Visual Yield Estimation in Vineyards: Experiments with Different Varietals and Calibration Procedures

Stephen Nuske  
*Carnegie Mellon University*

Supreeth Achar  
*Carnegie Mellon University*

Kamal Gupta  
*Carnegie Mellon University*

Srinivasa G. Narasimhan  
*Carnegie Mellon University*

Sanjiv Singh  
*Carnegie Mellon University*

Follow this and additional works at: [http://repository.cmu.edu/robotics](http://repository.cmu.edu/robotics)

Part of the [Robotics Commons](http://repository.cmu.edu/robotics)
Visual Yield Estimation in Vineyards: Experiments with Different Varietals and Calibration Procedures

Stephen Nuske, Supreeth Achar, Kamal Gupta, Srinivasa Narasimhan and Sanjiv Singh

CMU-RI-TR-11-39

December 2011

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

© Carnegie Mellon University
Abstract

A crucial practice for vineyard managers is to control the amount of fruit hanging on their vines to reach yield and quality goals. Current vine manipulation methods to adjust level of fruit are inaccurate and ineffective because they are often not performed according to quantitative yield information. Even when yield predictions are available they are inaccurate and spatially coarse because the traditional measurement practice is to use labor intensive, destructive, hand measurements that are too sparse to adequately measure spatial variation in yield. We present an approach to predict the vineyard yield automatically and non-destructively with cameras. The approach uses camera images of the vines collected from farm vehicles driving along the vineyard rows. Computer vision algorithms are applied to the images to detect and count the grape berries. Shape and texture cues are used to detect berries even when they are of similar color to the vine leaves. Images are automatically registered together and the vehicle position along the row is tracked to generate high resolution yield predictions. Results are presented from four different vineyards, including wine and table-grape varieties. The harvest yield was collected from 948 individual vines, totaling approximately 2.5km of vines, and used to validate the predictions we generate automatically from the camera images. We present different calibration approaches to convert our image berry count to harvest yield and find that we can predict yield of individual vineyard rows to within 10% and overall yield to within 5% of the actual harvest weight.
Contents

1 Introduction .................................................. 1

2 Related Work ............................................... 2

3 Berry Detection ............................................. 3
   3.1 Detecting Potential Berry Locations with a Radial Symmetry Transform .......... 3
   3.2 Classifying Interest Points that Appear Similar to Berries ....................... 4
   3.3 Group Neighboring Berries into Clusters ............................................. 5

4 Registration of image sequence to vines ............................................. 5
   4.1 Tracking vehicle motion .............................................................. 5
   4.2 Detecting vine stakes ............................................................... 5
   4.3 Overlapping images .................................................................. 6

5 Converting Berry Detections into Yield Predictions ................................... 7
   5.1 Occlusion ratio ........................................................................... 7
   5.2 Calibration of occlusion ratio from destructive hand samples ................. 8
   5.3 Calibration of occlusion ratio from prior year harvest ......................... 8

6 Results ........................................................................ 8
   6.1 Datasets .................................................................................. 8
       6.1.1 Gerwurztraminer ................................................................. 8
       6.1.2 Traminette and Riesling ..................................................... 8
       6.1.3 Flame Seedless ................................................................. 10
       6.1.4 Chardonnay ................................................................ 11
   6.2 Berry Detection Performance ...................................................... 11
   6.3 Berry Count Correlation to Yield .................................................. 13
   6.4 Calibrating to Harvest Yield .......................................................... 14
   6.5 Predicting Yield from Prior Harvest Calibration .................................. 15
   6.6 Predicting Yield using Hand Calibration .......................................... 16

7 Conclusion and Future Work ............................................................. 17

8 Acknowledgements ..................................................................... 20
1 Introduction

Forecasting harvest yield is an important task for any vineyard grower. Yield predictions are critical for deciding when and how to make adjustments to their vines to optimize growth, prepare a grower for the harvest operation, prepare a grower for shipping their crop, storing their crop and also selling their crop on the market. Typical yield predictions are performed using knowledge of historical yields and weather patterns along with measurements manually taken in the field. The current industry practice for predicting harvest yield is labor intensive, expensive, inaccurate, spatially sparse, destructive and riddled with subjective inputs. Typically, the process for yield prediction is for workers to sample a certain percentage of the vineyard and extrapolate these measurements to the entire vineyard. The sample size is often too small in comparison to the spatial variability across a vineyard and as a result the yield predictions are inaccurate and spatially coarse.

Figure 1: Example camera image of Gerwurztraminer wine grapes captured at véraison. Automatically detecting the grape crop within imagery such as this is difficult because of lack of contrast to the leaf background.

There is a gap between the methods available to predict the yield in a vineyard and the needs of a vineyard manager to make informed decisions for their vineyard operations with accuracy and precision. We present a new technology that can make dense predictions of harvest yield efficiently and automatically using cameras. Here we report results of an approach to automatically detect and count grapes to forecast yield with both precision and accuracy. The approach is to take conventional visible light cameras through a vineyard to image the vines and detect the crop and predict yield.

Traditional manual yield estimates look to sample the average number of grape clusters per-vine, the average number of grape berries per-cluster and average berry weight. Our approach is to estimate the total number of berries, essentially combining clusters per-vine and berries per cluster in the one measurement. Clusters per vine and berries per cluster account for 60% and 30% of variation in yield per vine respectively, therefore 90% of the variation in yield is accounted with accurate berry counts. Furthermore, the number of berries per-vine is a good measure to obtain because it is fixed from fruit-set all the way until harvest, unlike cluster weight for which a multiplier must be guessed and applied.

The challenges in visually detecting grape berries is their lack of color contrast to the background, which is often similarly colored to the grapes, and also occlusions causing not all grapes to be visible. Furthermore, localizing detected fruit is essential to avoid double counting fruit, but difficult because the fruit has similar appearance and difficult to distinguish when considering overlapping images. An example of the difficulties
of visually detecting grape crop can be seen in Fig. 1. Lack of color contrast is an important issue that occurs in the white-grape varieties and all the grape varieties prior to véraison (the onset of color development). We specifically address the issues of lighting and lack of color contrast, by using shape and texture cues for detection.

The issue of occlusion means it is not possible detect and count all berries on a vine. However, our detection of grape berries is precise, ensuring that there are very few false positives. The result of precise detection is that our berry count is a reliable measurement of yield, despite the fact that our algorithm only counts a percentage of all the grape berries on a vine. We present two approaches to calibrate the image berry count measurement to harvest yield, one way from prior harvest data and another from a small number of hand samples.

Preliminary results of our approach were reported in Nuske et al. [1]) and we extend our prior work in a three ways:

1. we present a method to automatically measure position of vehicle in vineyard and localize our berry detections to specific vines,

2. we demonstrate two different approaches to calibrate our image berry measurements to harvest yield

3. we present experimental results with image data collected at various stages during the growing season and in both wine and table-grape vineyards

We deployed our method on four different vine varieties and conducted experiments in which manual per-vine harvest weights were collected and used as ground truth to evaluate our automated yield measurements. The size of the experiment is significant, including 948 individual vines, totaling 2.5km vines, including four different grape varieties. Our method predicts weight with approximately 5% error of the overall actual harvest yield and approximately 10% error of the harvest weight for individual vineyard rows.

2 Related Work

Current practices to forecast yield are inaccurate because of sampling approaches that tend to adjust towards historical yields and include subjective inputs (Clingeleffer et al. [2]). The calculation of final cluster weight from weights at véraison use fixed multipliers from historic measurements, Wolpert and Vilas [3]. Unfortunately, multipliers are biased towards healthier vines thus discriminating against missing or weak vines and multipliers for cluster weights vary widely by vineyard, season and variety.

Sensor-based yield estimation in vineyards has been attempted with trellis tension monitors, multispectral sensors, terahertz-wave imaging and visible-light image processing. A dynamic yield estimation system based on trellis tension monitors has been demonstrated (Blom and Tarara [4]) but it requires permanent infrastructure to be installed. Information obtained from multispectral images has been used to forecast yields with good results but is limited to vineyards with uniformity requirements (Martinez-Casasnovas and Bordes [5]). A proof of concept study by Federici et al. [6] has shown that terahertz imaging can detect the curved surfaces of grapes and also has the potential to detect these through occluding thin canopy. The challenge for this approach is to achieve fast scan rates to be able to deploy the scanner on a mobile platform.

Small scale yield estimation based on simple image color discrimination has been developed by Dunn and Martin [7]. This approach was attempted on Shiraz post-véraison (i.e. after color development, very close to harvest) in short row segments. The method would not be applicable for the majority of real world examples where the fruit appears over a background of similarly-colored leaves, as is the case in white grape varieties and in all varieties before véraison. Other recent small scale experiments in vineyard work is Dey [8] present a method for classifying plant structures, such as the fruit, leaves, shoots based on 3D reconstructions generated from image sequences which unlike our work is sensitive to slight wind whilst imaging. Other crop detection based on computer vision methods using color pixel classification or shape analysis has been attempted on various fruit types – Jimenez et al. [9] provides a summary of fruit detection work, Singh et al. [10] present a method for detecting and classifying fruit in apple orchards and Swanson et al. [11] use the shading on the curved surfaces of oranges as a cue for detection.
3 Berry Detection

We deploy a sideways-facing camera on a small vineyard utility vehicle, see an illustration in Fig. 6. The images capture the vines and are processed with our algorithm to detect and count the crop. In traditional vineyard yield estimation the crop components that are measured to derive a final estimate are:

1. Number of clusters per vine (60% of the yield variation)
2. Number of berries per cluster (30% of the yield variation)
3. Berry size (10% of the yield variation)

These three components combine to describe all the variation in harvest yield. Current practice is to take samples of each of these components to compute an average and compute the final yield. We take an approach to estimate the first two of these items together in one measurement – that of the number of berries per vine. The reason being that it is difficult, especially late in the season, to delineate the boundaries of clusters within images. However, it is possible to count the total number of berries seen, hence combining the two components – number of clusters per vine and berries per cluster – into one measurement: berries per vine.

An interesting observation can be drawn that humans are better at counting clusters per vine and weighing individual clusters, whereas conversely it seems robotic sensing struggles to accurately count mature grape clusters. Instead it is easier to use robotic sensing to count the number of berries on vine, a measure which would not be possible for a human to directly produce.

Our approach does not attempt to measure berry weight. However, we account for 90% of the harvest yield variation with berries per vine ([2]). Furthermore, instead of taking a small sample and extrapolating, we aim to estimate non-destructively the specific yield at high resolution across the entire vineyard. Hence, we will not introduce sampling errors into the process.

Our algorithm to detect the berries in imagery has three distinct stages:

1. Detecting potential berry locations with a radial symmetry transform (Section 3.1)
2. Identifying the potential locations that have similar appearance to grape berries (Section 3.2)
3. Group neighboring berries into clusters (Section 3.3)

3.1 Detecting Potential Berry Locations with a Radial Symmetry Transform

The first step of our algorithm is to find points with a high level of radial symmetry as these points are potential centers for grape berries, see Fig. 2b. To find these points, we use the radial symmetry transform of Loy and Zelinsky [12]. The algorithm is robust to the issues of lighting and low color contrast, which cause problems for the existing crop detection techniques that rely on simple color discrimination (Jimenez et al. [9], Dunn and Martin [7]). The approach detects the centers of berries of all colors, even those that are similarly colored to the background leaves.

The radial symmetry transform requires us to know the radii of the berries as seen in the image ahead of time. The berry radii (in pixels) are dependent on the focal length of the camera, actual berry size and the distance from the camera. The focal length is kept fixed in our tests and the vehicle maintains a relatively constant distance from the vines. There is still variation in the radius the berries appear in the image from differing berry sizes and also some variation in location within the vine. We account for this variation by searching for radially symmetric points over a range of possible radii, N. Individual radii are denoted as n.

The transform first computes the locally normalized gradient $g$ with magnitude and orientation information at each image pixel. In a Hough Transform like setup, each edge pixel $p$, with a gradient value above a threshold $T$ votes for possible points of radial symmetry $p_s(p)$ given by:

$$p_s(p) = p + n \frac{g(p)}{\|g(p)\|} \quad (1)$$

for each radius $n$, these votes from the edge pixels are counted in a vote image $F_n$ which is then smoothed out with $A_n$, a 2D Gaussian filter, to produce $S_n$, the radial filter response at radius $n$. These filter responses
at different radii are then combined to form the overall radial filter response $S$ which is given by.

$$S_n = F_n \ast A_n$$  \hspace{1cm} (2)

$$S = \max_{n \in N} S_n$$  \hspace{1cm} (3)

We compute local maxima in the response image $S$ with a non-maximal suppression, and threshold to find the potential centers. We choose the threshold to ensure as many berry centers are detected as possible, at the expense of many false positive detections. We use the following stages in the algorithm to filter out the false positives.

![Figure 2: Example images showing the functioning of our visual berry detection algorithm on a Gerwurztraminer vine. Input image is seen in Fig 1. b potential berry locations in the image that have been detected as having radial symmetry. c points marked blue have been classified as having appearance similar to a berry. d classified berries that neighbor other classified berries are clustered together.](image)

3.2 Classifying Interest Points that Appear Similar to Berries

The next stage in our algorithm is to classify the detected points which appear most like grapes, see Fig. 2c. We first take a patch in the image around each detected center. The patch size has a radius defined by the previous radial symmetry detector step. We then compute features from that image patch. The features we
use are a combination of color and texture filters, which combine to form a 34 dimensional feature vector. We use the three RGB channels, the three L*a*b color channels and Gabor filters with 4 scales and 6 orientations. The features are not chosen specifically for the grape detection task – we use generic low-level image features.

We take a small number of training samples from our datasets, by selecting a random subset of images and manually define in the images which regions have grape berries. We compute our features in these regions which correspond to the positive examples of the appearance of berries. For negative examples we compute features at radially symmetric interest points outside of our defined crop areas.

Given an input image we take each radially symmetric interest point, compute the feature vector, and apply the k-Nearest Neighbors algorithm. The k-Nearest Neighbors algorithm computes the distance in feature space to every point in the training set and determines whether the nearest neighbors are positive berry examples or negative. If the k closest positive examples are closer than the k closest negative examples, that interest point is classified as a berry. We use a value of three for k, which empirically seems to function appropriately.

3.3 Group Neighboring Berries into Clusters

After classification of the interest points, a small number of false positives still remain. Most of the remaining false positive detections are isolated while grape berries naturally occur in clusters so we apply contextual constraints that dictate that there should be a minimum number of berries in a cluster. We cycle through each classified berry, computing the distance to every other berry, and remove berries that do not have at least 5 other berries within their immediate neighborhood, which we define as a radius of 150 pixels. The process results in the clustered berries, which are the output of our entire algorithm, see Fig. 2d.

4 Registration of image sequence to vines

Vines are usually trained along straight rows, where individual vines are planted at a fixed spacing, which are typically spaced at 6 to 8 feet. In the experiments we present later, we measure the harvest weight of the fruit for every individual vine in our test set to evaluate our image-based prediction. We collect a stream of camera images from the vehicle driving along vineyard rows and to produce a prediction for each vine we need to align the images to specific vines. Another issue is that there is significant overlap between consecutive images which results in our algorithm detecting the same grape berries in more than one image.

In our preliminary work, Nuske et al [1], we registered the images by manually defined the boundary of individual vines in the images and manually cropped out overlapping content between consecutive images. Since that prior work we have developed a framework to automatically maintain track of where the vehicle is along the row of vines and also to automatically account for overlapping images.

4.1 Tracking vehicle motion

To align images to vines, we use a visual odometry algorithm, [13], that estimates the position of the vehicle along the row. We use a stereo camera facing down the row with a visual odometry algorithm that estimates the motion by triangulating and tracking features within the stereo pair. The stereo camera is in addition to the main camera that is pointed sideways detecting the grape berries. The visual odometry algorithm precisely tracks the motion of the vehicle from frame to frame. Synchronizing the two cameras together gives an estimate of the position of the vehicle for when each of the main camera images was recorded. However, over time the visual odometry estimate of position drifts because of integrated error. The drift is only around 1% of distance travelled, but over long vineyard rows the error could be anywhere between 1-3m.

4.2 Detecting vine stakes

To correct for drift we detect the stakes used to support the vine trellis, which are erected at fixed spacings, and use these as landmarks to correct our estimate of the vehicle’s position. We use the main sideways camera to detect the stakes by searching in the images for vertical lines in the images. The algorithm finds areas of the image that is colored similar to the metal stakes and searches within those areas for vertical
edges, see Fig. 3. Because the stakes are at fixed spacings, when we detect the location of the stake we know our exact position along the row and correct for the visual odometry drift. Detecting a stake in a single image is not sufficient because we need to detect the stake in consecutive frames to allow for triangulation of the stakes position with respect to the camera, which then allows for correction of the position of the camera along the row, see Fig. 4.

Figure 3: Example demonstrating the stake detection algorithm used to help calculate the vehicle’s position along each row. A metal stake, as seen in the raw image, is fixed in the ground at every vine location, at fixed spacings along the row. We detect the stakes and count the berries between two neighboring stakes. The visualization of the detection algorithm is colored with blue regions demonstrating areas with similar color to the stake, magenta lines are the edges detected within those regions and the yellow line with the star in the middle demonstrates the detected stake which is the longest vertical edge. The blue line is drawn between the camera position and the stake detected for the image currently being processed by the algorithm.

4.3 Overlapping images

There is significant overlap between consecutive images; as a result we detect the same berries twice. We need the algorithm to be aware of the overlap and avoid counting berries twice. We have two ways to deal with the overlap of images. One method is suited to vertical shoot positioned vines where fruit hangs in a single vertical plane making it possible to treat the images as 2D orthographic projections. We compute the overlap, crop each image and form a mosaic. Figs. 5a and 5b present examples of a mosaic for individual vines.

To compute the overlap we take the position of the camera along the row, apply the calibration parameters of the lens, the distance of the camera from the vine and compute which portion of the vine is in the view for each image. We analyze the overlapping areas and we take the image portion with the most recorded berry detections. Using the image with the most detected berries for a given vine portion helps reduce the effect of occlusion when a part of the canopy occludes fruit from one image and not another.

The alternative method we use for dealing with overlapping images is designed for vines where the fruit does not hang in a single plane. In particular one of our experiments was in a table grape vineyard, planted with a split V gable, where the fruit hangs on an angled trellis, see Figs. 5c and 7g. In this type of vineyard the fruit does not hang in a single fruiting wall, making the overlapping images a more difficult 3D problem. Some fruit hangs close to the camera and some farther away, and a simple 2D mosaic will cause double counting or missed detections. Here we explicitly keep track of berries that we detect from frame to frame, by matching features [14] between neighboring images that lie at or near detected berries. We do not count berries that have features matched to berries detected in prior frames.
5 Converting Berry Detections into Yield Predictions

Earlier we discussed how to detect the berries in camera images, which is the basis of our crop measurement. This section now discusses how to take the image berry measurement and produce a harvest yield prediction.

The most important consideration here is that whilst the total number of berries on the vine is the best forecaster of eventual yield, our measurement of the berries is not complete, simply because not all berries are visible to the camera. There are number of occlusion sources causing berries to be hidden from view. One source of occlusion is that some leaves of the vine canopy lie in front of the grape clusters, another is that some grape clusters lie in front of other grape clusters, and finally even clusters that are in full view have berries at the front of the cluster that occlude berries at the back. Occlusions could be reduced with an improved imaging setup that has multiple cameras that help peer around leaves and clusters. Another possible improvement is a physical action or device on the vehicle that can physically move the canopy and clusters to reveal occluded fruit. However, even though we acknowledge there are steps that could be taken to reduce occlusions, we assume that occlusions could never be completely removed and therefore a method is required to account for those berries that are not seen. Following is our approach to take our image berry count, account for hidden berries and generate a yield estimate.

5.1 Occlusion ratio

After analyzing the harvest data against our detected berry count, we find there is consistency in the level of hidden berries from one section of vines to another. In fact, computing a ratio between berries detected and the harvested fruit on one portion of data is sufficient for predicting yield on another portion of data by applying the linear ratio to the detected berry count. Of course this relies on knowing the mean occlusion ratio of a given vineyard, which would be needed at the time of imaging, because of course it would defeat the
purpose of predicting yield if it were necessary to wait for the harvest yield to measure the occlusion ratio. We see two methods for acquiring the occlusion ratio at the time of imaging, well in advance of harvest.

5.2 Calibration of occlusion ratio from destructive hand samples

One approach takes a small number of hand measurement samples in a vineyard at the time of imaging. The vines are imaged first and then on a small sample of vines the fruit is destructively removed and produce a total berry count. Rather than counting individual berries, the hand measurements involve measuring the average berry weight and the total fruit weight of a vine. The process is repeated for all vines in the sample set.

The hand fruit weight is projected to harvest using the ratio between current berry weight and expected berry weight at harvest. Taking the hand estimate against the image berry count for these specific vines produces an occlusion ratio that can be estimated well in advance of harvest, and applied to predict yield of the remaining vines that were not destructively sampled.

5.3 Calibration of occlusion ratio from prior year harvest

The second method we evaluate for calibrating occlusion ratio is to use harvest data from prior growing seasons. We have analyzed harvest data from vines trained and prepared in a similar manner from year to year and noticed similar occlusion ratio. This method may not be applicable to all vineyards, especially where occlusion may change throughout the season. However, for vineyards trained with vertical shoot position trellises, that use leaf pulling for sun exposure, we have seen consistent occlusion ratio from year to year. The advantage of calibrating from a prior harvest season is that hand samples are not necessary. The method simply takes a total measurement of the fruit harvested and compared against the berry count detected in the imagery.

In the following results section we compare the accuracy of the various approaches to predict harvest yield.

6 Results

6.1 Datasets

We have conducted experiments at four different vineyards, including the following varietals: Gerwurztraminer, Traminette, Riesling, Flame Seedless, Chardonnay. The Gewurtztraminer, Traminette, Riesling are wine grape vineyards grown with vertical shoot positioning (VSP) training system, the Flame Seedless vineyard is a table-grape grown with a split-V gable system and the Chardonnay vineyard grows a wine grape in a semi-VSP system.

We demonstrate our method at a variety of stages during the growing season, from just after the fruit begins setting right up till just before harvest. Where the berries range from one tenth their final size to almost fully grown. See Table 1 for details of the different datasets. See Fig. 5 for image examples from the different datasets.

In each of the datasets, besides the Gerwurztraminer dataset, we collect harvest weights of the fruit to evaluate against our image measurements.

6.1.1 Gerwurztraminer

The Gerwurztraminer dataset was collected in Portland, New York, just before véraison, before color development, and the berries were green in color, see Fig. 1. Only 5 vines were included in the dataset and we used it purely for developing the berry detection algorithm. For this short dataset we use a Canon SX200IS capturing images by hand.

6.1.2 Traminette and Riesling

The Riesling and Traminette datasets were collected from an approximately one acre plot in Fredonia, New York. The Riesling cultivar is a ‘White Riesling’ Vitis vinifera and the Traminette is an intraspecific hybrid.
Figure 5: Example images of the four different varietals from our yield prediction experiments.
Table 1: Dataset Description

<table>
<thead>
<tr>
<th>Variety</th>
<th>Location</th>
<th>Date</th>
<th>Trellis</th>
<th>Time before harvest at imaging</th>
<th>Mean berry weight at imaging</th>
<th>Num. vines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gerwurztraminer</td>
<td>Portland, NY</td>
<td>Aug. 2010</td>
<td>VSP</td>
<td>45 days</td>
<td>N/A</td>
<td>5</td>
</tr>
<tr>
<td>Traminette</td>
<td>Fredonia, NY</td>
<td>Sep. 2010</td>
<td>VSP</td>
<td>10 days</td>
<td>1.6g</td>
<td>98</td>
</tr>
<tr>
<td>Riesling</td>
<td>Fredonia, NY</td>
<td>Sep. 2010</td>
<td>VSP</td>
<td>10 days</td>
<td>1.5g</td>
<td>128</td>
</tr>
<tr>
<td>Flame Seedless</td>
<td>Delano, CA</td>
<td>June 2011</td>
<td>Split-V</td>
<td>40 days</td>
<td>4.0g</td>
<td>88</td>
</tr>
<tr>
<td>Chardonnay</td>
<td>Modesto, CA</td>
<td>June 2011</td>
<td>Semi-VSP</td>
<td>90 days</td>
<td>0.15g</td>
<td>636</td>
</tr>
</tbody>
</table>

We used four rows of Traminette vines and four rows of Riesling varieties, 224 vines in total. The Traminette were at 8ft spacing and Riesling were at 6ft spacing, which totaled 450m of vines.

The vines in this acre plot were vertically shoot positioned and basal leaf removal was performed in the cluster zone, a practice performed by vineyard owners to expose the fruit to the sun to change the flavor characteristics of the grapes. The basal leaf removal also makes yield estimation feasible towards the end of the growing season because the occluding canopy is removed from the fruit-zone. On the Traminette vines the basal leaf removal was performed just on the East facing side of the row and performed on both sides of the Riesling vines. Our tests captured images from the East side of the rows.

The Traminette and Riesling vines vines are white grape varieties, the images of the crop were collected one week prior to harvest, and even though the grapes have at this late stage developed full coloration and still had similar coloring to the background of leaves.

For the Traminette and Riesling vines experiments we use a Canon SX200IS, mounted facing sideways on the vehicle at the same height of the fruit zone, capturing images of the crop. The camera is set in continuous capture mode, recording images at 3264 x 2448 resolution, at approximately 0.8Hz. We mount halogen lamps facing sideways, illuminating the field of view of the camera to improve the lighting of the fruit-zone, which is often in the dark shadows of the canopy. The camera vehicle is driven along the rows in the vineyard capturing images at approximately 0.5m/s.

6.1.3 Flame Seedless

The Flame Seedless dataset was collected at a 88 vine row of a split-V gable vineyard, in Delano, California. The vines are spaced at 7 feet spacings, and trained in a split system where each vine grows two distinct sections of fruit that hangs at an angle on each side of the row. We divide the vines into two-vine sections and collect harvest data for each two-vine segment, with one measurement for both the North and South side of the split vines, giving a total of 88 individual measurements.

The image data was collected five weeks before harvest at which stage the berries were nearly fully grown but yet to start coloration. The difficulties with this dataset is with the training structure of the vines making a challenge to register images to avoid counting the fruit twice or not at all. The fruit on the split-V gable hangs at a variety of distances from the camera meaning a cluster will appear to the left of another cluster in one image and to the right in another image. The other difficulty is that the fruit is split on two sides, meaning the algorithm must determine which side of the vine the detected fruit is hanging, to avoid counting fruit twice or not at all when imaging from both sides. We discuss earlier how we confront these issues in our algorithm.

The split-V gable creates a canopy curtain that hangs in front of the fruit, for these experiments a vehicle fitted with a canopy trimming toolhead was passed through the vines prior to imaging to make the fruit visible from the row. The conventional practice in this vineyard is to wait until closer to harvest to trim the canopy, which avoids the fruit being burnt. However, in future iterations of our equipment we will make imaging possible without trimming the canopy on these types of vines by using a plastic shield on the side of the vehicle which pushes the canopy up and over the camera as the vehicle drives down the row.

The imaging setup we used for this experiment can be seen in Fig. 6. The setup includes a Nikon D300s camera at 4288 x 2848 pixel resolution faced sideways imaging the fruit. We have the camera mounted
low down on the vehicle pointed upwards angle at the fruit to see below the trimmed canopy. We use an AlienBees ARB800 ring flash mounted around the lens to provide even lighting to the scene. We use a PointGrey BumbleBee2 stereo camera mounted pointed down the row to compute the visual odometry algorithm for estimating the position of the vehicle along the row. Both cameras were triggered by external pulses to keep the cameras in synchronization. The sideways facing camera was triggered at 1Hz and the stereo camera was triggered at 10Hz and the vehicle was driven at approximately 0.35m/s.

Figure 6: Photos of the equipment used during the experiments in Flames Seedless and Chardonnay vineyard. Equipment mounted on an aluminum frame fixed on the back tray of a Kawasaki Mule farm utility vehicle. The four main sensing equipment used are a Nikon D300s color camera facing sideways from the vehicle detecting the fruit, an AlienBees ARB800 ring flash mounted around the lens of the color camera illuminating the scene, a PointGrey BumbleBee2 stereo camera facing back down the row tracking the vehicle motion and a synchronization box generating pulses to keep the two cameras synchronized.

6.1.4 Chardonnay

The Chardonnay dataset was collected in Modesto, California in a vineyard which early in its life was trained in a vertical shoot positioned system and in recent years has not been strictly trained in this regiment and now considered only semi vertical shoot positioned. We collected images on 636 vines on 6 rows of this vineyard, the rows contained between 107 and 123 vines planted at 8 feet spacings. The images we collect just after the berries began to set at 12 weeks before harvest. At this stage the berries are very small, between 3-5mm in diameter and one tenth of their final weight.

We use the same imaging setup as the Flame Seedless dataset described before, except that we collected the images whilst driving at a faster velocity at 0.75m/s and hence increased the sideways facing camera frame-rate to 2Hz.

6.2 Berry Detection Performance

We first evaluate the performance of our berry detection algorithm. Qualitatively we present visual results of the berry detection of all five varieties in Fig 7. Quantitatively we analyze the detection performance statistically by selecting five images from each of three different datasets; Gerwurztraminer, Traminette and Riesling. We processed the images with the berry detection algorithm and also manually counted detection statistics, presenting these results in Table 2. The shows that our algorithm mistakenly detects only a minimal number of false berries, giving it a very high precision rate. However, it is conservative, it does not detect around 30% of berries that are visible in the images and therefore has a high false negative count and therefore a moderate recall rate.

To gain an understanding of what part of the algorithm are most responsible for the false negatives detections we break-down the false negatives into the three stages of the algorithm; False detections that
Figure 7: Example images demonstrating berry detection in the four different varietals.
Table 2: Berry Detection Statistics. Berry count – The number of berries reported by the algorithm. True positives – The number of berries detected that were actual berries. False positives – The number of false berry detections. False negatives – The number of berries visible in the image that were not detected. Recall – Percentage of visible berries detected. Precision – Percentage of detections that were berries.

<table>
<thead>
<tr>
<th>Variety</th>
<th>Berry Count</th>
<th>True Positives</th>
<th>False Positives</th>
<th>False Negatives</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gerwurztraminer</td>
<td>1073</td>
<td>1055</td>
<td>18</td>
<td>354</td>
<td>74.9%</td>
<td>98.3%</td>
</tr>
<tr>
<td>Traminette</td>
<td>1116</td>
<td>1096</td>
<td>20</td>
<td>658</td>
<td>62.8%</td>
<td>98.2%</td>
</tr>
<tr>
<td>Riesling</td>
<td>784</td>
<td>762</td>
<td>22</td>
<td>657</td>
<td>53.7%</td>
<td>97.2%</td>
</tr>
<tr>
<td>Overall</td>
<td>2973</td>
<td>2913</td>
<td>60</td>
<td>1659</td>
<td>63.7%</td>
<td>98.0%</td>
</tr>
</tbody>
</table>

are not detected by the radial-symmetry detector (Section 3.1), those that are misclassified (Section 3.2) and those that are not clustered to neighboring berries (Section 3.3). Table 3 presents the false negative break-down by algorithm stage. The table shows that around 60% of all missed detections are caused by the radial symmetry transform, around 30% are classified as non-berry and only 10% of the false negatives are to be blamed on the clustering. We show in the following section that, even with these false negatives, we can still acquire accurate yield prediction because of the high precision rate. However, to further improve performance we could look at modifying the radial symmetry transform to improve the number of berries it can detect without drastically increasing the false detections.

Table 3: Break-down of False Negatives

<table>
<thead>
<tr>
<th>Variety</th>
<th>Not-detected</th>
<th>Mis-classified</th>
<th>Not-clustered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gerwurztraminer</td>
<td>51.7%</td>
<td>31.9%</td>
<td>16.4%</td>
</tr>
<tr>
<td>Traminette</td>
<td>73.9%</td>
<td>16.0%</td>
<td>10.0%</td>
</tr>
<tr>
<td>Riesling</td>
<td>53.9%</td>
<td>40.2%</td>
<td>5.9%</td>
</tr>
<tr>
<td>Overall</td>
<td>61.1%</td>
<td>29.0%</td>
<td>9.7%</td>
</tr>
</tbody>
</table>

6.3 Berry Count Correlation to Yield

We compare our berry counts against actual harvest weights collected from the Traminette, Riesling, Flames Seedless and Chardonnay datasets. First, we register images together, and assign registered images to specific vines by defining the boundaries of the vines within the images, cropping-out overlapping content to avoid double counting. For the Traminette and Riesling datasets we conducted this process manually, but the more recent Flame Seedless and Chardonnay datasets we deployed the stereo camera to perform this process automatically.

Once registered to specific vines, we compare our automated berry counts with the harvest crop weights, see Fig. 8 for details. The figure shows the raw datapoints in the correlation and the distribution of measurements.

We saw in Table 2 that our detection recall rate is not high and we also know that occlusions will cause even more berries to not be counted by our algorithm – yet despite these issues our automatically generated berry counts produced a linear relationship with actual harvest crop weights with correlation score ranging from $r^2 = 0.6$ to 0.73 depending on the dataset, see Fig. 8 for detail. The correlation scores quantifies the ability of our method to successfully measure variation in yield from vine to vine.

Reasons that our measurements achieve good correlation are first through the high precision of our detection algorithm which rarely counts false positives and also because the occlusion level and the percentage of visible berries that are missed has reasonable constancy across the vineyard. Further increasing the correlation score could come from possible improvements to the detection algorithm and including a method
Figure 8: Correlation between our detected berry count and harvest crop weights. The black lines show the one-sigma standard deviation within the measurements, the red line represents a linear fit and each of the blue data points represents the raw measurement of a single vine. The caption below shows the r-square correlation score. These graphs illustrate our method generating a non-destructive measurement at every vine, whereas in conventional practice very sparse destructive samples are taken.

to estimate the berries that are not visible to the camera.

6.4 Calibrating to Harvest Yield

Now we have seen a linear correlation between our image berry counts and harvest yield. We study the calibrate portion of the data against harvest data and predict the yield in other portions of the data.

We fit a function to a section of vines that provides a mapping from berry count to harvest weight, and calibrate for the berries that are out of view and missed by the detection algorithm. Once we have functions calibrated from portions of our data we evaluate how accurate our berry counts are at predicting the total weight of other sections of vines for which we have not calibrated our measurements. Fig. 9 presents graphs of the estimated weights for individual sections, for equal comparison we use 20 vines from each section. The graphs show on average we can predict the yield to within 10% on the 20 vine segments.

Now we analyze the effect that the number of samples in the calibration set has on accuracy. We take each dataset and randomly draw a fixed number of datapoints which forms our calibration set. We take the set, derive a calibration function, apply the function to the remaining vines in the dataset to predict the
yield from the image berry count. We repeat 10,000 times and then repeat the whole process with varying sizes of calibration set. From this trial we accumulate the average error between the harvest yield and the predicted yield from image berry count. We present these statistics in Fig. 10. The figure indicates that increasing that while increasing the size of the calibration set from 5 to 20 samples reaps substantial accuracy improvements, however increasing the set larger than 20 samples does not reap significant more accuracy. Comparing error for individual vines compared to overall error, it is clear that the individual vine error is high but over large datasets the individual error cancels out and overall error is much lower.

### 6.5 Predicting Yield from Prior Harvest Calibration

The previous section presented results analyzing the calibration of berry count to harvest yield. The calibration functions were generated using the harvest data itself which is not practical in a real world system because we need to generate predictions well in advance of harvest. In this section we show that the harvest data can be used from prior seasons. In fact we do not currently have data from the same vineyards from consecutive growing seasons and we demonstrate calibrating the berry counts from prior harvest seasons in another vineyard grown in a similar trellis system.

We take the Traminette and Riesling datasets, collected in 2010, and apply is to the Chardonnay dataset.
from 2011. The Traminette and Riesling datasets were collected in New York state and the Chardonnay dataset in California, the vines are all wine varieties and grown in the Vertical Shoot Positioning training system. Although the Chardonnay vineyard is now not trained strictly and is considered only semi vertical shoot positioned.

We compute calibration functions from the Traminette and Riesling datasets and normalize the calibration based on the average berry weight. We then apply the calibration to the image berry counts we collected in the Chardonnay dataset. Fig. 11a shows a comparison between the data collected in the two vineyards after normalizing for respective berry weights. The graph shows that the Traminette and Riesling vines despite holding much less fruit, do have a trend between the image berry counts in the Chardonnay data. We apply the Traminette and Riesling calibration to the Chardonnay data and show the predicted weight in Fig. 11b. This result demonstrates the prediction of harvest yield 12 weeks out from harvest.

Fig. 13 presents a comparison of the different calibration approaches, we can see the Traminette dataset is slightly more accurate than the Riesling dataset at calibrating the Chardonay data, with the Riesling calibration under-estimating the overall yield by about 4.5% and the Traminette calibration over-estimating the yield by 4%.

It is important to note that it was not possible to convert the calibration from the Flame Seedless harvest to the Chardonnay vineyard. The Flame Seedless vineyard is a table-grape varietal grown in a substantially different split-V gable structure and it appears that the amount of visible berries is quite different for vines grown in different training systems, however we have shown that measurements from two Vertical Shoot Positioned wine vineyards do have similar properties.

6.6 Predicting Yield using Hand Calibration

If data from prior harvests is not available there is another mechanism to generate a calibration of berry counts well in advance of harvest. Taking sparse, destructive, hand samples of the fruit after imaging can be used to calibrate the image berry counts.

Fig. 12a shows a satellite image of the Chardonnay vineyard, highlighted with red to indicate the six vineyard rows that were imaged in our experiment. On bottom row, purple marks indicate the 15 vines in which the hand samples used for calibration. In Fig. 12b a graph shows the relationship between the hand fruit samples collected the day after imaging and the image berry counts. We derive a calibration function
Figure 11: Graphs demonstrating calibration from prior harvest seasons. Traminette and Riesling datasets were collected in 2010 and the Chardonnay dataset collected in the 2011 growing season. After normalizing for berry weight there appears to be a trend between the 2010 and 2011 datasets. The calibration functions computed from the 2010 datasets applied to the Chardonnay image berry counts produces harvest predictions 12 weeks prior from harvest.

from this relationship and predict the crop weight based on the image berry counts of the remaining vines that were not a part of the destructive hand sample. The harvest weight can be predicted by projecting the crop weight based on the average berry weight at the time of sampling to the expected berry weight at harvest.

We compare the different calibration approaches by analyzing the average error for the prediction of the individual vine weights of the Chardonnay dataset in Fig. 13a. The errors are between 17 and 19% for the different calibration approaches, with the hand calibration slightly more accurate. We see that some of the error averages out when comparing yield of entire rows, where the error is now between 7% and 8%. In Fig. 13b we present the error for prediction of the entire yield of the vines in the dataset. The hand calibration was most accurate at 3% error and using calibration from the 2010 Traminette dataset had 4% error. We see an under-prediction of overall weight by 4.5% using the 2010 Riesling dataset calibration. One thing is apparent that despite average absolute per-vine errors of around 18% for all approaches, the overall error is below 5%. To verify that the errors are well distributed we see in Fig. 14 a histogram of errors of the predictions from the hand calibration approach that illustrates the errors are Gaussian and average out to a mean within 5%. For comparison in Fig. 13b we present the estimate taken by extrapolating the hand samples alone, which is the traditional industry practice, and was found to be the least accurate estimate with -13% error.

7 Conclusion and Future Work

We have presented a computer vision method to detect and count grapes to automatically generate non-destructive yield predictions in vineyards with high-resolution. We use a sideways facing camera mounted on a vehicle to capture images of the vines and a stereo camera faced down the row to track the vehicle’s motion.

We evaluate our approach on four different grape varietals including both table and wine grapes. In all
Figure 12: Satellite image of the Chardonnay vineyard. Highlighted with red to indicate rows that were imaged by our setup and marked with purple on the bottom row to indicate where destructive hand samples were measured the day after imaging. Overall six rows were imaged totaling 665 vines, and 15 vines on the bottom row were destructively hand sampled. Graph showing calibration between the hand sample and the corresponding image berry counts.

948 vines were used in our experiments and validation for our predictions was performed with comparison with the harvest weight individually recorded from these vines, we believe this is the largest set of automated crop imaging experiments ever conducted in vineyards.

We demonstrate how to calibrate our image berry count to harvest yield either from prior harvest data or from hand samples and each generates similar accuracy in yield predictions; errors of approximately 18% for individual vines, 10% for individual rows and 4% for the overall harvest yield.

Our results have significance on the future of vineyard operations through our ability to make yield predictions with high resolution opening up the possibility of vineyard owners making precise adjustments to their vines, where previously they have been restricted to using coarse measurements.

There are a number of avenues of work to further improve our approach. First is to find ways to improve the recall rate of the current berry detection system. Another improvement could include augmenting the berry counts with a method that measures berry diameter as an indicator of berry weight, which is known to account for the remaining 10% of the variation in final yield. We also will look to develop an approach to count grape clusters early in the season, even before berries have formed, to give vineyard managers information with maximum time before harvest to make the necessary adjustments to their vines.
Figure 13: Results on the prediction of harvest yield in the Chardonnay dataset. We compare the prediction accuracy when calibrating using a destructive hand sample to calibrating using prior harvest data. We present two statistics in (a); the average absolute error computed for individual vine predictions and the average absolute error for the estimate of row weights. In (b) we present the error for prediction of the entire yield of the vines in the dataset. The hand calibration is slightly more accurate than the calibration from prior harvest season. For comparison we present the estimate taken by extrapolating the hand samples alone, which is the least accurate estimate with 13% error.
Figure 14: Histogram of vine yield errors. To verify that the errors are in fact well-distributed we present in a histogram of errors of the vine predictions from the hand calibration approach that illustrates the errors are Gaussian and average out to a mean within 5%, standard deviation of 20%.

8 Acknowledgements

This work is funded by the National Grape and Wine Initiative, 1415 L Street, Suite 460, Sacramento, California, 95814, USA, info@NGWI.org.

References


