Computers in Industry 99 (2018) 303–312

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Grapevine buds detection and localization in 3D space based on Structure from Motion and 2D image classification

Carlos Ariel Díaz[a,*], Diego Sebastián Pérez[b], Humberto Miatello[c], Facundo Bromberg[b]

[a] Universidad Tecnológica Nacional, Facultad Regional Mendoza, Laboratorio de Inteligencia Artificial DHARMa, Dpto. de Sistemas de la Información, Rodríguez 273, CP 5500, Mendoza, Argentina
[b] Universidad Tecnológica Nacional, Facultad Regional Mendoza, CONICET, Laboratorio de Inteligencia Artificial DHARMa, Dpto. de Sistemas de la Información, Rodríguez 273, CP 5500, Mendoza, Argentina
[c] iNOSUR LLC, New Lab, 19 Morris Ave, Brooklyn, NY 11205, United States

ARTICLE INFO

Keywords:
Computer vision
Grapevine bud detection
Precision viticulture

ABSTRACT

In viticulture, there are several applications where 3D bud detection and localization in vineyards is a necessary task susceptible to automation: measurement of sunlight exposure, autonomous pruning, bud counting, type-of-bud classification, bud geometric characterization, internode length, and bud development stage. This paper presents a workflow to achieve quality 3D localizations of grapevine buds based on well-known computer vision and machine learning algorithms when provided with images captured in natural field conditions (i.e., natural sunlight and the addition of no artificial elements), during the winter season and using a mobile phone RGB camera. Our pipeline combines the Oriented FAST and Rotated BRIEF (ORB) for keypoint detection, a Fast Local Descriptor for Dense Matching (DAISY) for describing the keypoint, and the Fast Approximate Nearest Neighbor (FLANN) technique for matching keypoints, with the Structure from Motion multi-view scheme for generating consistent 3D point clouds. Next, it uses a 2D scanning window classifier based on Bag of Features and Support Vectors Machine for classification of 3D points in the cloud. Finally, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for 3D bud localization is applied. Our approach resulted in a maximum *precision* of 1.0 (i.e., no false detections), a maximum *recall* of 0.45 (i.e. 45% of the buds detected), and a localization error within the range of 259–554 pixels (corresponding to approximately 3 bud diameters, or 1.5 cm) when evaluated over the whole range of user-given parameters of workflow components.

1. Introduction

In this work, we present an approach for the efficient 3D detection and localization of grapevine buds. 3D models were reconstructed from multiple images captured during the winter season in natural field conditions (i.e., natural sunlight and the addition of no artificial elements) using a mobile phone RGB camera.

Grapevine buds were recognized early in viticulture history as one of the most important parts of the plant, mainly because they contain the whole plant productive capacity, from which all sprouts, leaves, bunches, and tendrils grow. In particular, bud bunch fertility, a.k.a. *fruitfulness*, is of particular interest, as it has a direct impact on the main goal of vine production, that is, to increase productivity without affecting fruit quality. It has been shown that bud fruitfulness depends on the amount of sunlight exposure of buds during the period starting at bud initiation in early spring throughout its development stage up to 30 days after bloom [15,21,11,25,35,27]. Shading conditions during this period strongly depend on what we call *shading structure*, consisting in the localization and geometric characterization of those parts of the plant that occlude sunlight, mainly the leaves and bunches that grow after bloom. In addition, sunlight exposure can be used by growers to influence the productivity of the next period by choosing those buds that received the most sunlight exposure. In practice, this happens by deciding pruning procedures late in the winter [23]. There is a balance, however, as unpruned buds will produce vegetation, shading the newly initiated buds, and therefore, affecting the productivity of the next period. The decision of optimal pruning is, therefore, a complex task that must be carefully balanced between: (i) productivity maximization of the starting period determined by buds with maximum sun exposure, and (ii) productivity maximization of the following period determined by the shading conditions resulting from the green vegetation growing from those buds.

[*] Corresponding author.
E-mail addresses: carlos.diaz@frm.utn.edu.ar (C.A. Díaz), sebastian.perez@frm.utn.edu.ar (D.S. Pérez), humberto.m@inosur.com (H. Miatello), fbromberg@frm.utn.edu.ar (F. Bromberg).

https://doi.org/10.1016/j.compind.2018.03.033
Received 2 September 2017; Received in revised form 20 March 2018; Accepted 21 March 2018
Available online 11 April 2018

A solution to the first issue requires measuring the sun exposure of individual buds at regular intervals from initiation to 30 days after bloom and then recovering this value for each bud months later during winter pruning. Sunlight exposure has been measured so far through manual positioning of radiation sensors [25]. These manual procedures, however, are far from efficient for the massive measuring of sunlight exposure of individual plants, not to mention of individual buds. Our work aims to partially fulfill the need for an efficient method for measuring and recording the sunlight exposure of individual buds. The general rationale behind our approach is that it is possible to compute the sunlight exposure of a bud with high-precision when the precise 3D localization of the bud, the shading structure around it, the geo-positioning of the field, and the dates of interest are fed to a sun radiation model [29,8]. It is an ambitious goal, attended partially by the present work that provides a solution to the 3D localization of winter buds. Future work, however, will have to solve the problem of producing the shading model. This could be done by localizing buds from initiation till the end of summer, and then by identifying buds between consecutive 3D modelizations to allow the recording of long-term sun exposure. A solution to the second issue requires a thorough understanding of which summer shading structures result from different winter pruning procedures and trellis systems [11,14]. This demands measuring the shading structure, a procedure which is currently unavailable.

Simulations are a possibility for partially overcoming the inability to reconstruct the shading structure, necessary for solving both issues. There is a line of research that studies different procedures for producing *simulated* whole plant shading structures, including the canopy and bunches [13,16]. They typically require plant architecture and bud localization as input. However, bud localization information, being inexistent, is provided by randomly simulating their position. Our work provides a solution to the latter, while [26] is one of the many studies that provide a solution to the former. Despite being a simulated model, the shading structure has the potential to produce invaluable—and to this day inexistent—information on the (simulated) long-term sun exposure of large bud samples, including months with a fully grown canopy. In particular, with plant architecture before the winter pruning, it is possible to simulate the *backward* shading structure of the previous spring as well as different *forward* shading structures resulting from different pruning treatments.

Finally, we note that both issues require an autonomous system for executing pruning. Historically, pruning procedures have been simplified to be accessible for humans. However, this may change with the extra information provided by 3D modeling, namely, the identification of fruitful buds and predictions of next-period's shading structures. With this information, the resulting optimal pruning may be too sophisticated to be amenable for human execution, requiring autonomous pruning systems.

In addition to measuring sunlight exposure and guiding autonomous pruning, bud localization is also required as part of the measuring processes of other variables of interest in viticulture. These are bud count, type-of-bud classification, bud geometric characterization, internode length, and bud development stage. Their values at any location are of importance to agronomists for deciding on possible treatments (e.g., the application of fertilizers, canopy pruning), or for predicting plant productivity. Observation and measurement of crop variables is a fundamental task that offers the agronomist information about crop state, providing the means for informed decision-making of what treatments must be applied in order to maximize productivity and crop quality. At present, these variables are measured through direct or indirect human visual inspection, whose elevated cost often results in the measurement of only a small sample of all cases. When data are scarce, even powerful statistical techniques may still result in high uncertainty in the decision-making process, motivating the introduction of improved sensing procedures. Locating buds is a necessary task to conduct a proper measurement of the above variables. However, 2D localization is sufficient for all variables with the exception of internode length, for which 3D localization of two consecutive buds in a cane is necessary to avoid perspective errors. Still, automatic, high-throughput measurement of these variables would come with no extra cost with an autonomous 3D localization system in place.

1.1. *Related work*

There are many computational approaches to aid viticulture, including detecting grapes and bunches, estimating grape size and weight, estimating production and foliar area indexes, phenotyping, and autonomous selective pulverization [19,30,6,12,2,31]. For a more extensive review, see [37].

Specifically concerning the detection of grapevine buds, there are two recent studies (in 2D only) that address the problem of grapevine bud detection [38,12]. The first one presents a grapevine bud detection algorithm designed specifically to establish the groundwork for a future autonomous pruning system in the winter season (with no leaves left that may occlude the vision and operation of the cutting mechanism). Bud detection is performed from RGB images (the image resolution in this study is unknown). Furthermore, on top of this assumption, images are captured indoors with an industrial CCD camera with controlled background and lighting conditions. To discriminate between plant and background pixels, the authors apply a simple threshold resulting in a binary image to obtain a wire skeleton of the plant. Under the assumption that bud morphology is similar to that of the corners, they apply Harris' algorithm [9] to the skeleton image for detecting those corners. This process produces a recall of 0.702, i.e., 70.2% of buds detected. Although some improvements are suggested by the authors, the most striking limitations of this work are the need for images captured under controlled indoors conditions and the fact that the resulting localizations are in 2D. A second work for bud detection is presented by Herzog et al. [12]. This work introduces three methods of bud detection. The best results are obtained with the semi-automatic method that requires human intervention for validating the quality of the results. Detection is based on 3456 × 2304 RGB images, where the scene is altered with an artificial black background, producing a recall of 0.94. The authors argue that this recall is enough to satisfy the phenotyping of plants. However, as the authors themselves point out, these good results are mainly explained by the particular color and morphology of the buds, captured when bud sprouts are visibly green and their average size is around 2 cm (compared to a typical 5 mm diameter of a dormant bud) which makes it easier to discriminate them visually from other plant components. Although these works represent important advancements in specific bud detection applications, they suffer from some of the following limitations: (i) the use of an artificial background, (ii) controlled indoors luminosity, (iii) the need for human intervention, (iv) the detection of buds in an advanced stage of development, and (v) detection is in 2D.

Dey et al. [5] introduced a pipeline for recovering the 3D structure of the grapevine plant in the spring–summer season (i.e., with leaves and fruits) from a 3D point cloud. This 3D point cloud visually represents the surface parts of the environment, where each point is represented by a tuple containing the 3D position in world coordinates (x, y, z). Cloud reconstruction is obtained with the algorithm proposed by Snavely et al. [28]. Afterwards, the cloud is classified into leaves, branches, and fruits by means of a supervised classification algorithm that uses shape and color features. The experiments show an accuracy of 0.98 for grapes before maturation (still green) and 0.96 for fully ripe grapes (color change), where accuracy corresponds to the proportion of all observations (both grapes and background) that were correctly classified. Despite the similarities with our work, their work classifies grapes and ours classifies buds, making it hard to compare them. This is mainly due to the geometrical nature of the features they use that one would expect to work better for close-to-spherical shapes such as that of grapes, but which may work poorly for buds that present a highly irregular shape.

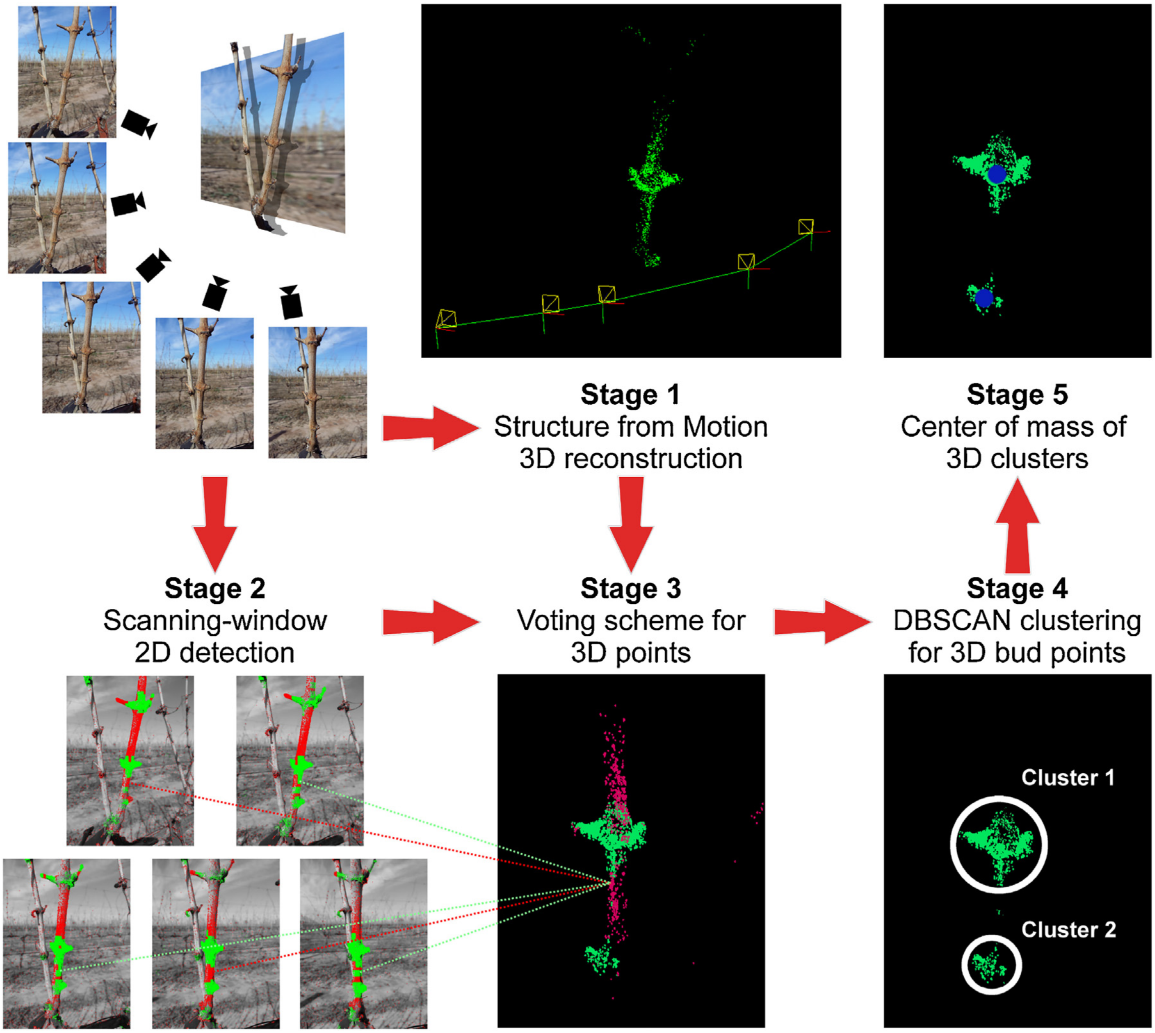

Fig. 1. Schematics of the workflow for 3D bud detection and localization. The input is a set of 2D images of some scene (upper-left). Stage 1: estimation of 3D points and camera pose (cones) for 3D scene reconstruction by *Structure from Motion*. Stage 2: scanning-window 2D detection of buds over each 2D image of the scene, showing in green those keypoints classified as bud, and in red, those classified as non-bud. Stage 3: voting scheme to produce the classification of 3D points as bud or not (green and red dots, respectively). Stage 4: spatial clustering of all 3D bud points to individualize buds, by considering different clusters as different buds (white circles). Stage 5: locates buds as the center of mass of 3D points of clusters (blue dots for each cluster). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

2. Materials and methods

In this section we provide a detailed description of our approach of 3D detection and localization of grapevine buds together with a detailed description of the input collection of images.

The detection and localization workflow consists of five stages as depicted in Fig. 1: (1) a 3D construction technique known as *Structure from Motion* [10] that, given as input a set of 2D images of some scene, produces both the 3D geometry (point cloud) of the scene and the camera pose of each 2D image; (2) a *scanning-window* technique [36] over each 2D image of the scene, used for classifying each of the image-patches corresponding to each window as either a bud or not, through the classifier presented by [20]; (3) a voting scheme for the classification of each 3D point in the cloud as being part of a bud or not, based on the number of patches and number of images in the scene that contain its projection; (4) a clustering stage for the 3D detection of buds by running the *DBSCAN* spatial clustering algorithm [7] over the 3D cloud points classified as part of a bud, with each cluster representing a detected bud; (5) localization of buds as the center of mass of the point cloud corresponding to each cluster.

The first stage consists in the use of the 3D reconstruction technique known as *Structure from Motion (SfM)* [10] that, given as input a set of 2D images of some scene, produces both the 3D geometry (point cloud) of the scene and the camera pose of each 2D image (see an illustrative result of stage 1 in Fig. 1, corresponding to an actual scene reconstruction from images in the collection). The method starts by detecting the keypoints of the 2D images using the *ORB (Oriented FAST and Rotated BRIEF)* algorithm [24]. These keypoints are then grouped in projection bundles, one per 3D point in the cloud, with each image contributing at best one keypoint to the bundle. Each of the bundle keypoints corresponds to the projection of the 3D point in its corresponding image. The trick is that it is possible to construct these projection bundles before knowing the actual location of the corresponding 3D point, by considering that keypoints are the projection of the same 3D point if they match visually. This matching is conducted by first applying the *DAISY* algorithm [32] to compute a visual feature descriptor of the local neighborhood of each keypoint, and then using the *FLANN (Fast Approximate Nearest Neighbor Search)* algorithm [18] to visually match keypoints of different images in the scene. To do this, it takes every two images of the scene and performs a symmetric distance comparison, in feature space, between the feature descriptors of their keypoints. More precisely, it considers that a keypoint k of the first image visually matches some keypoint descriptor k' in the other, if on the one hand, it holds that among all keypoints in the second image, descriptor k' is the closest to descriptor k. On the other hand, the opposite also holds, that is, if among all descriptors in the first image, descriptor k is the closest to descriptor k'. Ultimately, the goal is to use these bundles to determine not only the position of these 3D points, but also the camera pose of each image. Clearly, a single bundle is not enough, and since it provides at most one projected point per image, it is insufficient to constrain its pose. Instead, more bundles increase the constraint, as they provide more projected points per image, to eventually restrict its pose completely. In practice, the matching is noisy, and there is no analytical solution to this constraint problem, so the process proceeds through a minimization called *bundle adjustment* [33]. The bundle adjustment proceeds iteratively in an online minimization process, proposing at each step a value for the camera pose parameters as well as the coordinates of the 3D points and computes as cost function the so called *reprojection error*. This is computed as follows: (i) first it uses the camera poses to project each 3D point into each 2D image; (ii) then it computes the squared distance between each keypoint in the image to its corresponding projected position; and (iii) it sums these squared distance over all keypoints of all 2D images and reports its squared root, resulting in a quantity measured in pixel units. The implementation of SfM used in this work is that provided by the OpenCV 3.2.0 open source library [3], which implements the SfM version[1] of Hartley and Zisserman [10] described in this section. It also uses the third-party library *Ceres-Solver (A Nonlinear Least Squares Minimizer)* [1] for the bundle adjustment minimization process.

The second stage of the proposed workflow runs a scanning-window 2D detection technique [36] over each 2D image of the scene. This technique proceeds by sliding a fixed size window over the whole image, at fixed size steps with some overlap, and by classifying each image patch covered by each window either containing a bud or not. The classification is performed using the classifier proposed by [20]. The results are patches with known geometry and localization in the image, classified either containing a bud or not. Results of this stage are shown in stage 2 of Fig. 1, with keypoints belonging to patches classified as bud depicted in green (light gray) points, and those belonging to non-bud patches depicted in red (dark gray). The classifier of Perez et al. proceeds in a workflow of computer vision and machine learning sub-processes: (i) First, it runs *Scale-Invariant Features Transform (SIFT)* [17] for computing the low-level visual features of the keypoints of each patch; (ii) it then runs *Bag of Features (BoF)* [4] for constructing a higher level descriptor of the patch, based on patch keypoints and their SIFT descriptors; and (iii) it concludes by running a *Support Vectors Machine* [34] modeler for training a binary classifier based on a collection of labeled patches represented by their BoF descriptors. It is important to note that in this work, we reproduced the same classifier of Perez et al. by training with the parameters provided in their work and the training collection made publicly available,[2] leaving only the choice of scanning-windows parameters, i.e., window size and step. At first glance, it would seem that in order to obtain a good classification, one should choose a window and step sizes so that each bud in the image is perfectly circumscribed by some patch. This is clearly not only impossible to perform for all buds and images for fixed window and step sizes of the training collection—as buds are variable in size—but it is also impossible for a testing collection, since here bud sizes and positions would be unknown. However, together with the classifier, Perez et al. provide a robustness analysis for window geometry showing that the classifier is robust to patches that have lost up to 40% of the bud's pixels (i.e., at least 60% of the bud's pixels are visible), and it contains non-bud visual information covering up to 80% of the patch (i.e., bud pixels cover at least 20% of the patch). Based on these numbers and an approximate bud diameter of 150 pixels obtained from an inspection of our collection of 2D images (see below for more details of this collection), we chose a window size of 150 × 150 pixels and a step of 75 pixels. This guarantees a 50% overlap between contiguous patches, considering that these values should produce bud coverage within the accepted values of the robustness analysis.

The third stage of the workflow combines the results of the first two stages: the 3D position of keypoints and classification of patches to produce the classification of these 3D points as part of a bud or not. The 3D classification proceeds through a voting scheme for each 3D point that classifies it as being part of a bud whenever the number of images in which it has been detected surpasses a threshold τ_I. Here, a 3D point is considered as detected in some 2D image whenever the keypoint in the projected bundle of this 3D point corresponding to that image falls within a minimum number τ_P of bud patches of that image (see Fig. 1). The basic rationale behind this voting scheme is the intuition that only true bud visual aspects will show in all images, whereas noisy detections would show them in only one of the images and cancel them out by the voting filter as long as it is kept in low levels. As with previous stages, this process is illustrated in Fig. 1, showing five lines going from one keypoint in each 2D image in stage 2 to one 3D point in the reconstructed scene of stage 3. The keypoints at the point of origin of these 5 lines correspond to a bundle, with 3 (2) of them classified as bud (no-bud), so both the keypoint and its line were colored green (red), or light (dark) gray for grayscale versions of the image. As seen in the image, the 3D point is colored red (dark gray), corresponding to no-bud, a result of the voting scheme for threshold $\tau_I = 4$ or $\tau_I = 5$.

At this point we have a 3D point cloud, with each point in the cloud classified as being part of a bud or not. This however does not individualize buds, nor does it provide a localization for them (a process conducted in the last two stages of our workflow) also depicted in Fig. 1. To do this, the workflow continues with stage 4 that executes the *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)* [7] to spatially cluster the 3D bud points, considering different clusters as different buds. This algorithm works under the fundamental assumption that points located in dense regions belong to the same cluster, thus searching for high density regions separated by low density regions. An important property of this algorithm is that it requires no predetermination of the number of clusters, a property necessary to automatize detection in scenes with an a priori unknown number of buds. It is also designed to discover arbitrary-shaped clusters and is robust to noisy points excluding them from any cluster. The key idea of the cluster recognition process is to detect high density regions by requiring for each point of a cluster that the region of radius r around it contain at least m other points belonging to the same cluster. The two parameters r and m are user-specified and may drastically affect the outcome of this stage (as shown later in the results section 3). To conclude we have to deal with a rather technical issue, necessary for a proper reproducibility of our workflow. Scene reconstruction by the SfM method may result in rather arbitrary scales, with differences of orders of magnitude, resulting in parameter values r which greatly affect the DBSCAN process. To give a sense of this variation, we computed for each scene the *mean minimum distance (MMD)* that reports the mean value of the distance of each 3D point in the cloud of that scene to its closest 3D point in the same scene. Fig. 2 shows a histogram for MMD over the 47 scenes, in log scale, showing a variation range of over 15 orders of magnitude. To address this dispersion, we re-scaled the radius parameter r multiplying it by the MMD of the scene before passing it to DBSCAN.

The workflow then ends with a fifth and final stage that locates buds in the centers of mass of the 3D points of its cluster.

The final outcome of the workflow just described is bud clusters in 3D together with their respective centers of mass. An ideal correct outcome would, therefore, consist of a number of clusters matching exactly the number of buds in the scene, with their centers of mass coincident with the center of mass of the buds. Instead, wrong outcomes would consist of mislocated clusters, worse, spurious clusters, that

[1] http://docs.opencv.org/trunk/d4/d18/tutorial_sfm_scene_reconstruction.html.

[2] Available in http://dharma.frm.utn.edu.ar/vise/bc/.

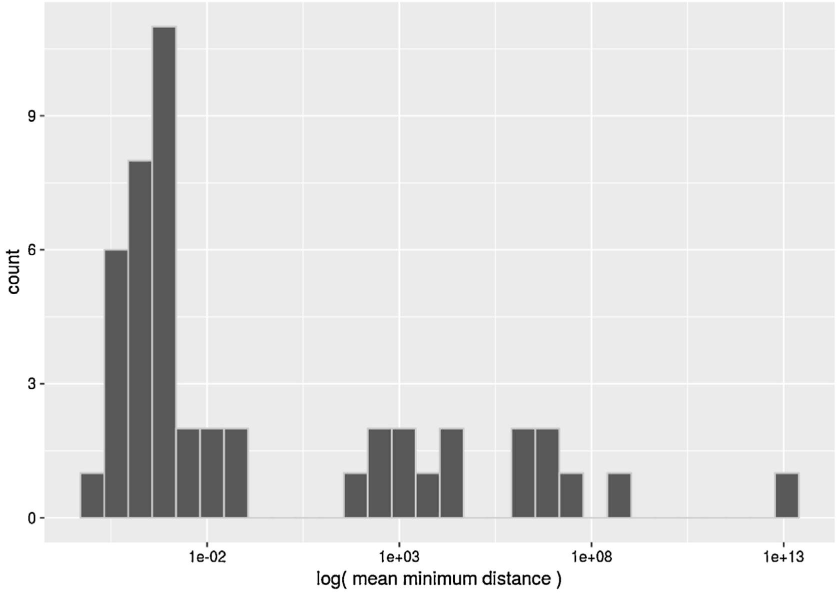

Fig. 2. Histogram of the *mean minimum distance (MMD)* over the 47 scenes of the corpus, with the *X*-axis shown in log scale. The histogram shows the enormous dispersion in MMD, with cases ranging over 15 orders of magnitude.

correspond to no actual bud of the scene or buds that have no cluster representing them. In the next subsection we describe in detail the collection of 47 scenes used in the evaluation described in the following section. It first introduces formally some performance measures that quantify these different aspects of the quality of the 3D bud detection workflow. Then, it reports their values for a representative spectrum of values for the four user-defined parameters that control these outcomes (i.e., image-voting threshold τ_I, patch-voting threshold τ_P, DBSCAN radius r, DBSCAN minPts m).

2.1. Collection of scenes and their 2D images

We captured a collection of images that satisfy the requirements of this work: they were taken in the winter season using RGB mobile phone cameras in natural field conditions. In addition, there are specific requirements for capturing 2D images imposed by the third-party modules of the proposed workflow: the SfM module of OpenCV 3.2.0 for 3D reconstruction of grapevine branches and the 2D detection algorithm based on the approach of Perez et al. [20]. Firstly, the documentation of the SfM algorithm[3] recommends in the order of 3–5 images for a proper reconstruction, captured from differing points of view, but as close as possible to one another. In addition, the elements of the scene (i.e., branches) need to be well focused, and exposition levels kept within reasonable values. Secondly, the scanning windows algorithm and the bud classifier used within require buds of at least 100 pixels to maintain the robustness of classification results, as recommended by the authors. This resulted in the following image captured:

1. with a Samsung Galaxy A5 mobile phone camera, without flash, in JPEG format, and a resolution of 4128 × 3096 pixels;
2. satisfying the focus and exposition level requirements of the SfM modules as detailed above, with 5 images per scene;
3. positioning the camera over an imaginary circular path around the branch, at approximately equal displacements between them, with an overlap above 80%, and always pointing toward the branch, conditions that guarantee a good reconstruction;
4. at a distance of 12 cm from the branches to guarantee that buds are at least 100 pixels in diameter for the chosen resolution;
5. on sunny days, under normal field conditions, without altering the scene with artificial elements, and maintaining natural lighting conditions;
6. between 15:00 and 17:00 h in late August (winter in the southern hemisphere), when leaves are either dry or have fallen, but before sprouting again (see Fig. 3).

We captured 60 scenes for a total of 300 2D images, corresponding to branch parts of a single grapevine plant (as exemplified by the 5 images of Fig. 3). It is worth mentioning that our workflow omits any automation for the selection of input images in order to guarantee the success of the 3D reconstruction. Therefore, from a total of 60 scenes, 10 were manually discarded for not following the focus and exposition quality requirements of the SfM module. After the SfM reconstruction, 3 more were discarded due to failure in reconstruction (detected by reprojection errors of 60 pixels or more). After this manual pruning, the collection was left with 235 images corresponding to the 47 remaining scenes, with mean and standard deviations of the reprojection error of 2.91 and 5.41 pixels, respectively. Among these scenes we counted a total of 106 buds, with an average of 2.25 buds per scene.

We ran the 2D bud classification over this image collection to assess the merit of the 2D bud classifier of [20] for stage 2, when pre-trained over the original image collection. To assess classifier recall, i.e., the proportion of true buds it could detect, we considered two different collections of patches representing true buds. The first was a collection of perfectly-circumscribed patches extracted from rectangles that perfectly circumscribe each bud in each image collection. Second, we ran a scanning-window of 150 × 150 pixels and a step of 75 pixels and collected all patches that overlapped a bud on at least one pixel. We also assessed the precision classifier, i.e., the proportion of detected buds that were indeed true buds. To do this, we considered the same scanning-window, but this time collected the complement set, i.e., all patches that did not contain a single bud pixel. After running the classifier over all these image patches, we obtained a recall of 0.978 for the perfectly-circumscribed patches, a recall of 0.0596 for the single pixel overlapping cases, and a precision of 0.0511 for the non-overlapping patches. The latter is a result of the fact that from all ≈559*K* patches of the scanning-window containing no buds, 15,756 were incorrectly classified as buds, i.e., were *false positives*, drastically reducing the proportion of *true positives* over all those classified as buds.

[3] http://docs.opencv.org/trunk/da/db5/group_reconstruction.html.

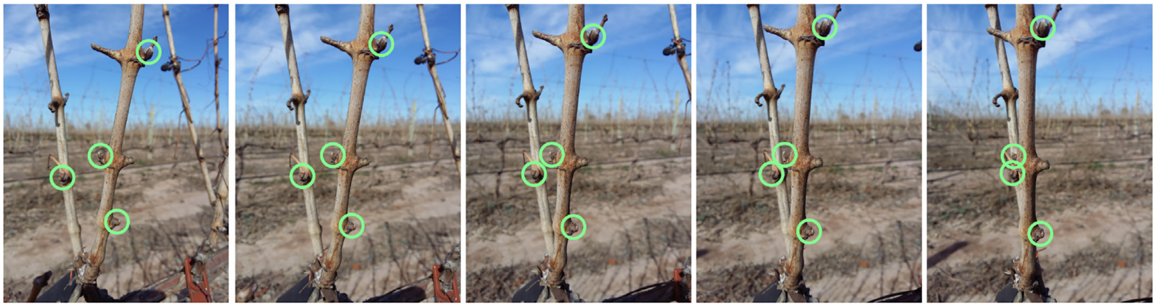

Fig. 3. Example of the images of one scene of the corpus, with circles marking the bud location.

3. Experiments

In this section we present results of systematic experiments that evaluate the quality of the 3D structures produced by our approach. We first introduce quantitative performance measures that assess *detection* and *localization errors* that report *hard* errors of true buds that were undetected, or clusters that detected no bud, and *soft* errors reporting how far the correctly detected buds fell from the actual position of the buds they detected. Values for these performance measures are reported systematically for a representative range of values of user-input parameters, the two thresholds τ_I and τ_P of the voting scheme (stage 4), the radius r, and minimum number of points m of the DBSCAN clustering algorithm.

3.1. Performance measures

Now, let us explain the details of the *detection* and *localization* errors.

Detection error: This measure represents the *hard* errors of true buds that were undetected or clusters that detected no bud, reported by the well-known *precision* and recall measures, respectively. These are formally defined as recall = $\frac{\text{TP}}{\text{TP + FN}}$ and precision = $\frac{\text{TP}}{\text{TP + FP}}$, with *TP*, *FP*, and *FN* denoting *true positives*, *false positives*, and *false negatives*, respectively [22]. These quantities contrast the results of our 3D detection workflow with the ground truth obtained from manual detection of buds, corresponding to the center of mass of the perfect circumscription rectangles described in the collection section above (cf. Section 2.1).

Specifically in this work, we consider that a bud has been correctly detected—that is, it is a TP—whenever it satisfies *symmetrical closeness* to some cluster—i.e., this bud is the closest bud to its closest cluster—with closeness being measured in Euclidean distance in pixels. This definition of TPs could result in clusters far away from a bud being counted as its TP, as long as they satisfy symmetrical closeness. In practice, however, our results show this is not the case, as worst localization errors are around 600 pixels. Additionally we consider that a bud has been missed—that is, it is a FN—when its closest cluster is itself closer to some other bud, and that a cluster detects no bud—that is, it is a FP—when it is not the closer cluster to its closest bud. The definitions of these quantities are illustrated in Fig. 4. Dotted rectangles *A* and *B* mark buds manually circumscribed with their center of mass marked as a dot within it. The blue (dark) dots 1, 2, and 3 within the dotted circles mark the projection of the center of mass of three detected bud clusters. Since cluster 1 is the closest to bud *B*, and at the same time, bud *B* is the closest bud to cluster 1, then, cluster 1 is the TP of bud *B*. In addition, even though clusters 2 and 3 have bud *B* as the closest one, they are themselves not the closest to *B* (cluster 1 is), so they are FPs. Finally, bud *A* is a false negative as none of the clusters has this bud as its closest.

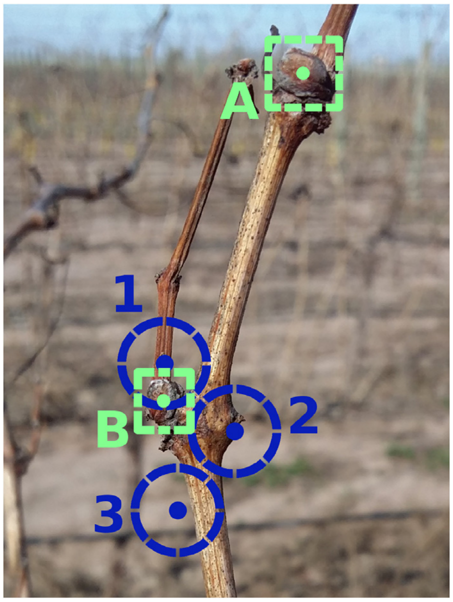

Fig. 4. The figure illustrates the definitions of *true positives* (TP), *false positives* (FP), and *false negatives* (FN). Dotted rectangles *A* and *B* mark buds manually circumscribed with their centers of mass marked as a dot within it. The blue (dark) dots 1, 2, and 3 within the dotted circles mark the projection of the center of mass of three detected bud clusters whose position has been selected manually for illustration purposes. Since cluster 1 is the closest to bud *B*, and at the same time, bud *B* is the closest bud to cluster 1, then cluster 1 is the TP of bud *B*. Even though clusters 2 and 3 have bud *B* as the closest one, they are themselves not the closest to *B* (cluster 1 is), so they are FPs. Finally, bud *A* is a FN as none of the clusters has this bud as its closest. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

Localization error: Detection error measured by precision and recall. It is an important measure of quality, but it may miss the *soft localization errors* that zoom into the detected buds represented by true positives and report how far their detection has fallen from their true position. Formally, we report as *localization error* the mean of the individual localization error of all buds, with the individual localization error computed as the distance between the center of mass of the circumscribed rectangle of the bud and the center of mass of its symmetrically closest cluster.

The computation of precision, recall, and localization error require the 3D coordinate of each bud's center of mass. In practice, this demands measuring the 3D localization of each bud over a common coordinate system for all of them, an extremely complex task to be performed manually, so the alternative of measuring ground-truth 3D localizations for our collection was discarded as an option. We considered instead an *approximated* alternative for measuring these errors, one that computes them in the 2D pixel space of each image. Therefore, instead of considering the 3D localizations of both clusters' center of mass and bud's center of mass, it considers their *reprojected* localizations over each individual image, i.e., their coordinates in the 2D pixel space of each image corresponding to their position in the field of view of the camera corresponding to that image. The computation of these reprojected localizations can be easily automated. Once computed, the

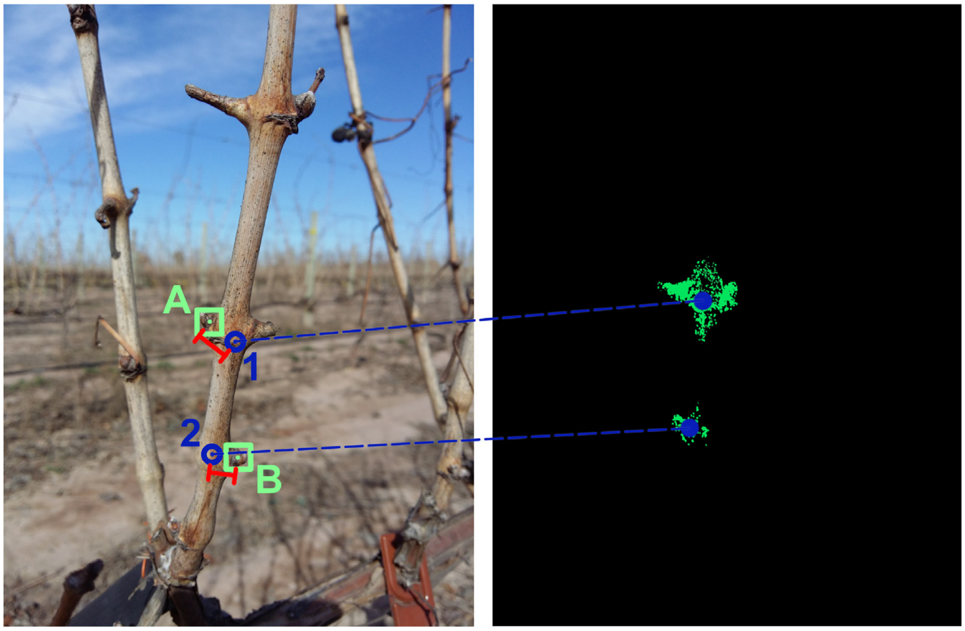

Fig. 5. This figure shows the reprojection into 2D of a 3D bud detection, together with its 2D localization error, computed as the reprojection error. In the figure, the light green squares *A* and *B* (or light-gray in gray-scale version) correspond to the actual localization of the two buds, whereas the blue circles 1 and 2 (dark gray in gray-scale version) represent the reprojected center of mass. The 2D localization error of each bud is represented by the length of red line segments 1*A* and 2*B* (dark gray in gray-scale version). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

computation of precision, recall and localization errors followed exactly their 3D definition, but over 2D localizations, replacing 3D Euclidean distance with 2D Euclidean distance in pixels. Fig. 5 illustrates this approximation with the image on the right showing two clusters of the 3D point geometry of a branch, with their centers of mass reprojected into one of the 2D images of the scene. The 2D localization errors are shown in red line segments.

Now, we proceed to discuss the results obtained from the systematic experiments.

3.2. Systematic results

Fig. 6 reports *precision* and *recall* detection errors as well as the localization error (in pixels) for all assignments obtained from the following values of the four free parameters $\tau_I \in \{1, 2, 3, 4, 5\}$, $\tau_P \in \{1, 2, 3, 4\}$, $r \in \{0.01, 0.05, 0.10, 0.50, 1, 2, 3, 5, 10, 50, 100\}$, and $m \in \{1, 3, 5, 10, 25, 50, 100, 200\}$ where τ_I and τ_P are the image and patch voting thresholds, respectively, and *r* and *m* are the DBSCAN radius and minPts, respectively. This figure shows a scatter plot of recall versus precision with a gray-scale color coding denoting the localization error. In this plot, darker colored dots represent assignments of the four free parameters with a lower localization error, with the best possible outcome for the detection error corresponding to both recall and precision equal to 1, located in the top-right corner at coordinates (1, 1). Results in the plot show an abrupt fall of recall for small precisions, next, a rather constant recall after a precision of 0.2, and finally, for a large precision, a fall in recall to its lowest value of *recall* = 0.2 for *precision* = 1. The worst localization errors of approximately 600 pixels

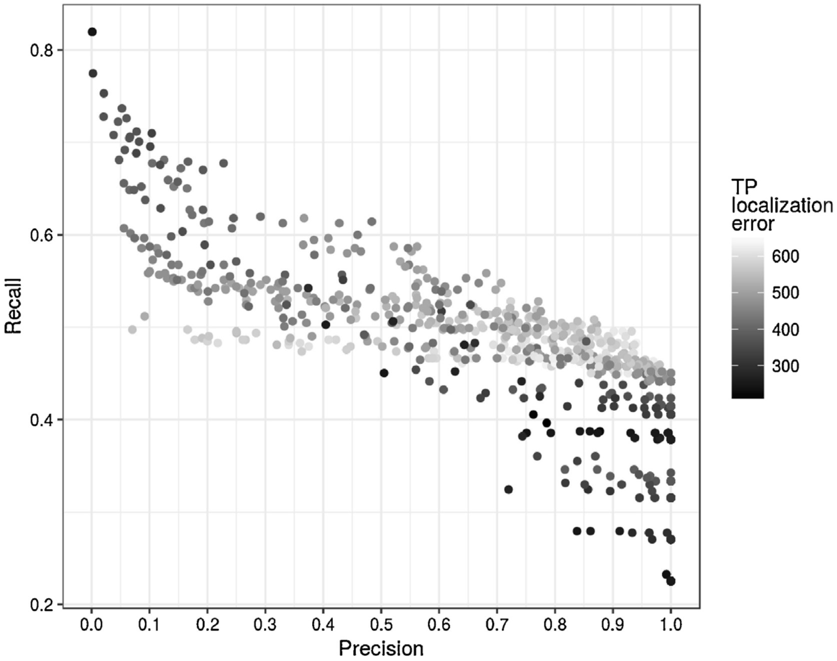

Fig. 6. The figure shows recall versus precision detection errors for all assignments of the free parameters τ_I, τ_P, *r*, *m*, with a gray-scale color coding denoting the localization error in pixels(with darker color for lower errors).

Table 1
A summary of best results with the top (bottom) 5 rows showing best results in terms of precision (recall). The values with the best precision (recall) are marked in bold. The column "#(Assignments)" corresponds to the number of different value assignments for the four free parameters that produced the precision and recall results of the first two columns. The table is completed with the mean and standard deviation of the true positive localization errors over these assignments and the mean of each of the four parameters over their values for each of these assignments.

Precision	Recall	#(Assignments)	Localization error of TPs	τ_I	τ_P	r	m
1	0.45	25	554.87 (34.7)	3.08	2.52	1.83	120.60
1	0.441	47	462.73 (21.98)	3.53	2.40	3.48	94.26
1	0.423	2	371.96 (2.45)	4.00	2.00	0.75	7.50
1	0.414	27	367.96 (0.0)	4.00	2.00	6.77	98.70
1	0.405	35	330.90 (0.00)	4.00	3.00	5.80	83.97
0.001	**0.82**	1	247.5	1.00	1.00	10.00	1.00
0.001	**0.82**	1	244.21	1.00	1.00	5.00	1.00
0.002	**0.775**	1	305.98	1.00	1.00	50.00	1.00
0.021	**0.753**	1	348.84	1.00	1.00	50.00	3.00
0.052	**0.737**	1	374.70	1.00	1.00	50.00	5.00

(light-gray) are concentrated at mid-range recalls of around 0.5 and decrease for either large and small recall values. As extreme assignments for the detection error, we have the upper-left case of *recall* = 0.85 and *precision* ≊ 0, meaning that although most buds have been detected (85% more precisely), an extremely large number of buds has been falsely detected. On the other end, we have the dark dots in the lower right sector corresponding to *recall* = 0.2 and *precision* = 1. This case corresponds to assignments of the free parameters that incorrectly miss 80% of the buds, but on the other hand, not a single detected bud is wrong. More details of extreme assignments are shown in Table 1. Although there are no assignments close to optimal values of (1, 1), it is worth highlighting that for a precision of exactly 1, recall values range between 0.22 and 0.45.

The data plotted in Fig. 7 is the precision and recall over all assignments of the four free parameters showing two box-plots, one for precision (in light-gray) and one for recall (in dark-gray) with boxes grouping all assignments of each image voting threshold, regardless of the value of the other parameters. The figure shows a clear trend for both precision and recall, with the distribution of precision assignments leaning toward the upper values for larger thresholds, concentrating on 90% for $\tau_I = 4$, and on 100% for $\tau_I = 5$. In contrast, recall distribution moves toward lower values for large thresholds, concentrating at 50% for $\tau_I = 1$ and decreasing down to 30% for $\tau_I = 5$.

4. Discussion

From Fig. 6 we considered as best outcomes those located at precision = 1 (i.e., all detections correspond to actual buds) and recall in a range from 0.38 to 0.45 (i.e., between 38% and 45% of buds detected). These assignments show localization errors in the range of 259–554 pixels, which correspond to approximately 3 buds and approximately 1.5 cm. This is because, for the image scale in the collection, average bud diameter is 159 pixels with 95% of the total probability mass falling within the range of [80,263] pixels. In the grapevine variety of our study, average bud diameter is approximately 5 mm.

We consider high precision at the expense of lower recall because we regard these to be best for the central application of our work: estimation of future shading (canopy) structure through simulations. As mentioned in the introduction, these simulation techniques take as input different numerical parameters of plant architecture including, in particular, the localization of its buds. Since in practice, it is an extremely difficult task to measure even the 3D localization of a few buds, these simulations contemplate the possibility of localizing missing buds—even all 100% of them—through stochastic procedures. In other words, they contemplate low recall values, even 0%. Furthermore, these methods may not easily tolerate the input of badly localized buds, or even worse, buds located where in practice there is none, as it would be the case of falsely detected buds. In those cases—equivalent to low precision—the simulated structure may end up with false shoots, bunches, fruits and leaves. These results, however, still present important limitations. First, the sampling of these 45% of buds cannot be controlled or designed, but is rather biased by unknown visual characteristics of the undetected buds. In addition, the workflow as presented here still depends on manual capturing of a handful of images for tens of scenes per plant, a clear bottleneck for high throughput. A fully automated workflow would require: (i) recording all reconstructed scenes in a common coordinate system, currently reconstructed into completely independent coordinate systems; (ii) automatic pre-selection of images, e.g., focused, valid exposures; (iii) validation of correct 3D scene reconstructions, e.g., those with low reprojection errors; and (iv) autonomous planning and positioning of an autonomous capturing device (e.g., drone) for producing valid image collections for each reconstruction.

While these issues render the current approach still unpractical for satisfying all the requirements of the measuring process of the variables of interest, these limitations may still be overcome by future research. Indeed, these results are strong enough to motivate further research on the possibilities of computer vision and machine learning for spatial modeling of vines. We conclude with some more detail on the limitations of the two motivating applications:

- **Optimal pruning design**: Despite all the limitations, our work provides agronomists with novel information on bud localization that is currently almost impossible to measure. As already mentioned, this information, together with a model of the plant's architecture, can become input for backward and forward simulators to improve the studies on optimal pruning procedures. Currently, those simulators only use the plant's architecture, since bud localization is unavailable, while with our work they can locate 45% of them with a maximum displacement of 1.5 cm. Subjective assessments indicate that these localization errors should not have a major impact on the shading structures simulated, an assessment that can only be rendered conclusive once actual simulations are performed.
- **Internode length**: This variable reports the distance between two consecutive nodes of the same branch. However, since buds always grow over nodes, the distance of consecutive buds over the branch are a very close approximation of internode length. On the one hand, bud localization alone is insufficient, as there is no information on whether those buds belong or not to the same branch. On the other hand, integration with plant architecture reconstruction techniques can easily overcome this limitation. However, a 45% recall presents a more difficult challenge. This recall is still too low for guaranteeing that two detected buds are indeed nearest neighbors over the cane. With larger recalls, statistics may be of help by reducing the probability that there is still an intermediate bud between any two detected buds.

The trend of precision boxes Fig. 7 highlights a positive feature of the workflow's voting step: a drastic improvement in precision from 2D to 3D. As already discussed above in Section 2.1, the 2D classification resulted in a precision of 0.0511 corresponding to 15,756 non-bud patches falsely classified as bud patches. Interestingly, the precision 1.0 for a voting threshold of 5 implies that none of these 2D patches contributed to a 3D bud cluster. This is explained by two facts: first, that larger voting thresholds require that more 2D images agree on their classification of a patch for it to contribute with its keypoints in the 3D cloud. Second, this helps clean up the noise by our intuition that only

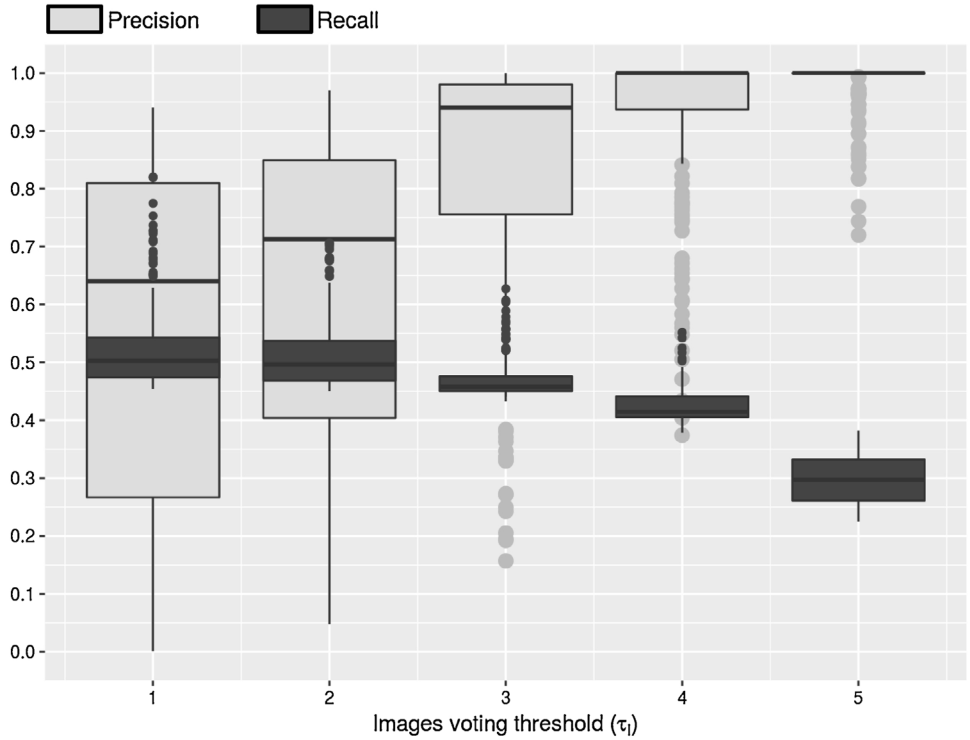

Fig. 7. Trends for precision and recall. The light-gray boxes show precision and dark-gray boxes recall, with boxes grouping all assignments of the four free parameters of each voting threshold τ_I. The data plotted in the figure is precision and recall over all parameter assignments.

true bud visual aspects will show in all images, while noisy aspects will tend to show in only few images.

5. Conclusions

In this work we introduce a workflow for the localization of grapevine buds in 3D space obtained from plant parts 3D models reconstructed from multiple 2D images, captured during the winter season, using RGB mobile phone cameras in natural field conditions. The proposed workflow is based on well-known computer vision and machine learning algorithms, such as SfM, SIFT, BoF, SVM, DAISY, ORB and DBSCAN. We justified the importance of bud 3D detection through their potential applications, such as prolonged sunlight exposure, autonomous pruning systems, and internode length. When assessed over a representative range of values of user-input parameters, the best outcome obtained was a precision of 1 and a recall in the range of 0.38–0.45 with a localization error in the range of 259–554 pixels equivalent to approximately 3 buds. These results represent an important impact of our approach to the problem of designing optimal pruning procedures with measurement of bud sunlight exposure and autonomous pruning as two relevant and challenging sub-problems. Our approach has the potential of providing novel information for producing both backward (previous Spring) and forward (following Spring) simulated shading structures paramount for estimating sunlight exposure of buds, and with it, the potential productivity of the pruning procedure. There are several automation steps still missing, however, which are all addressable by future work: registering of all the scenes in a common coordinate system, automatic pre-selection of images, autonomous detection of valid scene reconstructions (e.g., low reprojection errors), and autonomous positioning and posing of the capturing device. Finally, further research is required for improving recall, for instance, exploring novel reconstruction techniques and novel means for aggregating 2D patch classification into a detection algorithm. One could also consider integrating information from other parts of the plant, for instance, following the information provided by Xu et al. [38]. As discussed in Section 1.1, their work uses only information about plant architecture to position buds. This information is independent of that used by the workflow of our work, suggesting interesting possible integrations.

Acknowledgments

This work was funded by the National Technological University (UTN), the National Council of Scientific and Technical Research (CONICET), Argentina, and the National Fund for Scientific and Technological Promotion (FONCyT), Argentina. We thank the National Agricultural Technology Institute (INTA) for offering their vineyards to capture the images used in this work.

References

[1] S. Agarwal, K. Mierle, Ceres Solver, (2012) http://ceres-solver.org.
[2] R. Berenstein, O.B. Shahar, A. Shapiro, Y. Edan, Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer, Intell. Serv. Robot. 3 (4) (2010) 233–243.
[3] G. Bradski, The OpenCV library, Dr. Dobb's J.: Softw. Tools Prof. Program. 25 (11) (2000) 120–123.
[4] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, Workshop on Statistical Learning in Computer Vision, ECCV, vol. 1, Prague, 2004, pp. 1–2.
[5] D. Dey, L. Mummert, R. Sukthankar, Classification of plant structures from uncalibrated image sequences, 2012 IEEE Workshop on Applications of Computer Vision (WACV), IEEE, 2012, pp. 329–336.
[6] M.-P. Diago, C. Correa, B. Millán, P. Barreiro, C. Valero, J. Tardaguila, Grapevine yield and leaf area estimation using supervised classification methodology on RGB images taken under field conditions, Sensors 12 (12) (2012) 16988–17006.
[7] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, Kdd, vol. 96 (1996) 226–231.
[8] P. Fu, P.M. Rich, A geometric solar radiation model with applications in agriculture and forestry, Comput. Electron. Agric. 37 (1) (2002) 25–35.
[9] C. Harris, M. Stephens, A combined corner and edge detector, Alvey Vision Conference, vol. 15, Manchester, UK, 1988, pp. 10–5244.
[10] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge Books Online, Cambridge University Press, 2003.
[11] E.W. Hellman, Grapevine structure and function, in: E.W. Hellman (Ed.), Oregon

Viticulture, Oregon State University, Corvallis, 2003, pp. 5–19.

[12] K. Herzog, et al., Initial steps for high-throughput phenotyping in vineyards, Australian and New Zealand Grapegrower and Winemaker (603), (2014), p. 54.

[13] A. Iandolino, R. Pearcy, L. Williams, Simulating three-dimensional grapevine canopies and modelling their light interception characteristics, Aust. J. Grape Wine Res. 19 (3) (2013) 388–400.

[14] M. Keller, The Science of Grapevines: Anatomy and Physiology, Academic Press, 2015.

[15] S. Khanduja, V. Balasubrahmanyam, Fruitfulness of grape vine buds, Econ. Bot. 26 (3) (1972) 280–294.

[16] G. Louarn, J. Lecoeur, E. Lebon, A three-dimensional statistical reconstruction model of grapevine (*Vitis vinifera*) simulating canopy structure variability within and between cultivar/training system pairs, Ann. Bot. 101 (8) (2008) 1167–1184.

[17] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[18] M. Muja, D.G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, International Conference on Computer Vision Theory and Application (VISSAPP'09), INSTICC Press, 2009, pp. 331–340.

[19] S. Nuske, S. Achar, T. Bates, S. Narasimhan, S. Singh, Yield estimation in vineyards by visual grape detection, 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2011, pp. 2352–2358.

[20] D.S. Pérez, F. Bromberg, C.A. Díaz, Image classification for detection of winter grapevine buds in natural conditions using scale-invariant features transform, bag of features and support vector machines, Comput. Electron. Agric. 135 (2017) 81–95.

[21] J. Perez, W.M. Kliewer, Effect of shading on bud necrosis and bud fruitfulness of Thompson seedless grapevines, Am. J. Enol. Vitic. 41 (2) (1990) 168–175.

[22] D.M. Powers, Evaluation: From Precision, Recall and *f*-Measure to ROC, Informedness, Markedness and Correlation, (2011).

[23] A.G. Reynolds, J.E.V. Heuvel, Influence of grapevine training systems on vine growth and fruit composition: a review, Am. J. Enol. Vitic. 60 (3) (2009) 251–268.

[24] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to sift or surf, 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 2564–2571.

[25] L.A. Sánchez, N.K. Dokoozlian, Bud microclimate and fruitfulness in *Vitis vinifera* L, Am. J. Enol. Vitic. 56 (4) (2005) 319–329.

[26] F. Schöler, V. Steinhage, Towards an automated 3D reconstruction of plant architecture, International Symposium on Applications of Graph Transformations with Industrial Relevance, Springer, 2011, pp. 51–64.

[27] P. Skinkis, A.J. Vance, et al., Understanding Vine Balance: An Important Concept in Vineyard Management, (2013).

[28] N. Snavely, S.M. Seitz, R. Szeliski, Modeling the world from internet photo collections, Int. J. Comput. Vis. 80 (2) (2008) 189–210.

[29] M. Šúri, J. Hofierka, A new GIS-based solar radiation model and its application to photovoltaic assessments, Trans. GIS 8 (2) (2004) 175–190.

[30] J. Tardaguila, M. Diago, J. Blasco, B. Millán, S. Cubero, O. García-Navarrete, N. Aleixos, Automatic estimation of the size and weight of grapevine berries by image analysis, International Conference of Agricultural Engineering, Valencia, Spain, 2012, pp. 8–12.

[31] J. Tardaguila, M. Diago, B. Millan, J. Blasco, S. Cubero, N. Aleixos, Applications of computer vision techniques in viticulture to assess canopy features, cluster morphology and berry size, I International Workshop on Vineyard Mechanization and Grape and Wine Quality, vol. 978 (2012) 77–84.

[32] E. Tola, V. Lepetit, P. Fua, Daisy: an efficient dense descriptor applied to wide-baseline stereo, IEEE Trans. Pattern Anal. Mach. Intell. 32 (5) (2010) 815–830.

[33] B. Triggs, P.F. McLauchlan, R.I. Hartley, A.W. Fitzgibbon, Bundle adjustment – a modern synthesis, International Workshop on Vision Algorithms, Springer, 1999, pp. 298–372.

[34] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 2013.

[35] M.C. Vasconcelos, M. Greven, C.S. Winefield, M.C. Trought, V. Raw, The flowering process of *Vitis vinifera*: a review, Am. J. Enol. Vitic. 60 (4) (2009) 411–434.

[36] X. Wang, T.X. Han, S. Yan, An HOG-LBP human detector with partial occlusion handling, 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 32–39.

[37] J. Whalley, S. Shanmuganathan, Applications of Image Processing in Viticulture: A Review, (2013).

[38] S. Xu, Y. Xun, T. Jia, Q. Yang, Detection method for the buds on winter vines based on computer vision, 2014 Seventh International Symposium on Computational Intelligence and Design (ISCID), vol. 2, IEEE, 2014, pp. 44–48.