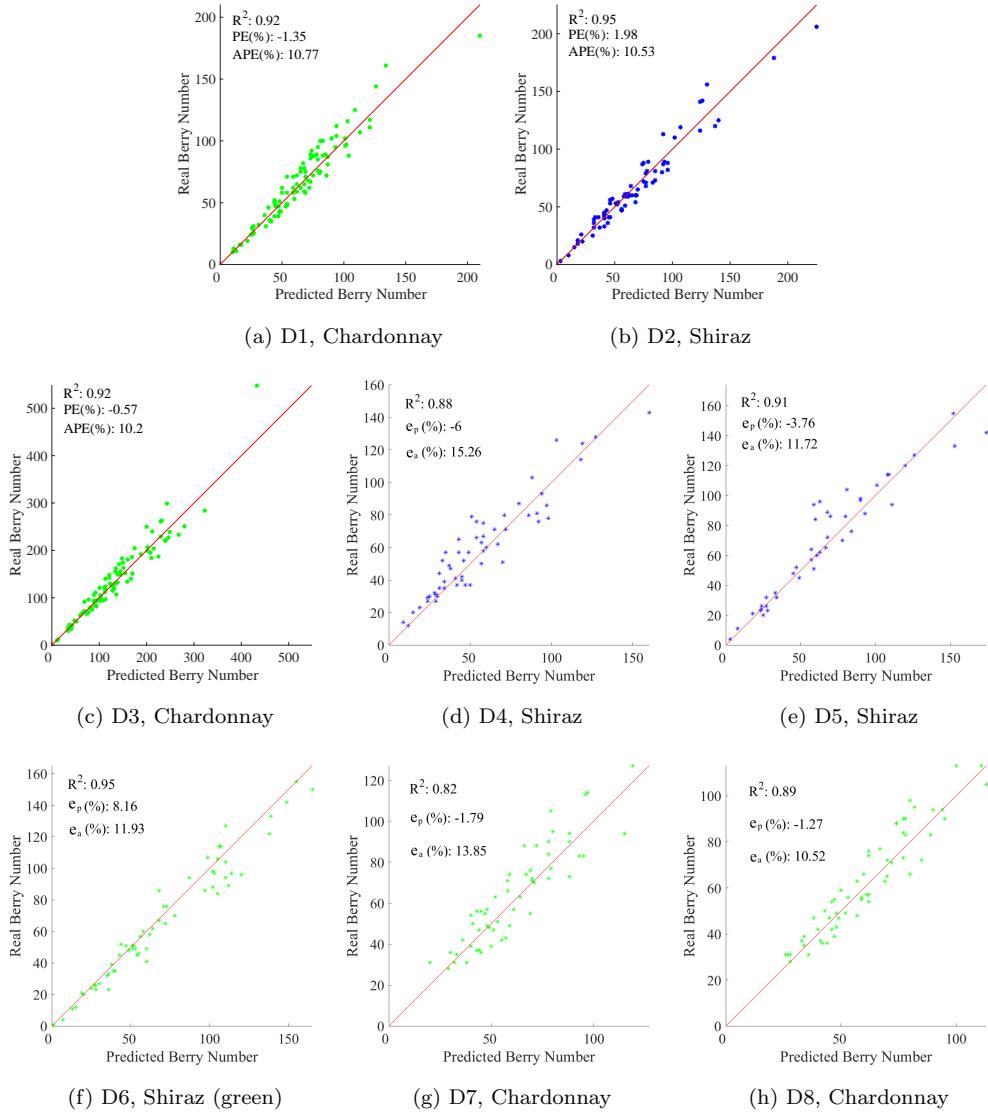


Figure 6: The accuracy of actual berry counting both for green and red grapes (A1). The red line indicates a 1:1 relationship.

Table 2: Comparing existing algorithms with the presented algorithm for berry counting

<b>Method</b>	<b>Cultivar</b>	<b>Color</b>	<b>Number of Bunches</b>	<b>Process Time per Bunch</b>	$R^2$	$e_a(\%)$	$e_p(\%)$
Schóler <i>et al.</i> [3]	Riesling	Green	4	> 10m	NA	12.35	3.30
Rist <i>et al.</i> [5]	Calardis Blanc, Dornfelder, Pinot Noir and Riesling	Red and Green	12 per culti-var	$\approx 1m$	0.94	NA	NA
	Calardis Blanc	Green	222		0.95		-21.4
Diago <i>et al.</i> [7]	7 common European cultivars	Red	10 per culti-var	$\approx 2s$	0.82	NA	NA
		Green	11 per culti-var		$\approx 0.0407s$	0.75	15.65
Herrero-Huerta <i>et al.</i> [13]	Tempranillo	Red	20	slow	0.78	NA	
Proposed Method	Chardonnay, D1	Green / Light Yellow	94	$\approx 0.1s$	0.92	10.77	-1.35
	Shiraz, D2	Red	73		0.95	10.53	1.98
	Chardonnay, D3	Green / Light Yellow	86		0.92	10.20	-0.57
	Shiraz, D4	Red	44		0.88	15.26	-6.00
	Shiraz, D5		56		0.91	11.72	-3.76
	Shiraz, D6	Green	63		0.95	11.93	8.16
	Chardonnay, D7		56		0.82	13.85	-1.79
	Chardonnay, D8		57		0.89	10.52	-1.27

382 In addition, the code was converted into an iOS app (3DBunch) tested  
383 on an iPad mini 4 with an Apple A8 processor, PowerVR GX6450 GPU, 2G  
384 LPDDR3 RAM using iOS version 12.3.1. 100 photos were tested on that  
385 device and the average processing time for each image was approximately 1  
386 second (excluding human interaction). The average absolute error of berry  
387 counting for those 100 images is around 8% and the average percentage  
388 error is 2.6%. Discrepancies between the desktop and mobile versions of  
389 these results are only due to limited image processing library functions on  
390 mobile operating systems. Detailed experimental results from the mobile  
391 platform are expected to be provided in future work.

392 *4.2. Robustness to Development Stage (A2)*

393 Since the compactness or sparsity of the bunches varies as they grow,  
394 the accuracy of the 3D reconstruction method was examined in the con-  
395 text of these different growth stages and the results are shown in Table 3.  
396 Datasets D4 to D8 were of images captured *in-field* and their development  
397 stage varied from E-L stage 31 (Pea-size stage) to E-L stage 38 (Harvest)  
398 following the Modified E-L naming convention [23]. The absolute average  
399 error from lag-stage to harvest stage for the green bunches was in the range  
400 of 10.20% — 15.26% with an average of 12.66%, very similar to the harvest  
401 stage results. Hence, this method is robust to different development stages,  
402 back as far as pea-sized bunches.

403

404 The ability of the proposed method to accurately estimate the berry  
405 number in the early stages of development allows the user to rapidly esti-  
406 mate current bunch weights non-destructively.

Table 3: Performance of the algorithm for bunches at different development stages (A2)

<b>Dataset</b>	<b>Cultivar</b>	<b>Color</b>	<b>E-L Stage</b>	$R^2$	$e_a(\%)$	$e_p(\%)$	
D4	Shiraz	Red	38 Harvest	0.88	15.26	-6.00	
D5				0.91	11.72	-3.76	
D6		Green	31 Berries pea-size	0.95	11.93	8.16	
D7	Chardonnay		33 Berries still hard and green	0.82	13.85	-1.79	
D8			34 Berries begin to soften	0.89	10.52	-1.27	

407

408 *4.3. Accuracy of Bunch Weight Estimation (A3)*

409     Figure 7 and Table 4 show the results of the algorithm being applied  
 410     to estimate bunch weights. The results demonstrate high correlation ( $R^2$   
 411     ranges from 0.83 — 0.92) between estimated bunch weight and measured  
 412     bunch weight. For an individual bunch, the errors were larger than by  
 413     direct comparison with berry number, but this is to be expected given the  
 414     uncertainty in berry weight. Averaged over several dozen bunches, the error  
 415     reduced to less than 8%, suggesting this is a feasible method for farmers to  
 416     rapidly obtain a measure of average bunch weight non-destructively.

417 *4.4. Yield Estimates Utilising Berry Counts (A4)*

418     We then applied the algorithm along with a shoot counting method [24]  
 419     to assist grape yield estimation in 2017 at two different vineyards, using the  
 420     yield estimation method and data specified in Whitty et al. [1]. In summary,  
 421     shoots were counted from a mobile camera across the whole block, then in

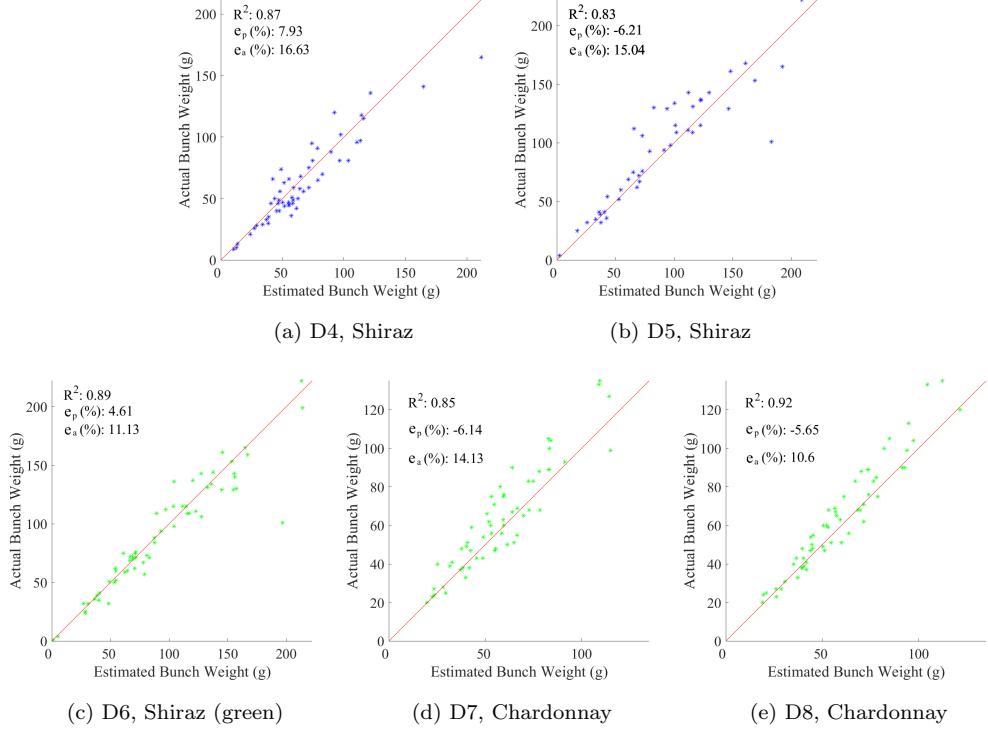


Figure 7: A3, the accuracy of bunch weight estimation based on the proposed berry counting algorithm. The red lines indicate 1:1 relationships.

Table 4: Performance of the reconstruction algorithm when comparing estimated and actual bunch weights

Dataset	Cultivar	Color	E-L Stage	$R^2$	$e_a(\%)$	$e_p(\%)$
D4	Shiraz	Red	38 Harvest	0.87	16.63	7.93
D5				0.83	15.04	-6.21
D6	Chardonnay	Green	31 Berries pea-size	0.89	11.13	4.61
D7			33 Berries still hard and green	0.85	14.13	-6.14
D8			34 Berries begin to soften	0.92	10.60	-5.65

Table 5: Yield estimation assisted by the proposed berry counting method in 2017 for three blocks. Two methods of determining berry weight at harvest are compared, along with a standard industry approach for yield estimation.

Block		40A	47A	B12
<b>Actual yield (t)</b>		33.50	106.09	45.27
<b>Estimated yield (t)</b>	Using historical berry weight	34.12	77.99	50.43
	Measured berry weight	35.56	88.99	46.77
<b>Manual yield Estimate (t)</b>		30.04	80.17	43.97
<b>Yield estimation error (%)</b>	Using historical berry weight	1.84	-26.49	11.39
	Measured berry weight	6.16	-16.11	3.31
<b>Manual yield estimate error (%)</b>		-10.34	-24.43	-2.88

422 30 sample locations the number of bunches per shoot was measured and the  
 423 bunches were photographed using a smartphone. The average berry weight  
 424 at harvest was calculated by two different methods, either using historical  
 425 averages for each block or by direct measurements of berry weight. The  
 426 combination of shoot counts, bunches per shoot, berries per bunch (from  
 427 the method in this paper) and harvest berry weight, along with correction  
 428 factors such as harvester efficiency were used to predict the yield in each  
 429 block, as shown in Table 5. Manual yield estimation results using industry  
 430 standard approaches are also provided for comparison, see Whitty et al. [1]  
 431 for further details.

432

433 The data demonstrated in Table 5 presents an encouraging application

434 for the proposed berry counting algorithm. Given automated image-based  
435 shoot counts from over 65km of vine rows in three blocks, combined with  
436 253 smartphone images of bunches and average berry weights from a subset  
437 of bunches, the yield close to harvest was estimated within 6%, 16% and  
438 3% of the actual yield respectively. This was generally better than industry  
439 standard manual yield estimates of the same blocks, despite shoot counts  
440 being taken five months prior to harvest and a substantially reduced require-  
441 ment for manual labour. The result for block 47A was noticeably poorer  
442 due to an unexpectedly large drop in shoot number between the shoot and  
443 harvest stage.

444 When the historical berry weights were used, the error increased sub-  
445 stantially, as there was a notable variation in average berry weight in this  
446 season for the blocks tested. Hence a reliable measure of berry weight is  
447 required to improve the yield estimation.

## 448 5. Conclusions

449 This paper has presented a novel and fast algorithm which is able to  
450 count berries and estimate the 3D structure of both red and green grapes  
451 *in-field* from pea-sized to harvest development stages from a range of bunch  
452 architectures. Using only a single image from a smartphone and no cali-  
453 bration or prior information, the accuracy of the method was 89% when  
454 directly compared to the number of berries on a bunch. When averaged  
455 across 50-80 images, the accuracy was over 99%, showing the limited bias  
456 present in this uncalibrated approach.

457 The algorithm was found to be robust to different bunch architectures  
458 qualitatively as well as give consistent results from pea-sized to harvest

459 development stages. The rapid processing time of 0.1 seconds per image  
460 is dramatically faster than manual counting and faster than existing ap-  
461 proaches in the literature as well as requiring no human interaction once  
462 the image has been captured. When implemented on a mobile operating  
463 system, the processing time was within one second per image, allowing it  
464 to be used in the field.

465 When the proposed method was used to estimate bunch weights, an  
466 accuracy of more than 92% was found on average over several dozen bunches  
467 in each dataset. Furthermore, the algorithm was applied to yield estimation  
468 and found to have an error of between 3% and 16% when compared directly  
469 to measured tonnes at harvest, using automated shoot counting [24] and a  
470 berries per shoot method [1]. Hence, the algorithm has applicability in field  
471 scenarios and the potential to speed up and improve the accuracy of yield  
472 estimates for farmers using smartphones.

473 Given the location of each image from a smartphone, this could be di-  
474 rectly applied to map bunch weight and yield variation across a block. In  
475 addition, by combining with existing work on flower counting [25, 26, 27],  
476 the possibility of efficient determination of fruit set ratios on a large scale  
477 is envisaged.

478 Some varieties of grape berries elongate noticeably following véraison,  
479 and this method could be extended to fitting ellipses to each berry [28] and  
480 reconstruction using corresponding ellipsoids. Improvements could be made  
481 to the determination of the extended sparsity factor however this has been  
482 left to future work and to aid the generalisability of the algorithm. The re-  
483 construction also provides opportunities for estimating more detailed bunch  
484 parameters, which are left for future work. Converting this algorithm into

485 an app for farmers would allow rapid and non-destructive estimates of berry  
486 counts and bunch weights with limited bias. Further research is currently  
487 conducted on the bunch segmentation step with the aim of removing the  
488 need for the high-contrast back board. This would make it easier for farm-  
489 ers to adopt the method in the field.

490

## 491 Acknowledgements

492 The datasets used in this paper were obtained with the support of  
493 Wine Australia project DPI1401 Improved Yield Estimation for the Aus-  
494 tralian Wine Industry. All these datasets are freely available for down-  
495 load and comparison from the Smart Robotic Viticulture group's website:  
496 <http://www.robotics.unsw.edu.au/srv/datasets.html>.

497

## 498 References

- 499 [1] M. Whitty, S. Liu, S. Cossell, H. Jayakody, M. Woods, J. Tang, S. Singh, *etc.*, Im-  
500 proved yield estimation for the australian wine industry, Tech. rep., Wine Australia  
501 (11 2017).
- 502 [2] S. Nuske, K. Gupta, S. Narasimhan, S. Singh, Modeling and calibrating visual yield  
503 estimates in vineyards, in: Proceedings of Field and Service Robotics, Springer,  
504 2014, pp. 343–356.
- 505 [3] F. Schöler, V. Steinhage, Automated 3d reconstruction of grape cluster architecture  
506 from sensor data for efficient phenotyping, Computers and Electronics in Agriculture  
507 114 (2015) 163–177. doi:10.1016/j.compag.2015.04.001.
- 508 [4] S. Martin, R. Dunstone, G. Dunn, How to forecast wine grape deliveries using grape  
509 forecaster excel workbook version 7, GWRDC: Adelaide, Australia (2003) 100.

- 510 [5] F. Rist, K. Herzog, J. Mack, R. Richter, V. Steinhage, R. Töpfer, High-precision  
511 phenotyping of grape bunch architecture using fast 3d sensor and automation, Sensors 18 (3) (2018) 763. doi:10.3390/s18030763.
- 512
- 513 [6] S. Liu, S. Marden, M. Whitty, Towards automated yield estimation in viticulture, in:  
514 Proceedings of the Australasian Conference on Robotics and Automation, Sydney,  
515 Australia, Vol. 24, 2013.
- 516 [7] M. P. Diago, J. Tardaguila, N. Aleixos, B. Millan, J. M. Prats-Montalban, S. Cubero,  
517 J. Blasco, Assessment of cluster yield components by image analysis, Journal of the  
518 Science of Food and Agriculture 95 (6) (2015) 1274–1282. doi:10.1002/jsfa.6819.
- 519 [8] E. Ivorra, A. Sánchez, J. Camarasa, M. Diago, J. Tardaguila, Assessment of grape  
520 cluster yield components based on 3D descriptors using stereo vision, Food Control  
521 50 (2015) 273–282. doi:10.1016/j.foodcont.2014.09.004.
- 522 [9] A. Kicherer, R. Roscher, K. Herzog, S. Šimon, W. Förstner, R. Töpfer, BAT (Berry  
523 Analysis Tool): A high-throughput image interpretation tool to acquire the number,  
524 diameter, and volume of grapevine berries, Vitis 52 (3) (2013) 129–135.
- 525 [10] M. Grossetete, Y. Berthoumieu, J.-P. Da Costa, C. Germain, O. Lavialle, G. Gre-  
526 nier, et al., Early estimation of vineyard yield: site specific counting of berries  
527 by using a smartphone., in: Infomation Technology, Automation and Precision  
528 Farming. International Conference of Agricultural Engineering-CIGR-AgEng 2012:  
529 Agriculture and Engineering for a Healthier Life, Valencia, Spain, 8-12 July 2012,  
530 CIGR-EurAgEng, 2012, pp. C–1915.
- 531 [11] A. Aquino, M. P. Diago, B. Millán, J. Tardáguila, A new methodology for esti-  
532 mating the grapevine-berry number per cluster using image analysis, Biosystems  
533 engineering 156 (2017) 80–95. doi:10.1016/j.biosystemseng.2016.12.011.
- 534 [12] A. Aquino, I. Barrio, M.-P. Diago, B. Millan, J. Tardaguila, vitisBerry: An Android-  
535 smartphone application to early evaluate the number of grapevine berries by means  
536 of image analysis, Computers and Electronics in Agriculture 148 (2018) 19–28. doi:  
537 10.1016/j.compag.2018.02.021.
- 538 [13] M. Herrero-Huerta, D. González-Aguilera, P. Rodriguez-Gonzalvez, D. Hernández-  
539 López, Vineyard yield estimation by automatic 3D bunch modelling in field condi-  
540 tions, Computers and Electronics in Agriculture 110 (2015) 17–26. doi:10.1016/

- 541           j.compag.2014.10.003.
- 542 [14] S. Liu, M. Whitty, S. Cossell, A lightweight method for grape berry counting based  
543         on automated 3d bunch reconstruction from a single image, ACRA, Seattle, WA,  
544         USA, 2015.
- 545 [15] L. Luo, Y. Tang, X. Zou, C. Wang, P. Zhang, W. Feng, Robust grape cluster  
546         detection in a vineyard by combining the adaboost framework and multiple color  
547         components, Sensors 16 (12) (2016) 2098. doi:10.3390/s16122098.
- 548 [16] L. Luo, Y. Tang, X. Zou, M. Ye, W. Feng, G. Li, Vision-based extraction of spatial  
549         information in grape clusters for harvesting robots, Biosystems Engineering 151  
550         (2016) 90–104. doi:10.1016/j.biosystemseng.2016.08.026.
- 551 [17] R. Perez-Zavala, M. Torres-Torriti, F. A. Cheein, G. Troni, A pattern recognition  
552         strategy for visual grape bunch detection in vineyards, Computers and Electronics  
553         in Agriculture 151 (2018) 136–149. doi:10.1016/j.compag.2018.05.019.
- 554 [18] N. Otsu, A threshold selection method from gray-level histograms, Automatica  
555         11 (285-296) (1975) 23–27.
- 556 [19] O. Mirbod, L. Yoder, S. Nuske, Automated measurement of berry size in images,  
557         IFAC-PapersOnLine 49 (16) (2016) 79–84. doi:10.1016/j.ifacol.2016.10.015.
- 558 [20] K. Dahal, S. Bhattacharai, A. Kicherer, D. Oag, K. Walsh, Assessment of hen and  
559         chickendisorder for marketable yield estimates of table grape using the berry analysis  
560         tool, Vitis: Journal of Grapevine Research 57 (1) (2018) 27–34.
- 561 [21] R. C. Gonzalez, R. E. Woods, S. L. Eddins, Digital image processing using MAT-  
562         LAB, Pearson Education India, 2004.
- 563 [22] J. H. Friedman, J. L. Bentley, R. A. Finkel, An algorithm for finding best matches in  
564         logarithmic expected time, ACM Transactions on Mathematical Software (TOMS)  
565         3 (3) (1977) 209–226. doi:10.1145/355744.355745.
- 566 [23] P. Dry, B. Coombe, Grapevine growth stages-the modified e-l system, Viticulture 1  
567         (2004) 150–154.
- 568 [24] S. Liu, S. Cossell, J. Tang, G. Dunn, M. Whitty, A computer vision system for early  
569         stage grape yield estimation based on shoot detection, Computers and Electronics  
570         in Agriculture 137 (2017) 88–101. doi:10.1016/j.compag.2017.03.013.
- 571 [25] S. Liu, X. Li, H. Wu, B. Xin, J. Tang, P. R. Petrie, M. Whitty, A robust automated

- 572 flower estimation system for grape vines, Biosystems Engineering 172 (2018) 110 –  
573 123. doi:10.1016/j.biosystemseng.2018.05.009.
- 574 [26] A. Aquino, B. Millan, D. Gaston, M.-P. Diago, J. Tardaguila, vitisFlower: Devel-  
575 opment and Testing of a Novel Android-Smartphone Application for Assessing the  
576 Number of Grapevine Flowers per Inflorescence Using Artificial Vision Techniques,  
577 Sensors 15 (9) (2015) 21204–21218. doi:10.3390/s150921204.
- 578 [27] J. Grimm, K. Herzog, F. Rist, A. Kicherer, R. Töpfer, V. Steinhage, An adapt-  
579 able approach to automated visual detection of plant organs with applications in  
580 grapevine breeding, Biosystems Engineering 183 (2019) 170–183.
- 581 [28] Q. Wang, Y. Tang, Z. Xiao, Grape size detection and online gradation based on  
582 machine vision, International Journal of Agricultural and Biological Engineering  
583 10 (1) (2017) 226–233. doi:10.25165/ijabe.v10i1.2517.