Can You Pick a Broccoli? 3D-Vision Based Detection and Localisation of Broccoli Heads in the Field

Keerthy Kusumam Tomáš Krajník Simon Pearson Grzegorz Cielniak Tom Duckett

***Abstract*— This paper presents a 3D vision system for robotic harvesting of broccoli using low-cost RGB-D sensors. The presented method addresses the tasks of detecting mature broccoli heads in the field and providing their 3D locations relative to the vehicle. The paper evaluates different 3D features, machine learning and temporal filtering methods for detection of broccoli heads. Our experiments show that a combination of Viewpoint Feature Histograms, Support Vector Machine classifier and a temporal filter to track the detected heads results in a system that detects broccoli heads with 95.2% precision. We also show that the temporal filtering can be used to generate a 3D map of the broccoli head positions in the field.**

***Index Terms*— robotic vision, RGB-D sensing, field robotics, automated harvesting**

I. Introduction

Sustainable intensification of agriculture can be achieved through various technological innovations such as automated harvesting. Automated harvesting approaches bring benefits of reduced labour costs, economic sustainability, higher productivity, less waste and better use of natural resources. Selective harvesting methods choose only mature crops for harvesting, as opposed to "slaughter harvesting" where an entire field is harvested in a single pass. Broccoli is an instance of the crops that demand selective harvesting since the flowers exhibit a high variation in maturity levels, even when grown in the same field. To address these challenges, an automated selective harvesting machine would require an intelligent vision sensing unit that can detect and locate the harvestable broccoli heads. However, such systems encounter several difficulties arising from the natural variations, partial views of the heads and occlusions due to leaves and weeds.

The main objective of this paper is to investigate the feasibility of using low-cost consumer 3D cameras to identify mature broccoli heads in real, unstructured outdoor field conditions, providing the locations of the detected heads in 3D image coordinates. The presented method applies state-of-the art 3D feature extraction methods, machine learning, and temporal filtering to remove false positives and track the detected heads. Future work will address the problems of measuring the size of the detected broccoli heads to determine when a head is ready for harvest and the development of a cutting mechanism to physically harvest the crop.

The paper evaluates different 3D features, machine learning and temporal filtering methods for detection of mature broccoli heads. We show that a combination of Viewpoint Feature Histogram (VFH) and Support Vector Machine (SVM) allows to detect the broccoli heads with 94.7% precision. Moreover, we demonstrate that the integration of detection results across multiple frames (temporal filtering) allows to prune false positive detections, further improving the precision to 95.2%. We also demonstrate that the temporal filtering can be used to generate a 3D map of the broccoli head positions in the field.

All authors are with the University of Lincoln, UK. The work was funded by BBSRC and Innovate UK, project BB/N004841/1. Many thanks to Adam Turner for all his help with ground truthing the datasets used in this paper.

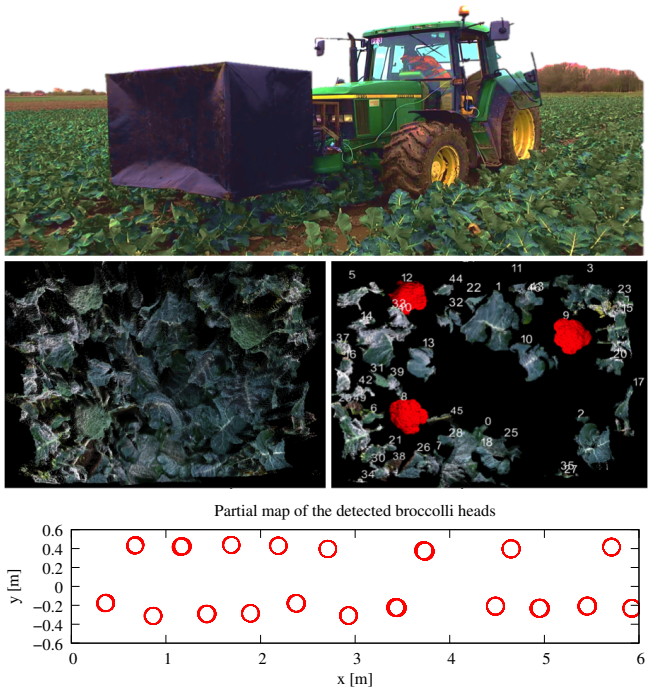


Fig. 1: System overview. Top: Tractor equipped with 3D sensors used for field data collection. Middle: RGB-D images of broccoli plants (left) are analysed for identifying head locations using 3D recognition algorithms (right). Bottom: Temporal filtering then combines detections from multiple frames to localise individual broccoli heads.

II. Related work

Several approaches can be identified in precision agriculture for detection, recognition, localisation and harvesting of different crop varieties. Analysis of 2D images acquired from high resolution, industrial grade cameras such as CCD is one of the most prominent approaches as shown in [1],[2]. Several methods use colour analysis such as the strawberry picking robot in [3] and apple harvesting using colour and shape features in [4]. Okamoto et al. [5] developed a citrus

harvesting robot by using template matching to find circular objects on an edge image. Circular Hough transform voting was used to detect apples in [6]. Haug et al. [7] classify carrot crops versus weeds using geometric features combined with a random forest classifier. Other sensors were also investigated such as NIR spectral imaging for harvesting capsicum in a cluttered environment using texture features [8].

Using only image analysis may be insufficient for reliable estimation of the crop locations for machine vision systems in outdoor field conditions. Hence there has been an increase in the use of 3D sensors for depth perception [9]. The advantages of 3D include encoding the object geometry and providing different viewpoints under clutter. Barnea et al. [10] used combined information from RGB and depth for detecting sweet peppers, by detecting highlights in the image planes on registered RGB-D images to identify fruit regions and classifying peppers based on surface normal distribution and 3D object symmetry. The cucumber harvesting robot in [11] employs high resolution CCD cameras for detection of crops in greenhouses and 3D data for localisation. Weiss et al. [12] use 3D LIDAR for maize row detection and mapping using 3D geometry features. Gai et al. [13] proposed to discriminate crops from weeds using a Kinect 2 sensor and a combination of 2D and 3D morphological features. Nguyen et al. [14] employ an RGB-D sensor to detect apples, by using 3D data processing to segment out clusters in a point cloud and applying a circular Hough transform to the 2D transformed image to detect apples. The method uses color filtering which limits the kind of apples that can be detected.

Several approaches use RGB images to identify broccoli heads. Ramirez [15] explored a number of standard image processing techniques including texture to identify broccoli heads and showed that the approach has promise for in field selection, but the study was limited to a very small sample size (13 images). Tu et al. [16] showed that image analysis techniques and neural networks could be applied to identify broccoli quality parameters, but the approach was limited to heads imaged on a white light stable background, isolated from the leaves. Our approach extends the state of the art by detecting and localising broccoli heads with low-cost 3D sensors in real, unstructured outdoor field conditions.

III. Hardware Platform

One of the main requirements of the data collection is reliable RGB-D data capture in outdoor field conditions under different weather conditions such as sunny or overcast. In earlier work we found that an Asus Xtion Pro sensor performed poorly under ambient sunlight due to interference with its infra-red sensing for measuring depth. So in this work we evaluated the Kinect 2, a state-of-the-art low-cost RGB-D sensor based on time-of-flight technology [17]. The Kinect 2 provides high resolution RGB images at 1920×1080 pixels along with a depth resolution of 512×424. The sensor was fixed inside a specially constructed enclosure, which was mounted on the front of a tractor. The enclosure acts as a "shroud" to block direct sunlight incident at the sensor to alleviate noise and also as an "umbrella" during rainy conditions. The enclosure was equipped with an artificial lighting source, comprising strip LED lighting, to help regularize the colour images from the sensor, and to enable data capture during both day-time and night-time conditions. The sensor was mounted upright and at different heights from the ground (65-80 cm in our experiments), see Figure 2. We used the auto-calibration of colour and depth images using the factory defaults provided by the Kinect 2, and the sensor data were recorded with a standard laptop.

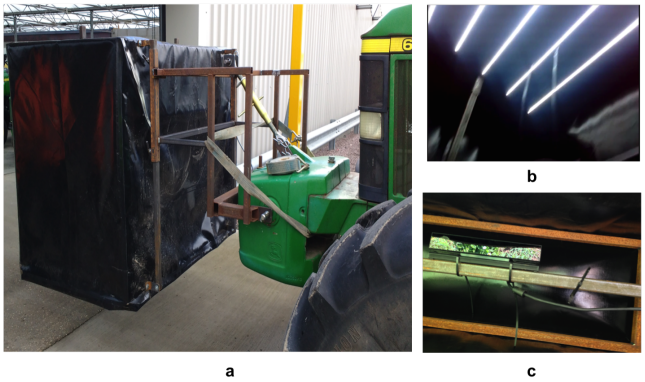


Fig. 2: Enclosure and lighting set-up. (a) Manually adjustable shroud attached to the front of the tractor (b) artificial lighting using LED strip lights (c) inside view of the enclosure with Kinect 2 mounted perpendicular to the field.

IV. Vision System

The vision system processes the depth data and the pipeline comprises four main steps: (i) 3D point cloud pre-processing, (ii) feature extraction, (iii) classification, and (iv) temporal filtering, as shown in Figure 3.

A. 3D point cloud pre-processing

We pre-process the raw point cloud data captured by the sensor to remove outliers, segment out the ground plane and group the remaining point cloud segments into clusters. We use the algorithms available as part of the PCL C++ library [18] for processing point clouds.

1) Outlier removal: The input point cloud data may contain outliers resulting from sensor measurement inaccuracies which can be considered as noise. It is important to remove these noisy data as they may lead to errors in subsequent processing. We use a statistical outlier removal algorithm that analyses the distribution of the distances between neighbouring points. For each point in the input cloud the algorithm computes the distances to its k neighbours and finds the mean and standard deviation of these distances ($k = 1000$ in our experiments). It removes all those points that fall outside a certain distance threshold defined by the sum of the global mean and standard deviation.

2) Ground filtering: The segmentation of the ground is achieved through thresholding the depth range, i.e, the z dimension of the input point cloud. The depth data are filtered at a user defined range of 0.5-1m and the points

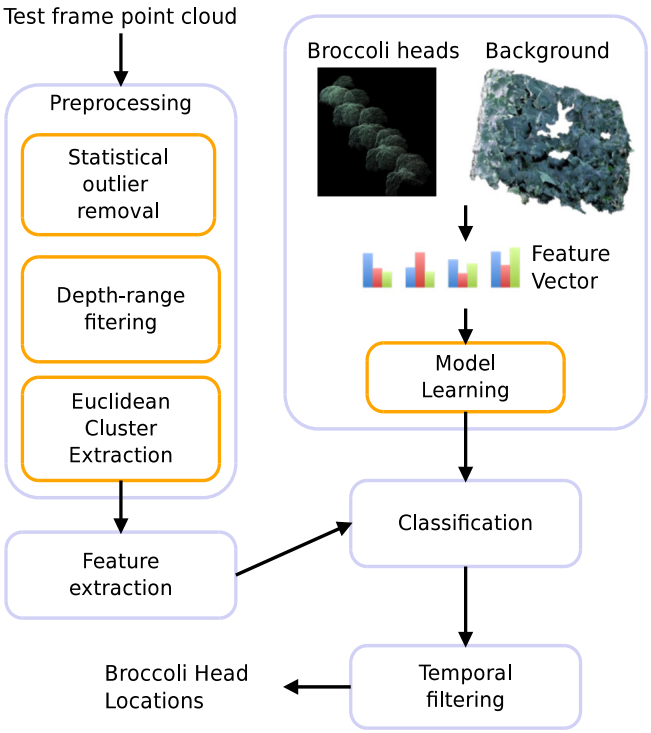


Fig. 3: 3D vision system pipeline. The frames of 3D point cloud data are first processed by pre-processing routines for outlier removal, depth filtering and cluster extraction. Then features are extracted and analysed using a learned model to predict the target class. The returned detections are used by temporal filtering to remove false positives.

that lie outside the range are discarded. We defined these parameters based on the distance of the sensor to the ground measured during data collection.

3) Cluster segmentation: The next step is to group the remaining point cloud segments into different clusters for segmentation. We use a distance-based clustering algorithm to cluster points based on the Euclidean distance between point pairs. The algorithm chooses each point and considers its neighbours defined by a certain radius. It greedily adds new points to the current cluster if the distance of the neighbouring points are within a user-defined cluster tolerance.

An important parameter to this algorithm is the cluster tolerance, which is set to 5mm. Smaller values would result in over-segmentation and higher values would result in merged clusters. In this case, we search for prominent foreground objects, and set the maximum and minimum size of the returned clusters as 500 and 10000 respectively.

Figure 4 shows example results of the pre-processing steps.

B. Feature Extraction

We use global 3D feature descriptors that describe the geometry of an object as a whole. The features are extracted for each of the clustered segments in the input point cloud derived after pre-processing. A good feature descriptor should be discriminative with respect to the two given classes,

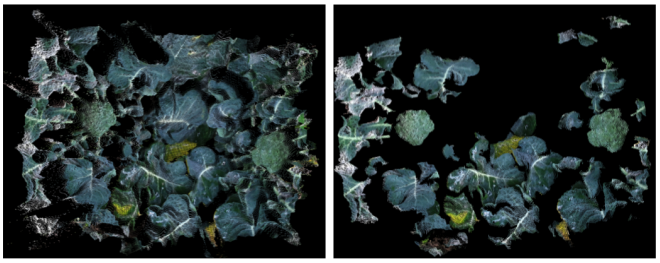


Fig. 4: 3D pre-processing. (a) Point cloud after depth filtering. (b) Point cloud after cluster extraction.

i.e., broccoli heads and non-broccoli segments representing leaves, ground or weeds. We describe the different 3D feature extraction algorithms used in the pipeline as follows.

1) Histogram Angle Features: We implemented a simple 3D feature based on the distribution of the orientation of surface normals of the point clouds. The essential idea is that the distribution of the surface normal directions should encode the underlying geometry of the broccoli heads and hence be discriminative compared to that of the leaves or other background clusters. For each of the segments returned by the clustering, we extract the surface normals. The view point normal contains the direction of orientation of the sensor, which is (0,0,-1) in this case, as the sensor is pointing down at the field. We then compute the angle between the surface normals for each point and the viewpoint normal. These angles are then binned into a histogram of the range 0-180 degrees with 12 orientation bins and normalised further.

2) VFH Features: The viewpoint feature histogram descriptor [19] is based on the FPFH features in PCL. The descriptor consists of two parts, a viewpoint direction and an extended Fast Point Feature Histogram (FPFH) descriptor. The viewpoint component is computed by first calculating the centroid of the input cluster and then the vector between the viewpoint or the position of the sensor and the centroid is computed. The resulting viewpoint vector is then normalized and for each point in the cluster, the angle between the viewpoint vector and the surface normal is calculated and binned into a histogram of 128 bins. The viewpoint vector is translated to each point location before calculating the angle to achieve scale invariance. The extended FPFH component is computed by calculating the roll, pitch and yaw angles between the viewpoint direction vector at the centroid and the surface normal at each point. These angles values contribute to 3 histograms of 45 bins each. The feature vector also has a shape distribution component that computes the distances of the points of cluster to its centroid. The final resulting feature vector has a dimension of 308.

3) CVFH Features: Clustered Viewpoint Feature Histogram features [20] are an extension to the VFH features for robustness against occlusions and partial views. The features are computed by first dividing the cluster into multiple smooth and stable regions using a region-growing segmentation algorithm. The VFH features are then calculated for each of the smooth regions forming the final descriptor. This

property allows the recognition of the clusters given that only partial views are available.

4) Geometric features: The geometric features consists of a set of measures that define different geometric properties of objects. We use the measures defined in [21], which characterise the underlying geometry of the segments by using the morphological attributes defined as follows:

i Compactness: estimates the compactness of the cluster by computing the ratio of the total surface area of the cluster to the surface area of the smallest binding sphere. The more spherical the cluster looks, the higher would be the compactness value.

ii Smoothness: measures if the neighbourhood around a point is spread uniformly by projecting the neighbourhood points to the tangent plane defined by the surface normal at the point. The entropy of the point distribution in 2D represents the smoothness value, where high entropy implies high smoothness.

iii Local convexity: measures whether the local convexity by determining the convexity of the polygonal edges in the mesh. Each cluster is ranked by the percentage of detected convex edges.

iv Symmetry: computes the score for reflective symmetry of a cluster through three different principal axes using eigen values. The cluster is reflected through the principal axis and the overlap between the original and reflected points is measured.

v Area: defined by the total number of points in a cluster.

C. Model Learning and Classification

The final stage is classification, where machine learning algorithms are used to learn the appearance of the broccoli heads using one or more of the above features. We use the models learnt by the learning algorithms to distinguish between broccoli heads and background leaves or ground.

1) K-Nearest Neighbours: KNN is a popular instance based classification algorithm where the training phase involves storing the feature vectors along with the class labels. Given a new instance x_i, the nearest neighbour algorithm searches for the k nearest neighbours to the query point in the training set. A distance metric such as Euclidean or Hamming distance can be used to rank the neighbours. The class that represents the majority of the neighbours is assigned as the predicted class of the query instance.

2) Support Vector Machine: The Support Vector Machine (SVM) is a binary classification algorithm. The SVM is shown to be efficient even in cases where the data is not linearly separable. It can also be used to classify data in higher dimensions using kernels.

Given a set of training data consisting of N input vectors $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}$, where $\mathbf{x}_i \in \Re^n$, along with the corresponding class labels, $t_n \in \{-1, +1\}$, the linear discriminant function that separates the two classes is given by

$$y(\mathbf{x}) = \mathbf{w^T}\boldsymbol{\phi}(\mathbf{x}) + b, \tag{1}$$

where $\mathbf{w}$ is the weight vector, $\boldsymbol{\phi}(\mathbf{x})$ is the feature vector and b is the bias. If the training data is linearly separable, then the sign of the function determines the target class assigned to the data points, i.e., $t_n y(\mathbf{x}_n) > 0$ holds true for all correctly classified instances. The parameter C controls the trade-off between training errors and generalization or complexity of the classifier and this parameter can be tuned for the given data using cross-validation on a grid search.

3) Training phase: We collected the training data from the dataset collected using Kinect 2 as described in Section III. The training data set comprises 32 point cloud segments of broccoli heads representing the positive class and 324 cloud segments representing the negative class. The training images were generated from pre-processing of the training dataset. The processed clusters were hand-labelled as broccoli head or background for ground truth.

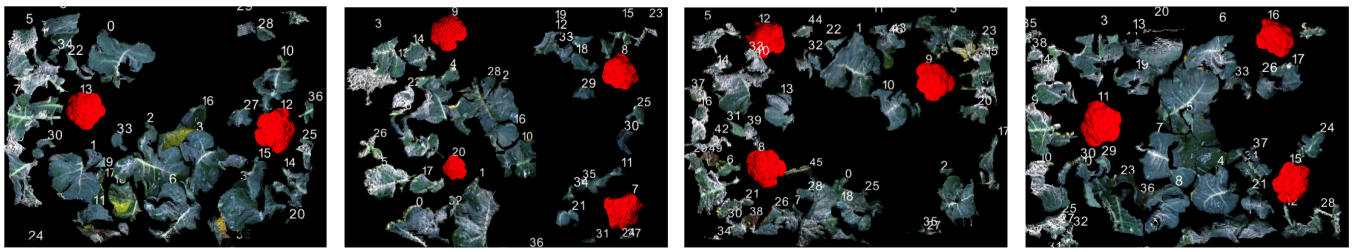
While using KNN for training, we simply extract the feature descriptors of the training data using one of the feature extraction algorithms. Each of the feature vectors is labelled as 1 or 0 representing the broccoli head or background class. We finally use these labelled feature histograms to classify the testing instances.

In order to train an SVM, we collect the training feature vectors and provide it as input the classifier along with the corresponding class labels. The SVM algorithm computes a model that can discrimate between the two classes. The parameter selection for optimal C value is also performed before the model learning using five-fold cross-validation on a grid search.

4) Classification phase: We first process each of the test cloud frames using the pre-processing pipeline to identify clusters. We then extract 3D global features from each of the cluster segments. The aim of the recognition algorithm is to classify each feature vector $\phi(\mathbf{x})$ as one of the target classes, t_1 or t_2. For KNN classification, we use Euclidean distance (L2 measure) to compute the distances between the feature histograms from the clusters in the test data and the corresponding feature histograms for the training set. For SVM classification, the classifier assigns scores to each of the segments in the test data using the learned model according to Equation 1. The scores are then thresholded to determine the class labels corresponding to the cluster. The output of the algorithm is a set of clusters representing the broccoli heads as shown in Figure 5, along with the x, y, z positions of the centroid locations of each of the clusters.

D. Temporal filtering and 3D mapping

Since the RGB-D sensor provides 15 frames per second and the harvester speed is approximately 0.3 m/s, the x, y, z positions of the individual broccoli heads in consecutive frames differ only by a few centimeters. This allows to track the locations of the broccoli heads over several frames as they pass through the sensed area. Thus, each detected broccoli head is assigned an additional 'tracking' score that indicates its number of detections. A low tracking score indicates that a given broccoli head was not detected consistently, which means that it is likely to be a false positive detection. Thus, broccoli heads with a low score are rejected as outliers.



(a) (b) (c) (d)

Fig. 5: The output of the detection algorithm. Numbers indicate the point cloud segments after cluster extraction, ordered by the size. The segments marked in red are the detections retrieved from the vision system using VFH features and SVM classifier.

The tracking results also allow to estimate the relative positions of the sensor between the individual frames, i.e. the tracking provides a position estimate of the harvester in the field. This estimate allows to transform the detected x, y, z positions of the broccoli from the coordinate system of the sensor to a global coordinate frame. In other words, the tracking mechanism allows to create a 3D map of the detected broccoli heads in the field as shown in the Figure 6.

The method assumes a constant forward velocity of 0.3 m/s in order to associate the detections for tracking. For details, please see the feature tracking method presented in [22].

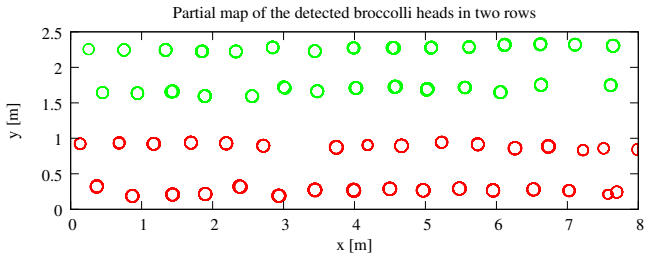


Fig. 6: 3D map. The locations of tracked detections of broccoli shown on a 3D map, generated from the testing data containing sequences from 2 tractor rows (4 rows of broccoli).

V. Experiments & Results

The data were captured near Surfleet, UK, using the set-up described in Section III towards the end of the UK harvesting season in November 2015. The tractor was driven through the broccoli field at a speed of approx. 0.3 m/s, with 2 rows of mature broccoli plants being imaged by the sensor at a rate of 15 frames per second. The weather included sunny, overcast and rainy conditions, with broccoli varying in maturity levels from small to large to already harvested (missing).

We pre-processed the 3D point cloud data using the described pipeline and extracted clusters or segments for each frame. We then labelled these segments as a broccoli head or not for the ground truth. The training images consisted of 32 point clouds representing broccoli heads and 324 point cloud segments representing other objects, including leaves, weeds and background. The test set comprises 600 point cloud frames of 1619 annotated instances of broccoli heads taken from two separate tractor rows, different from that of the training set, but from the same session. The data comprises a sequence of 300 images for each row, in order to facilitate the evaluation of multi-frame filtering. We implemented the software in C++ with PCL library on a PC running Ubuntu and i7 processor with 8 GB RAM.

We evaluated the vision system for detecting broccoli heads using multiple feature descriptors as mentioned in Section IV and using two classifiers KNN and SVM. The parameters of the pre-processing algorithms, feature descriptors and classifiers were tuned according to a validation set. The normals were computed around a neighbourhood of 5 mm. The cluster tolerance was set to 2 mm and the depth range filter had a range of 0.5 m to 1 m. The best C value for the linear SVM was chosen as 0.002 for the best performing features, by using cross-validation on a grid search. The number of nearest neighbours was chosen as 11. We use precision-recall curves to evaluate the performance of the classifier and report the average precision as in [23]. We report the results of the experiments in Figure 7 using average precision. The results show that the surface normal geometry based features, VFH along with SVM, gives the highest accuracy on the test data of 94.7%. We show that temporal filtering improves the average precision of all the feature combinations using SVM. The average run time of the entire pipeline per images is 5-6s. The broccoli head detection algorithm provides the 3D coordinate locations of the centroids of the detected broccoli segments. The z location corresponds to the lowest value in the z direction that indicates the top of the head cluster.

Finally, we provide the results of the SVM-based classification augmented by the temporal filtering method described in Section IV-D. Figure 7c shows that rejecting detections which were not consistent over several frames improved the precision of the classification. The figure indicates that the combination of the Viewpoint Feature Histogram, Support Vector Machine and temporal filtering results in a system that detects broccoli heads with more than 95% precision.

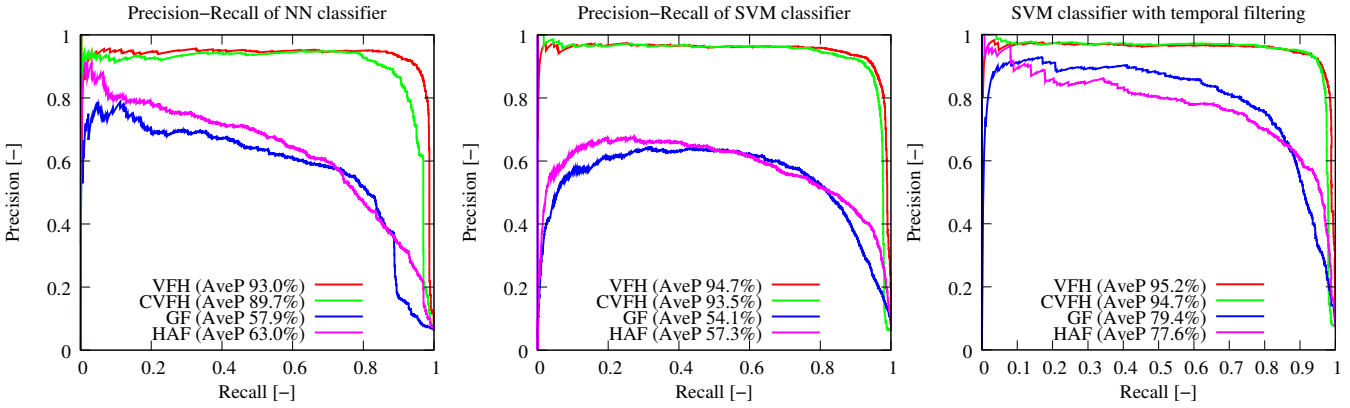


Fig. 7: Performance evaluation of different 3D features and (a) KNN, (b) SVM, (c) SVM with Temporal Filtering. VFH features with SVM gives the highest precision at 94.7%. Temporal filtering further improves this to 95.2%.

VI. Conclusion

This paper demonstrated the development of a 3D-based approach for detecting mature broccoli heads that could be applied in an automatic broccoli harvester. We showed that the depth images provided by low-cost RGB-D sensors can be used for reliable detection and localisation of broccoli heads in cluttered outdoor field conditions. We also showed that the information from a sequence of detections can be used to reliably track the individual broccoli and filter out false detections with a precision rate of 95.2%. Future work will include fusion of GPS and IMU data to geo-locate the broccoli, enabling further applications in field mapping and yield prediction. Texture features from the RGB images could also be added to further improve the results.

References

[1] C. L. McCarthy, N. H. Hancock, and S. R. Raine, "Applied machine vision of plants: a review with implications for field deployment in automated farming operations," *Intelligent Service Robotics*, vol. 3, no. 4, pp. 209–217, 2010.

[2] A. Jimenez, R. Ceres, and J. Pons, "A survey of computer vision methods for locating fruit on trees," *Transactions of the ASAE*, vol. 43, no. 6, p. 1911, 2000.

[3] S. Hayashi, K. Shigematsu, S. Yamamoto, K. Kobayashi, Y. Kohno, J. Kamata, and M. Kurita, "Evaluation of a strawberry-harvesting robot in a field test," *Biosystems Engineering*, vol. 105, no. 2, pp. 160–171, 2010.

[4] W. Ji, D. Zhao, F. Cheng, B. Xu, Y. Zhang, and J. Wang, "Automatic recognition vision system guided for apple harvesting robot," *Computers & Electrical Engineering*, vol. 38, no. 5, pp. 1186–1195, 2012.

[5] H. Okamoto and W. S. Lee, "Machine vision for green citrus detection in tree images," *Environmental Control in Biology*, vol. 48, no. 2, pp. 93–99, 2010.

[6] J. P. Wachs, H. Stern, T. Burks, and V. Alchanatis, "Low and high-level visual feature-based apple detection from multi-modal images," *Precision Agriculture*, vol. 11, no. 6, pp. 717–735, 2010.

[7] S. Haug, A. Michaels, P. Biber, and J. Ostermann, "Plant classification system for crop/weed discrimination without segmentation," in *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*. IEEE, 2014, pp. 1142–1149.

[8] I. Sa, C. McCool, C. Lehnert, and T. Perez, "On visual detection of highly-occluded objects for harvesting automation in horticulture," in *Proc. ICRA*, Seattle, Washington, May 26-30 2015.

[9] K. Kapach, E. Barnea, R. Mairon, Y. Edan, and O. Ben-Shahar, "Computer vision for fruit harvesting robots–state of the art and challenges ahead," *Int. J. Computational Vision and Robotics*, vol. 3, no. 1-2, pp. 4–34, 2012.

[10] E. Barnea, R. Mairon, and O. Ben-Shahar, "Colour-agnostic shape-based 3D fruit detection for crop harvesting robots," *Biosystems Engineering*, 2016.

[11] E. J. van Henten, J. Hemming, B. Van Tuijl, J. Kornet, J. Meuleman, J. Bontsema, and E. Van Os, "An autonomous robot for harvesting cucumbers in greenhouses," *Autonomous Robots*, vol. 13, no. 3, pp. 241–258, 2002.

[12] U. Weiss and P. Biber, "Plant detection and mapping for agricultural robots using a 3D LIDAR sensor," *Robot. Auton. Syst.*, vol. 59, no. 5, pp. 265–273, May 2011. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2011.02.011

[13] J. Gai, L. Tang, and B. Steward, "Plant recognition through the fusion of 2D and 3D images for robotic weeding," in *2015 ASABE Annual International Meeting*. American Society of Agricultural and Biological Engineers, 2015, p. 1.

[14] T. T. Nguyen, K. Vandevoorde, E. Kayacan, J. De Baerdemaeker, and W. Saeys, "Apple detection algorithm for robotic harvesting using a RGB-D camera," in *International Conference of Agricultural Engineering, Zurich, Switzerland*, 2014.

[15] R. A. Ramirez, "Computer vision based analysis of broccoli for application in a selective autonomous harvester," Master's thesis, Virginia Polytechnic Institute and State University, July 2006.

[16] K. Tu, K. Ren, L. Pan, and H. Li, "A study of broccoli grading system based on machine vision and neural networks," in *Proc. ICMA*. IEEE, 2007, pp. 2332–2336.

[17] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in *in Proc. ICAR*, 2015, pp. 388–394.

[18] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proc. ICRA*, Shanghai, China, May 9-13 2011.

[19] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *Proc. IROS*, 2010, pp. 2155–2162.

[20] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, "CAD-model recognition and 6DOF pose estimation using 3D cues," in *IEEE International Conference on Computer Vision Workshops*, Barcelona, Spain, Nov. 6-13 2011, pp. 585–592.

[21] A. Karpathy, S. Miller, and L. Fei-Fei, "Object discovery in 3d scenes via shape analysis," in *Proc. ICRA*, 2013, pp. 2088–2095.

[22] T. Krajník et al., "Simple yet stable bearing-only navigation," *Journal of Field Robotics*, vol. 27, no. 5, pp. 511–533, 2010.

[23] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.