Dense Segmentation of Textured Fruits in Video Sequences

Waqar S. Qureshi[1,3], Shin'ichi Satoh[2], Matthew N. Dailey[3] and Mongkol Ekpanyapong[3]
[1]*School of Mechanical & Manufacturing Engineering, National University of Science & Technology, Islamabad, Pakistan*
[2]*Multimedia Lab, National Institute of Informatics, Tokyo, Japan*
[3]*School of Engineering and Technology, Asian Institute of Technology, Pathumthani, Thailand*

Keywords: Super-pixels, Dense Classification, Visual Word Histograms.

Abstract: Autonomous monitoring of fruit crops based on mobile camera sensors requires methods to segment fruit regions from the background in images. Previous methods based on color and shape cues have been successful in some cases, but the detection of textured green fruits among green plant material remains a challenging problem. A recently proposed method uses sparse keypoint detection, keypoint descriptor computation, and keypoint descriptor classification followed by morphological techniques to fill the gaps between positively classified keypoints. We propose a textured fruit segmentation method based on super-pixel oversegmentation, dense SIFT descriptors, and and bag-of-visual-word histogram classification within each super-pixel. An empirical evaluation of the proposed technique for textured fruit segmentation yields a 96.67% detection rate, a per-pixel accuracy of 97.657%, and a per frame false alarm rate of 0.645%, compared to a detection rate of 90.0%, accuracy of 84.94%, and false alarm rate of 0.887% for the baseline sparse keypoint-based method. We conclude that super-pixel oversegmentation, dense SIFT descriptors, and bag-of-visual-word histogram classification are effective for in-field segmentation of textured green fruits from the background.

1 INTRODUCTION

Precision agriculture aims to help farmers increase efficiency, enhance profitability, and lessen environmental impact, driving them towards technological innovation. The global trend towards large-scale plantation and demand to increase productivity has increased precision agriculture's importance.

One aspect of precision agriculture is crop inspection and monitoring. Conventional methods for inspection and monitoring are tedious and time consuming. Farmers must hire laborers to perform the task or use sample-based monitoring.

Over the years, researchers have investigated many technologies to reduce the burden and broaden the coverage of crop monitoring. Satellite remote sensing facilitates crop vegetation index monitoring through spectral analysis. Spectral vegetation indices enable researchers to track crop development and management in a particular region albeit at a very coarse scale.

Another solution to the monitoring problem is to use one or more camera sensors mounted on an autonomous robot. Among the difficulties with this approach are limitations in computational resources and image quality. However, it is possible to use simplistic processing for navigation but send sensor data to a host machine for more sophisticated offline processing to detect fruits or other features of interest; in this case the processing need not satisfy hard real-time constraints.

Our focus is thus to build an autonomous fruit crop inspection system incorporating one or more mobile camera sensors and a host processor able to analyze the video sequences in detail. The first step is to retrieve images containing fruits. Then we must segment the fruit regions from the background and track the fruit regions over time. Such a system can be used to monitor fruit health and growth trajectories over time and predict crop yield, putting more detailed information in the hands of the farmer than has previously been possible.

Several research teams have been developing methods to classify, segment, and track fruit regions from video sequences for fruit health monitoring, crop management, and/or yield prediction. The common theme is the use of supervised learning for classification and detection. The classifier may incorporate

Qureshi W., Satoh S., Dailey M. and Ekpanyapong M..
Dense Segmentation of Textured Fruits in Video Sequences.
DOI: 10.5220/0004689304410447
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 441-447
ISBN: 978-989-758-004-8

features characterizing color, shape, or texture of fruit or plants. Schillaci et al. (2012) focus on tomato identification and detection. They train a classifier offline on visual features in a fixed sized image window. The online detection algorithm performs a dense multiscale scan over a scanning window on the image. Sengupta and Lee (2012) present a similar method to detects citrus fruits. However, such methods are dependent on the shape of the fruit and will not work when the shape of the fruit region is highly variable due to occlusion by plant material.

Roy et al. (2011) introduce a method to detect pomegranate fruits in a video sequence that uses pixel clustering based on RGB intensities to identify frames that may contain fruits then uses morphological techniques to identify fruit regions. The authors use *k*-means clustering based on grayscale intensity, then, for each cluster, they calculate the entropy of the distribution of pixel intensities in the red channel. They find that clusters containing fruit regions have less random distributions in the red channel, resulting in lower entropy measurements, allowing frames containing fruits to be selected efficiently. Dey et al. (2012) demonstrate the use of structure from motion and point cloud segmentation techniques for grape farm yield estimation. The point cloud segementation method is based on the color information in the RGB image. Another method based on RGB intensities is presented by Diago et al. (2012). They characterize grapevine canopy and leaf area by classification of individual pixels using support vector machines (SVMs). The method measures the area (number of pixels) of image regions classified into seven categories (grape, wood, background, and 4 classes of leaf) in the RGB image. All of these methods may work with fruit that are distinguishable from the background by color but would fail for fruits that have color similar to the color of the plants.

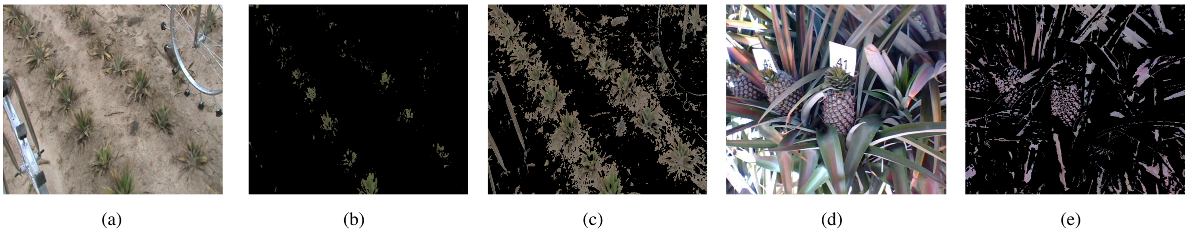
Much of the previous research in this field has made use of the distinctive color of the fruits or plants of interest. When the object of interest has distinctive color with respect to the background, it is easy to segment based on color information then further process regions of interest. To demonstrate this, consider the image of young pineapple plants in Figure 1(a). We built a CIELAB color histogram from a ground truth segmentation of a sample image then thresholded the back-projection of the color histogram onto the original image. As can be seen from Figures 1(b)–1(c), the plants are quite distinctive from the background. However, color based classification fails when the objects of interest (e.g., the pineapples in Figure 1(d)) have coloration similar to that of the background, as shown in Figure 1(e).

To address the issue of objects of interest that blend into a similar-colored background, Chaivivatrakul and colleagues (Chaivivatrakul et al., 2010; Moonrinta et al., 2010) describe a method for 3D reconstruction of pineapple fruits based on sparse keypoint classification, fruit region tracking, and structure from motion techniques. The method finds sparse Harris keypoints, calculates SURF descriptors for the keypoints, and uses a SVM classifier trained offline on hand-labeled data to classify the local descriptors. Morphological closing is used to segment the fruit using the classified features. Fruit regions are tracked from frame to frame. Frame-to-frame keypoint matches within putative fruit regions are filtered using the nearest neighbor ratio, symmetry test, and epipolar geometry constraints, then the surviving matches are used to obtain a 3D point cloud for the fruit region. An ellipsoid model is fitted to the point cloud to estimate the size and orientation of each fruit. The main limitation of the method is the use of sparse features with SURF descriptors to segment fruit regions. Filling in the gaps between sparse features using morphological operations is efficient but leads to imprecise delineation of the fruit region boundaries. To some extent, robust 3D reconstruction methods can clean up these imprecise boundaries, but the entire processing stream would be better served by an efficient but accurate classification of *every pixel in the image*.

Unfortunately, calculating a texture descriptor for each pixel in an image then classifying each descriptor using a SVM or other classifier would be far too computationally expensive for a near-real-time video processing application.

In this paper, we therefore explore the potential of a more efficient dense classification method based on the work of Fulkerson et al. (2009). The authors construct classifiers based on histograms of local features over super-pixels then use the classifiers for segmentation and classification of objects. They demonstrate excellent performance on the PASCAL VOC challenge dataset for object segmentation and localization tasks. For fruit detection, super-pixel based methods are extremely useful, because super-pixels tend to adhere to natural boundaries between fruit and non-fruit regions of the image, leading to precise fruit region boundaries, outperforming sparse keypoint methods in terms of per-pixel accuracy. To our knowledge, dense texture-based object segmentation and classification techniques have never been applied to detection of fruit in the field where color based classification does not work.

In the rest of the paper we describe our algorithm and implementation, perform a qualitative and quan-



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

Figure 1: Example CIELAB color histogram based detection of plants. (a) Young plants. (b) Detection of plants in image (a) using a conservative threshold. (c) Detection of plants in image (a) using a liberal threshold. (d) Grown plants. (e) CIELAB color histogram based detection of fruit in image (d) using a conservative threshold.

titative analysis of experimental results, and conclude with a discussion of possibilities for further improvement.

2 METHODOLOGY

Following Fulkerson et al. (2009), our methodology for localized detection of pineapple fruit is based on SVM classification of visual word histograms. It consists of two components, offline training of the classifier and online detection of fruit pixels using the trained classifier. Offline training requires a set of labeled images selected from one or more training video sequences. Here we summarize the algorithm and provide implementation details for effective fruit segmentation.

2.1 Training Algorithm

Inputs: I (number of training frames), V (maximum number of visual words), D (number of descriptors used to train clustering model), H (number of super-pixel histograms to select per image), N (number of adjacent super pixels to include in each super-pixel's histogram), CRF flag (whether to include a conditional random field in the model).

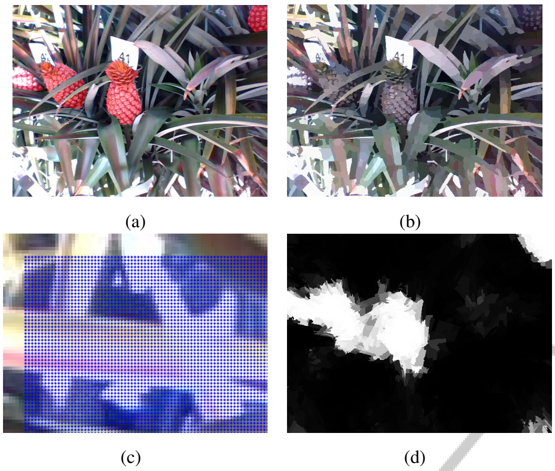
1. Generate the training data by manual annotation:
   (a) Select I frames manually and segment the required object of interest (pineapple) for each frame.
   (b) Divide the data into training and cross validation data.
2. Perform super-pixel based image segmentation for each frame.
3. Extract dense descriptors for each frame using the dense-SIFT algorithm.
4. Create dictionary of visual words:
   (a) Randomly select D descriptors over all training images for clustering.
   (b) Run k-means to obtain V clusters corresponding to visual words.
5. Extract training bag-of-visual-word histograms:
   (a) Randomly select H super-pixels per image.
   (b) For each super-pixel and its N nearest neighbors, construct a histogram counting the occurrence of each dense-SIFT visual word.
   (c) Normalize each histogram using $L1$ norm.
6. Train a RBF-based SVM on the training histograms.
7. Validate the classifier using cross validation images.
8. If conditional random field (CRF) training is desired, train a CRF model.

Fulkerson et al.'s CRF is trained to estimate the conditional probability of each super-pixel's label based on the SVM classification results and the super-pixel adjacency graph. A unary potential encourages labeling with the SVM classifier's output, while binary potentials encourage labelings consistent with neighboring super-pixels. The optimization tends to improve the consistency of the labeling over an entire image.

2.2 Runtime (Model Testing) Algorithm

Inputs: test video sequence, model from training phase.

1. For each frame in the video sequence:
   (a) Perform super-pixel based image segmentation.
   (b) Extract visual word histograms for each super-pixel.
   (c) Classify each super-pixel as fruit or non-fruit using the trained classifier.
   (d) If CRF post-processing is desired, reclassify each super-pixel using the CRF.
   (e) Compare segmentation result with ground truth.



(a)

(b)

(c)

(d)

Figure 2: Sample procesing of an image acquired in a pineapple field. (a) Ground-truth annotation of pineapple fruit. (b) Quick-shift super-pixel segmentation of image (a). (c) Dense-SIFT descriptors are calculated for every image pixel except those near the image boundary (the image shown is a magnification of the upper left corner of image (a). (d) Confidence map for pineapple fruit. Higher intensity indicates higher confidence in fruit classification.

2.3 Implementation Details

Construction and training of the offline classifier requires training images and ground-truth data. Ground-truth labels (fruit or non-fruit) for each pixel in each training and test image must be prepared manually. Figure 2(a) shows an example of the ground-truth generated for pineapple fruits, including the fruit crown in one test image. The red regions are fruit pixels.

We use quick-shift (Vedaldi and Soatto, 2008) for super-pixel segmentation. Quick-shift is a gradient ascent method that clusters five-element vectors containing the (x,y) position and CIELAB color of each pixel in the image. We manually adjusted quick-shift's parameters (ratio=0.5, kernel size = 2, and maximum distance=6) through preliminary experiments such that the boundaries of fruit and plants are preserved with large possible segments. These paramenters are dependent on image resolution, distance to the camera, and clutteredness of the scene. The CIELAB color space separates luminosity from chromaticity, which normally leads to more reasonable super-pixel boundaries than would clustering based on color spaces such as RGB that do not separate luminosity and chromaticity components. Figure 2(b) shows a super-pixel segmentation of the image from Figure 2(a) constructed by assigning each super-pixel's average color to all of the pixels in that super-pixel.

The algorithm uses dense-SIFT as the local descriptor for each pixel in a super-pixel. Dense-SIFT is type of histogram of oriented gradient (HOG) descriptor that captures the distribution of local gradients in a pixel's neighborhood. We used the fast DSIFT implementation in the VLFeat library (Vedaldi and Fulkerson, 2008). The method is equivalent to computing SIFT descriptors at defined locations at a fixed scale and orientation.

We use spatial bin size of 3×3, which creates a descriptor spanning a 12×12 pixel image region. We use a step size of 1, meaning that descriptors should be calculated for every pixel in the image except for boundary pixels (see Figure 2(c)).

We use k-means to cluster the SIFT descriptors across the training set. Based on experience from preliminary experiments, we use 1 million randomly selected descriptors and a maximum number of visual words $V = 100$.

A local histogram of the visual words (k-means cluster IDs) in each super-pixel is extracted then normalized using the $L1$ norm. Since local histograms based on a small number of pixels in a region with uniform coloration can be quite sparse, each histogram may be augmented by including the SIFT descriptors of neighboring super-pixels. We experimented with including 0, 1, 2, and 3 neighboring super-pixels in creating histograms. Figure 3 shows a comparison of the effect of adding adjacent super-pixels to two sample super-pixel histograms in a training image. The figure shows the increase in density as adjacent neighboring super-pixels are added to the histogram. In the experimental results section, we further discuss experiments to establish the best number of adjacent super-pixels.

Our method uses SVMs to perform binary classification of super-pixel visual word histograms as fruit or not fruit. To find the optimal parameters of the classifier, for each training image, we randomly extracted 100 histograms with an equal number of positive and negative examples. Each training super-pixel is labeled with the ground-truth label of the majority of pixels in the super-pixel. We use the radial basis function (RBF) kernel for the SVM classifier, using cross validation to find the hyperparameters c and γ. We use Fulkerson et al.'s conditional random field (CRF) as a post-process on top of the SVM classification. The CRF is trained over a subset of the training images.

Segmenting a new image using the trained model requires super-pixel over-segmentation, dense-SIFT descriptor computation, visual word histogram calculation, and SVM classification. The classifier outputs a confidence for each super-pixel; a super-pixel is classified as a fruit region if the confidence is higher than a threshold. Figure 2(d) shows a confi-

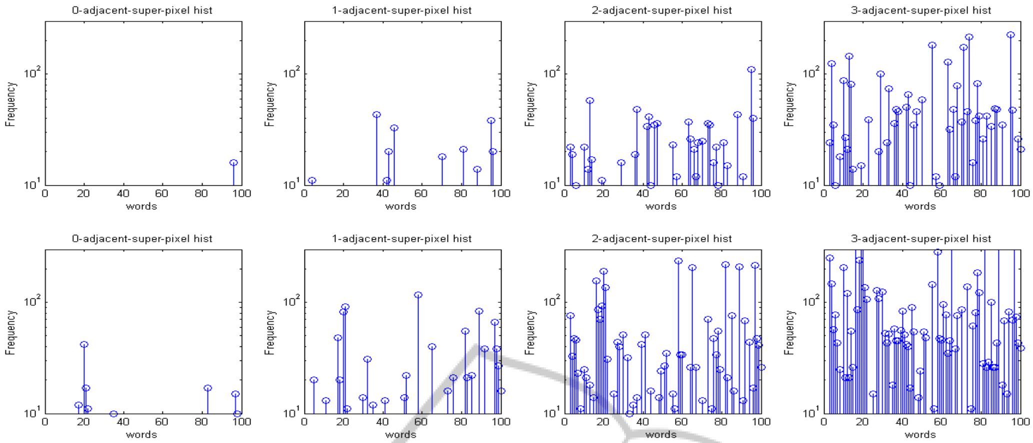


Figure 3: Comparison of visual word histograms of two randomly selected super-pixels with 0, 1, 2, or 3 adjacent super-pixels included in the histogram.

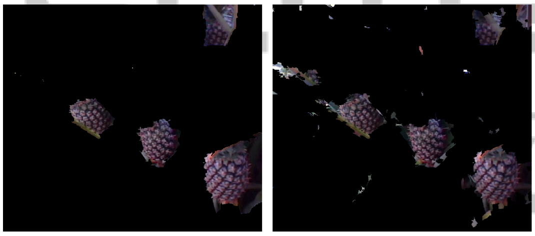


Figure 4: Comparison of fruit segmentation with (left) and without (right) CRF post-processing of the SVM classifier's output.

dence map for the image from Figure 2(a). Finally, the SVM classification results are post-processed using the trained CRF. Figure 4 shows the qualitative improvement in the segmentation after CRF post-processing.

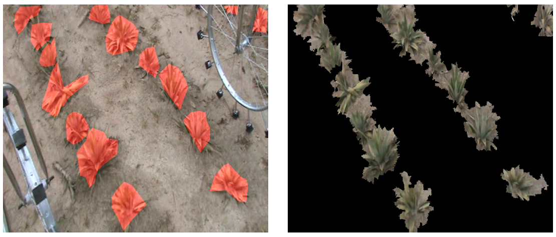
3 EXPERIMENTAL RESULTS

In order to evaluate the proposed method, we performed three experiments. In the first experiment we processed video data on young pineapple plants with the aim of segmenting the plants from the background, and in the second experiment, we processed video data on mature plants with the aim of segmenting fruit from the background. In the third experiment, we compare the best classifier from Experiment II with the method of Chaivivatrakul and colleagues. We obtained the datasets from the authors. The video sequences were acquired from monocular video cameras mounted on a ground robot. We used dual-core machine with 2GB of memory as the host machine. Details of each experiments are given below.

3.1 Experiment I: Young Pineapple Plants

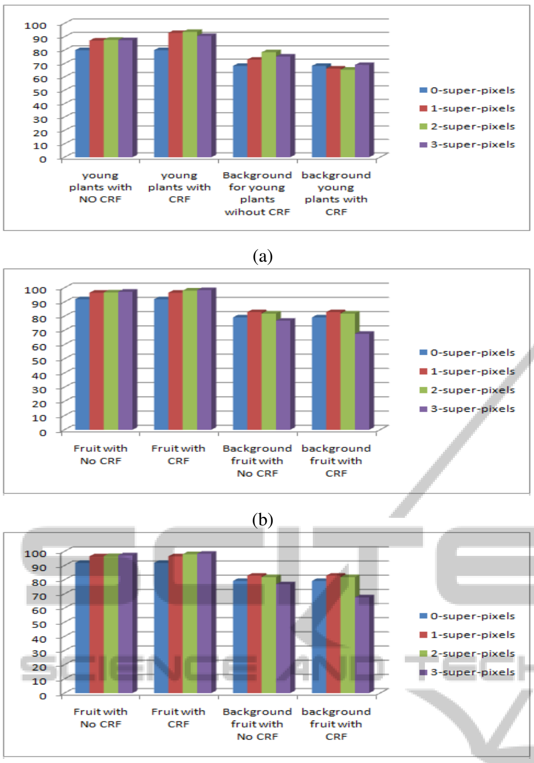
The young pineapple data set is a 63-second video from one row of a field acquired at 25 fps. We extracted 27 images, selecting 20 for training and reserving seven for the test set. We ensured that the plants in the training images did not overlap with the plants in the test images. We performed ground-truth labeling of plant and non-plant regions using the VOC tool (Vanetti, 2010). Figure 5(a) shows the labeling of a sample image.

We tested with different values for N (the number of adjacent neighboring super-pixels to include in each histogram). A sample result of online detection, obtained with $N = 2$ and CRF post-processing, is shown in Figure 5(b). The per-pixel accuracy data are summarized in Figure 6(a).



(a) (b)

Figure 5: Detection of young pineapple plants. (a) Manually generated ground-truth segmentation. (b) Automatic detetction using proposed method.



(a)

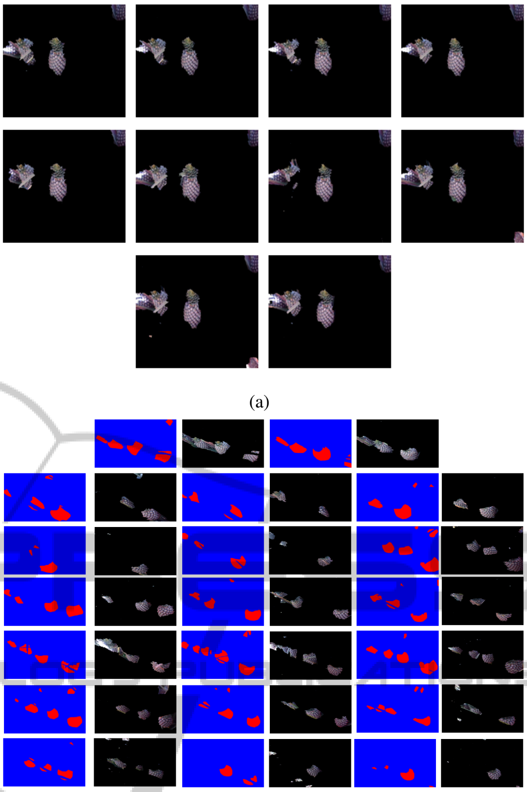
(b)

(c)

Figure 6: Per-pixel segmentation accuracy in Experiments I and II. (a) Experiment I results (young pineapple plants). (b) Experiment II results (pineapple fruit segmentation) labeling fruit crown as non-fruit. (c) Experiment II results (pineapple fruit segmentation) labeling fruit crown as fruit.

3.2 Experiment II: Mature Pineapple Plants

The mature pineapple plant dataset contains video sequences for six rows of plants. To train a first version of the model, we selected 50 images from the sixth row, out of which 40 were used for training and 10 were reserved for testing. For this data set, we prepared ground-truth labeling that includes the fruit crown as part of the fruit. For a second version of the model that can be compared directly to that of Moonrinta et al. (2010), we used a set of 120 images partitioned into six subsets. Each 20-image subset is extracted from the video of one row of plants such that each image contains at least one new fruit. The training comprised thw 100 images from rows 1, 2, 4, 5, and 6, while the test data comprised the 20 images from row 3. We selected 50 random images from the training dataset and used the same 20 images from the third row as the test set and used the same manual labeling provided by Moonrinta et al., in which only the fruit skin without the crown is labeled as fruit.



(a)

(b)

Figure 7: Experiment II results. (a) Results including fruit crown as part of the fruit. (b) Results not including fruit crown as part of the fruit.

The per-pixel accuracy for both versions of the model and ground truth data are summarized in Figures 6(b) (crown not included as part of the fruit) and 6(c) (crown included as part of the fruit).

The accuracy data indicate that the classifier performs best with $N = 2$ and CRF post-processing. We use this configuration in subsequent comparisons. Sample segmentation results for both versions of the model with different ground truth conditions are shown in Figure 7.

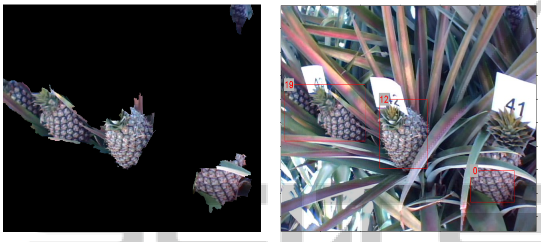
3.3 Experiment III: Comparison with Sparse Keypoints

In Experiment III, we compared our results with those of Chaivivatrakul and colleagues. We used the version of our model trained on the same data with fruit crowns not included in the ground truth labeling of fruit. The comparison is shown in Table 1. Clearly, oversegmentation and visual word histogram classification outperforms the state of the art method for segmentation of textured fruit.

The main reason for the improvement in perfor-

Table 1: Comparison of segmentation accuracy with other methods.

Method	Pixel Accuracy	Hits	Misses	False alarms
Moonrinta et al. (2010)	84.94%	90%	10%	0.887%
Chaiviva-trakul et al. (2010)	87.79%	NA	NA	NA
Proposed Method	97.657%	96.67%	3.22%	0.645%



(a)

(b)

Figure 8: Pineapple fruit detection. (a) Detection and precise segmentation using our implementation. (b) Detection using method of Moonrinta et al. (2010).

mance is that the super-pixel segmentation tends to conform to the fruit-background boundary, whereas the morphological processing following sparse keypoint classficiation does not. Figure 8 shows a sample of the resulting of segmentation by each method.

4 CONCLUSIONS AND FUTURE WORK

We have demonstrated a dense approach to textured fruit segmentation. An immediate extension of our work would be to find dense correspondences between fruit regions in subsequent images in the video sequence; this would help us track fruit accurately over time. The dense correspondence method could be performed as described by Lhuillier and Quan (2005). Dense-depth information could be estimated for fruit regions using structure from motion methods (Pollefeys et al., 2004), which would further aid in tracking of individual fruit. We intend to evaluate the demonstrated method on other crops such as corn and on video sequences obtained from aerial vehicles in addition to ground vehicles.

ACKNOWLEDGEMENTS

Waqar S. Qureshi was supported by a graduate fellowship from NUST, Pakistan as well as an internship at the Satoh Multimedia Lab at NII, Japan. We thank Supawadee Chaivivatrakul for providing the dataset for Experiment II.

REFERENCES

Chaivivatrakul, S., Moonrinta, J., and Dailey, M. N. (2010). Towards automated crop yield estimation: Detection and 3D reconstruction of pineapples in video sequences. In *International Conference on Computer Vision Theory and Applications*.

Dey, D., Mummert, L., and Sukthankar, R. (2012). Classification of plant structures from uncalibrated image sequences. In *IEEE Winter Conference on Applications and Computer Vision*, pages 329–336.

Diago, M.-P., Correa, C., Millán, B., Barreiro, P., Valero, C., and Tardaguila, J. (2012). Grapevine yield and leaf area estimation using supervised classification methodology on rgb images taken under field conditions. *Sensors*, 12(12):16988–17006.

Fulkerson, B., Vedaldi, A., and Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *International Conference on Computer Vision (ICCV)*, pages 670–677.

Lhuillier, M. and Quan, L. (2005). A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433.

Moonrinta, J., Chaivivatrakul, S., Dailey, M. N., and Ekpanyapong, M. (2010). Fruit detection, tracking, and 3d reconstruction for crop mapping and yield estimation. In *IEEE International Conference on Control, Automation, Robotics and Vision*.

Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., and Koch, R. (2004). Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232.

Roy, A., Banerjee, S., Roy, D., and Mukhopadhyay, A. (2011). Statistical video tracking of pomegranate fruits. In *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, pages 227–230.

Schillaci, G., Pennisi, A., Franco, F., and Longo, D. (2012). Detecting tomato crops in greenhouses using a vision based method. In *International Conference on Safety, Health and Welfare in Agriculture and Agro*.

Sengupta, S. and Lee, W. S. (2012). Identification and determination of the number of green citrus fruit under different ambient light conditions. In *International Conference of Agricultural Engineering*.

Vanetti, M. (2010). Voc dataset manager software.

Vedaldi, A. and Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms. urlhttp://www.vlfeat.org/.

Vedaldi, A. and Soatto, S. (2008). Quick shift and kernel methods for mode seeking. In *European Conference on Computer Vision (ECCV)*.