



American Society of  
Agricultural and Biological Engineers

*An ASABE Meeting Presentation*

*Paper Number: 063060*

## **Segmentation of Apple Fruit from Video via Background Modeling**

### **Amy L. Tabb**

Engineer and graduate student, Appalachian Fruit Research Station/ARS/USDA and  
Electrical and Computer Engineering Department, Purdue University.  
atabb@afrs.ars.usda.gov

### **Donald L. Peterson**

Agricultural Engineer. ddpeterson@adelphia.net

### **Johnny Park**

Principal Research Scientist, School of Electrical and Computer Engineering, Purdue  
University. jpark@purdue.edu

**Written for presentation at the  
2006 ASABE Annual International Meeting  
Sponsored by ASABE  
Oregon Convention Center  
Portland, Oregon  
9 - 12 July 2006**

**Abstract.** *A method for locating apples was developed to process real-time video image sequences captured with an over-the-row harvester. The concepts of background modeling in RGB color were used, which is a novel approach to the apple segmentation problem. In background modeling, the distributions of background colors are approximated from real data. The algorithm developed for this task, Global Mixture of Gaussians (GMOG), is based on the principles of Mixture of Gaussians (MOG), which is used for motion-detection applications. The algorithm correctly identified ~85-96% of both red and yellow apples and performed at ~14-16 frames per second. This is the first work to our knowledge that uses video sequences to detect apple fruit. The potential advantages of using video include allowing harvesting on-the-go, detecting occluded fruit via camera movement to the occluded areas, using visual servoing of robotic grippers to grasp fruit, and achieving a faster harvest time.*

**Keywords.** Computer vision, apple, video, segmentation, harvesting, orchard automation

---

The authors are solely responsible for the content of this technical presentation. The technical presentation does not necessarily reflect the official position of the American Society of Agricultural and Biological Engineers (ASABE), and its printing and distribution does not constitute an endorsement of views which may be expressed. Technical presentations are not subject to the formal peer review process by ASABE editorial committees; therefore, they are not to be presented as refereed publications. Citation of this work should state that it is from an ASABE meeting paper. EXAMPLE: Author's Last Name, Initials. 2006. Title of Presentation. ASABE Paper No. 06xxxx. St. Joseph, Mich.: ASABE. For information about securing permission to reprint or reproduce a technical presentation, please contact ASABE at [hq@asabe.org](mailto:hq@asabe.org) or 269-429-0300 (2950 Niles Road, St. Joseph, MI 49085-9659 USA).

---

All programs and services of the USDA are offered on a nondiscriminatory basis with regard to race, color, national origin, religion, sex, age, marital status, or handicap. Mention of trade names or commercial products in this publication is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the U.S. Department of Agriculture.

---

The authors are solely responsible for the content of this technical presentation. The technical presentation does not necessarily reflect the official position of the American Society of Agricultural and Biological Engineers (ASABE), and its printing and distribution does not constitute an endorsement of views which may be expressed. Technical presentations are not subject to the formal peer review process by ASABE editorial committees; therefore, they are not to be presented as refereed publications. Citation of this work should state that it is from an ASABE meeting paper. EXAMPLE: Author's Last Name, Initials. 2006. Title of Presentation. ASABE Paper No. 06xxxx. St. Joseph, Mich.: ASABE. For information about securing permission to reprint or reproduce a technical presentation, please contact ASABE at [hq@asabe.org](mailto:hq@asabe.org) or 269-429-0300 (2950 Niles Road, St. Joseph, MI 49085-9659 USA).

---

## Introduction

Labor for orchard tasks remains the orchardist's largest expense (Hinman et al. 1998). Consequently, a computer vision system that successfully locates apples in an image would be beneficial for orchard automation research. This paper describes a technique to segment apple fruit in video sequences. The possible applications of such a system include robotic harvesting, precision agriculture (including the mapping of fruit distributions with GPS, or spraying), and fruit load calculations. Most past approaches investigated the apple segmentation problem with the aim of robotically harvesting fruit.

The apple segmentation problem contains some unique computer vision challenges. The environment is outdoor, which means that lighting is variable. Secondly, since the object of interest (apple fruit) is biological, there is a great deal of variety in fruit size, shape, and color. In addition, the presence of differently-colored varieties as well as the ongoing emergence of new varieties means that a successful computer vision system for this problem must be adaptable and robust. Finally, one of the largest problems is that of occlusion, including occlusion of the fruit by leaves and branches of the tree, in addition to occlusion by other apples and by trellis poles and wires.

Throughout this paper, the objects of interest, namely the fruit, are termed "foreground," and everything else in a scene, such as the trellis poles and wires, the harvester frame, leaves, and branches is referred to as "background."

Past approaches to apple segmentation in an orchard aimed to develop decision systems that select pixels based on the color, near infrared (NIR), texture, and the shape of the fruit. All of these approaches are a form of foreground modeling. However, the goal in this work is to segment fruit by modeling the background's color (in RGB) properties. Background modeling requires fewer user-supplied parameters than foreground modeling, and it allows the same model to be used for testing many differently colored apple varieties, is more robust, and is able to adjust to varying illumination levels by self-initialization. The self-initialization process learns a new background model with a video sequence containing background elements. From this information, all the parameters of the model are set. The use of video sequences in this work allows for the possibility of performing automation on-the-go, without start and stop periods for image acquisition. The harvester frame's semi-static background, artificial lighting, and the self-initialization process ensure a successful operation in many different outdoor conditions without specialized knowledge on the part of the operator.

## Fruit Segmentation

A survey of state-of-the-art processes of locating all fruit, not just apples, is presented in Jiménez et al. (2000). They divide past research on detecting fruit into two categories: local analysis (intensity and color information on the desired object) and shape analysis (fitting of circles or ellipses). They conclude that shape analysis is more robust, while local analysis is faster. They also conclude that the major problems any system must overcome are shadows, lack of depth information, and confusing regions (such as light shining through a canopy). Their solution to these problems is to use range sensors and shape analysis for oranges. However, even this approach does not totally overcome the another significant challenge, which is the occlusion of fruit by leaves, branches, trellis poles, and even by other fruit.

There have not been very many published reports of research into this problem for apple since the excellent review of the topic by Jiménez et al. (2000), but those found are summarized here. Bulanon et al. (2001) use luminance and color difference transformations of RGB color to

recognize apple fruit in images. Stanjanko et al. (2004) apply thermal imaging to detect apples in the late afternoon for the purposes of calculating the fruit load. Zhao et al. (2005) designed a system to locate fruit in single images, as a precursor to a stereo-vision system for robotic harvesting. They used a combination of redness index ( $r = 3R - (G + B)$ ), texture-based edge detection, and circle fitting in RGB color. No quantitative results were given, but both green and red fruit were recognized.

## Background Modeling

Background modeling (also referred to as background subtraction) is a technique used for motion detection (see the survey by Wang et al. 2003). The simplest version of background subtraction involves the assumption that each pixel in a static scene can be modeled by a unimodal Gaussian distribution to account for camera and scene noise. This distribution is computed by acquiring motion-free video frames and computing the mean  $\mu$  (RGB vector) and the standard deviation  $\sigma$  (scalar) of each pixel in the image space. When test images are presented, the empirical rule of Gaussian distributions is employed to determine the background or foreground class membership. In the following equation,  $X$  is a test pixel,  $\mu$  and  $\sigma$  are the stored mean and the standard deviation at that pixel,  $c$  is a constant that typically ranges between 2 and 3:

$$|X - \mu| \leq c \cdot \sigma \quad (1)$$

If the statement above is true, the pixel is counted as background. Otherwise, pixel  $X$  is classified as foreground.

A well-known and frequently-used development of a scene's unimodal Gaussian modeling is the mixture of Gaussians (MOG) technique, as presented by Stauffer and Grimson (2000). In MOG, each pixel is modeled as a mixture of  $k$  Gaussian distributions, where  $k$  is typically 3-5 for motion detection purposes. Each distribution contains three attributes: the mean  $\mu$  (RGB vector), the variance  $\sigma^2$  (scalar), and the weight  $\omega$  (scalar), which are sorted in a descending order of  $\omega/\sigma$ . The advantage of using the MOG technique for motion detection is that a dynamic background can then be modeled, such as waving trees or water. Figure 1 illustrates the MOG background model. The model is represented as a cuboid, with 8 columns, 3 rows, and 4 distributions per pixel ( $k$ ). A new pixel  $X$  is tested for foreground or background class membership by determining if a match exists with the current distributions (i.e. Equation (1) is true for any distribution in the model). If no distribution matches  $X$ , then that pixel is classified as foreground. Conversely, as soon as a match is found, the search is terminated and labeled as background. Consequently, the worst-case running time for this pixel occurs when a motion pixel is encountered.

More formally, a test pixel  $X$  is considered to belong to the background class if it is a member of the set  $B$ :

$$B = \left\{ (X, \mu, \sigma^2) \mid X \in RGB, \mu \in RGB, \sigma^2 \in \mathfrak{R}, c = 2.5, n \in [1, k], |X - \mu_n| \leq c \sqrt{\sigma_n^2} \right\} \quad (2)$$

And the foreground class  $F$  is described as

$$F = \bar{B} . \quad (3)$$

## The Global Mixture of Gaussians (GMOG) Method

Although the application of the MOG method to the apple segmentation problem initially produced promising results in preliminary tests, a drawback of the MOG method for the apple segmentation problem is that there is a high degree of redundant data storage. For apple segmentation in video, the number of distributions  $k$  was set to 20-30, which means that when using 640 by 480 pixel images, the MOG data cube in Figure 3 stores  $640 * 480 * 30 = 9,216,000$  separate Gaussian distributions. Besides the storage requirements, much of the data was repeated within the cube for the apple segmentation problem. The large  $k$  also resulted in slow run times. Instead of using the MOG per-pixel model, the GMOG method was conceived to reduce running time and storage requirements while increasing performance in repetitive, dynamic backgrounds such as apple segmentation. The GMOG method effectively describes the background colors with a multimodal Gaussian distribution for the whole image. Consequently, the data storage is reduced to the GMOG data vector, as shown on the right side of Figure 3. The number of distributions in the GMOG method is usually 60-80.

During the initialization period, the frequency of each RGB value of each pixel is recorded via a histogram  $H$  ( $256^3$  bins). With this frequency information, the multimodal Gaussian distribution of the background colors is found by way of two user-supplied parameters: a distance measure  $\varepsilon$  and a minimum frequency  $\lambda$ . The RGB color representing the greatest frequency is found in the histogram  $H$ . This color,  $\mu_0$ , represents the current hypothesis of a distribution mean. All RGB points within the distance  $\varepsilon$  of  $\mu_0$  represent the set  $S$ :

$$S = \{X \mid X \in RGB, |X - \mu_0| \leq \varepsilon\} \quad (4)$$

See Figure 4 for an illustration of  $S$ . The mean  $\mu_S$  and standard deviation  $\sigma_S$  of set  $S$  are computed based on the frequencies of these values, as found in  $H$ .  $\mu_S$  and  $\sigma_S$  identify a Gaussian distribution, which is added to the bottom of the GMOG data vector described in Figure 3. The search for a new  $\mu_0$  is then repeated. RGB colors already incorporated into the current GMOG model are not taken into consideration. The iteration process stops when the frequency of  $\mu_0$  is less than  $\lambda$ . This serves to eliminate outliers in the training data.

After the GMOG vector-generation process detailed above, the distributions are sorted in decreasing probability. The values of  $k$  for this particular application were  $k = 50-80$ . However, this large value of  $k$  is restrictive to fast testing. Notice that in Figures 5 and 6, the space described by the GMOG vector is somewhat contiguous and has an overlap between sections. By observing that the input pixels are discrete and that the GMOG model describes the whole image space, the GMOG model can be converted into a class membership lookup table. Specifically, notice that the GMOG data vector can be expressed in the following way (where  $B$  is the set of background class pixels, and  $\mu_i$  and  $\sigma_i$  are the  $i$ th distributions in the GMOG model):

$$B = \{X \mid X \in RGB, i \in [1, k], c = 2.5, |X - \mu_i| \leq c\sigma_i\} \quad (5)$$

Consequently, all RGB values in the lookup table are set to class membership according to  $B$  and  $F$ :

$$F = \overline{B} \quad (6)$$

Testing for class membership on a per-pixel basis then is reduced to looking up the class for each RGB value. The major advantage of this method is that it is very fast. The pseudocode of the GMOG method is shown in Text Box 1.

Figures 5 and 6 also indicate that the decision boundary of the background and foreground classes is not linear, i.e. the division of the two classes cannot be adequately represented with a hyper plane.

## Video Imaging of Apples

All the tests were performed at the Appalachian Fruit Research Station (USDA, ARS) orchard in Kearneysville, West Virginia during the fall of 2005. The varieties tested include 'Gold Blush' (yellow), 'Dixie Red' (red), and 'Ace Spur' (red), each on a Y-trellis. A digital camera (Sony DFW-x700 IEEE-1394 8-bit 1024x768 Color CCD camera, Tokyo, Japan) was mounted on an over-the-row harvester frame, and images were acquired at approximately two frames per second with an image size of 640x480 and a machine speed of approximately 0.482 kilometers per hour. A black curtain was placed in between the arms of the Y-trellis so that apples from the other side would not be shown. Apple fruit were removed from the first and last 2.5-3 meters of each row. Then the harvester passed over the entire row while the camera recorded the images. Each test sequence represents one row. The resulting video sequence was divided into initialization and test sets. Some of the initialization images had to be discarded because of missed fruit not apparent during the time of testing but visible in the images. As a result, no decisions were made as to what trees should form the initialization and testing sets other than their location in the row.

Table 1 describes the video sequences, including the number of frames used for initialization and testing, as well as the ambient lighting conditions on the day of acquisition. All sequences were collected with artificial lighting consisting of two halogen and two incandescent lamps. An illustration of the view from the camera is presented in Figure 2. The results were generated by

testing each video sequence off-line on a laptop computer with 1.5GHz processor.

### Text Box 1. Algorithm for GMOG

#### Parameters:

$\epsilon = 30$ ;  $\lambda = 10$ ;  $c = 2.5$ .

$X$  is a test value.

$B$  and  $F$  are sets of RGB values describing the background and foreground classes, respectively.

#### Initialization

1. Generate a histogram  $H$  (256x256x256) of RGB values using an apple-free initialization sequence.
2. The background model  $B = \{\}$ .
3. Find the most probable RGB value  $\mu_0$  not already in  $B$  by running a search through the histogram.
4. **If** the frequency of  $\mu_0 \geq \lambda$ :
  4. Set  $S =$  all RGB values whose distance to  $\mu_0$  is  $\leq \epsilon$ .
  6. Mean and standard deviation of  $S = \mu_S$  and  $\sigma_S$ .
  5. Set  $B' = \{X | X \in RGB, c = 2.5, |X - \mu_S| \leq c \sigma_S\}$ .
  6.  $B = B' \cup B$ .
7. **endif**
8. Loop back to line 3 until (frequency of  $\mu_0 \geq \lambda$ ) = **false**.
8. Generate a lookup table for each RGB value and set to class memberships according to  $B$  and  $F$  ( $F = \bar{B}$ ).

#### Testing

1. Look up the RGB value for each test pixel to determine class membership.
2. Size filter the resulting image.

## Results

The results were generated by running the algorithm off-line on four separate video sequences. A size filter of 40 pixels was applied to the result. The binary result image was saved during the running of the algorithm for further analysis. Ground truths were generated in order to quantify the results. Sample image results are given in Figure 7. Videos of the four sequences and the results can be found at

<http://rvl1.ecn.purdue.edu/RVL/Projects/AppleSegmentation/>.

### ***Ground Truth Generation***

In order to quantify the results, ten images were selected via a random-number generator from the set of test images. The location of apples in these selected images was manually marked, and these marked images form the ground truth set. In order to mark the images, an ellipsoid was used to contain the approximate apple location in the image plane, regardless if partial occlusion was present or not. See figure 6 for a test image and for some of the resulting ground truths. Separate files were created for each apple since this work focused on determining how many apples can be correctly detected, even in the presence of partial occlusion.

### ***Quantitative Analysis***

A blob from the result set is determined to correctly identify an apple in the ground-truth set if the blob identifies 5% of the apple's area in the image plane, even in the presence of partial occlusion as mentioned above. Table 2 shows the results. The percentage of correctly identified apples was very high, from 85.6% to 95.6%. Red apples were successfully detected, as expected, as well as yellow apples, at a rate of 93.04%. False positives were reasonable, at 1.2 to 5.3 per image.

### ***Qualitative Analysis***

While one of the final goals of this project is to reduce the negative effect of occlusion by using video and a moving camera, in this preliminary paper the goal is to correctly identify apple fruit in images with few false positives and establish a sufficiently fast running-time algorithm that could be applicable to real-time data collection and testing. These goals have been realized, as even partially occluded fruit was identified and false positives were low. In addition, the run-time of the algorithm was sufficiently high, at 14-16 frames per second, qualifying this work to be suitable for real-time video application and allowing room for processing these results further.

### **Conclusions and Future Directions**

It has been shown that GMOG is a fast, accurate method for apple segmentation on the tree. The GMOG method requires little operator input or fine-tuning by variety, compared to past approaches to this problem. The use of video for this problem provides possibilities for fast orchard automation in the future.

Future directions include tracking the location of the apple fruit as the camera moves in order to confirm or deny the presence of apple fruit as well as fruit singularization.

### ***Acknowledgements***

Many thanks go to William Anger and Scott Wolford for collecting data in West Virginia at the Appalachian Fruit Research Station orchard, as well as to the Robot Vision Lab at Purdue for lab space.

### **References**

Bulanon, DM, T. Kataoka, S. Zhang, Y. Ota, T. Hiroma. Optimal thresholding for the automatic recognition of apple fruits. 2001. ASAE Paper No. 01-3133. St. Joseph, Mich.:ASAE.

- Hinman, H., K. Williams, D. Faubion. Planting and production costs of high density Fuji. 1998. Agr. Res. Center Pub. EB1878, Washington State University, Pullman, WA. 52 pp.
- Jiménez, A.R., R. Ceres, J.L. Pons. A survey of computer vision methods for locating fruit on trees. 2000. *Trans. ASAE* 43(6): 1911-1920.
- Stanjko, D., M. Lakota, M. Hočevár. Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. 2004. *Computers and Electronics in Agr.* 42: 31-42.
- Stauffer, C. and W. Grimson. Learning patterns of activity using real-time tracking. 2000. *IEEE Trans. PAMI* 22(8): 747-757.
- Wang, L., W. Hu, T. Tan. Recent Developments in Human Motion Analysis. 2003. *Pattern Recognition* 36: 585-601.
- Zhao, J., J. Tow, J. Katupitiya. On-tree fruit recognition using texture properties and color data. 2005. *IEEE/RSJ Int. Conf. Intell. Robots and Systems*.

## Tables and Figures

Table 1. Description of the video sequences to test GMOG.

Sequence Name	Variety	Number of Initialization Images	Number of Test Images	Date	Ambient Lighting
Seq. 1	Dixie Red	200	650	11 October 2005	
Seq. 2	Dixie Red	92	613	13 September 2005	
Seq. 3	Ace Spur	85	680	11 October 2005	
Seq. 4	Gold Blush	91	427	5 October 2005	

Table 2. Quantitative results for GMOG as compared to 10 ground truth (GT) images.

Sequence name	Total apples in	Correctly identified	False positive	Average area false	Frames per second for
---------------	-----------------	----------------------	----------------	--------------------	-----------------------

	10 GT images	apples (%)	blobs/ image	positive blob	testing
Seq. 1	113	95.58%	1.2	64.67	15.11 fps
Seq. 2	188	85.64%	5.3	85.64	15.86 fps
Seq. 3	213	94.84%	4.3	113.09	14.06 fps
Seq. 4	158	93.04%	5	110.28	14.78 fps

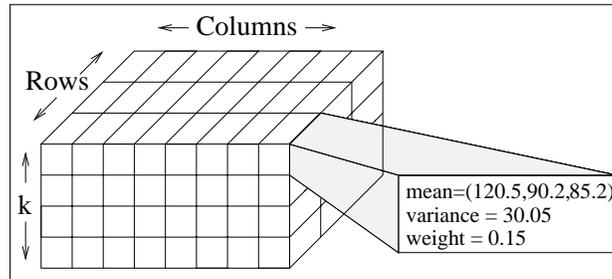


Figure 1. Illustration of the MOG background model. The background model contains Rows \* Columns \* k Gaussian distributions, each containing a mean, variance, and standard deviation.



Figure 2. View from the camera of the harvester background (left) and the harvester background with Dixie Red apple tree (right).

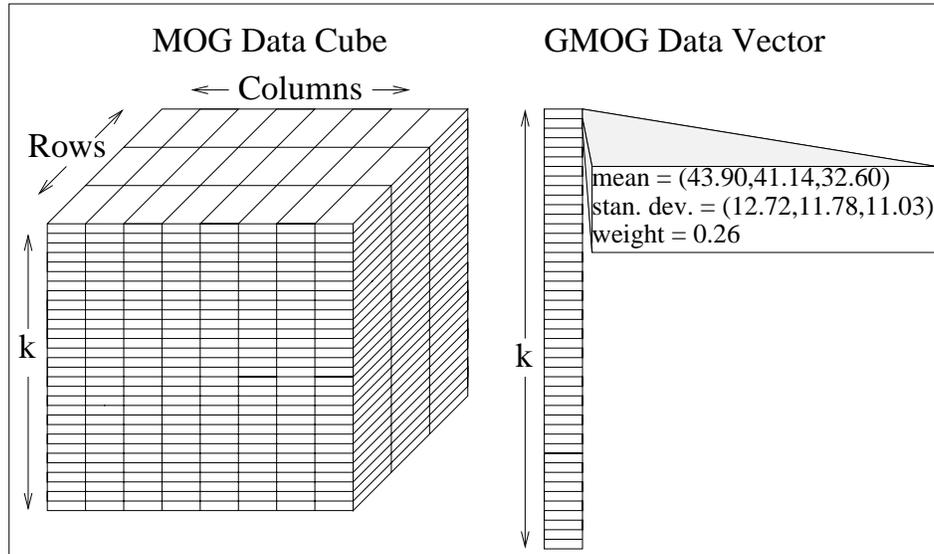


Figure 3. Comparison of the MOG background model and the GMOG background model for apple segmentation

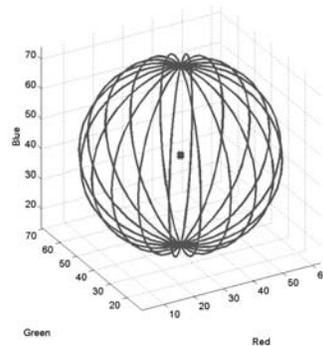


Figure 4. Illustration of the sphere bounding the search for distributions to match  $\mu_0$ .  $\mu_0$  is in the middle of the sphere and the boundary and interior of the sphere represent  $S$ .

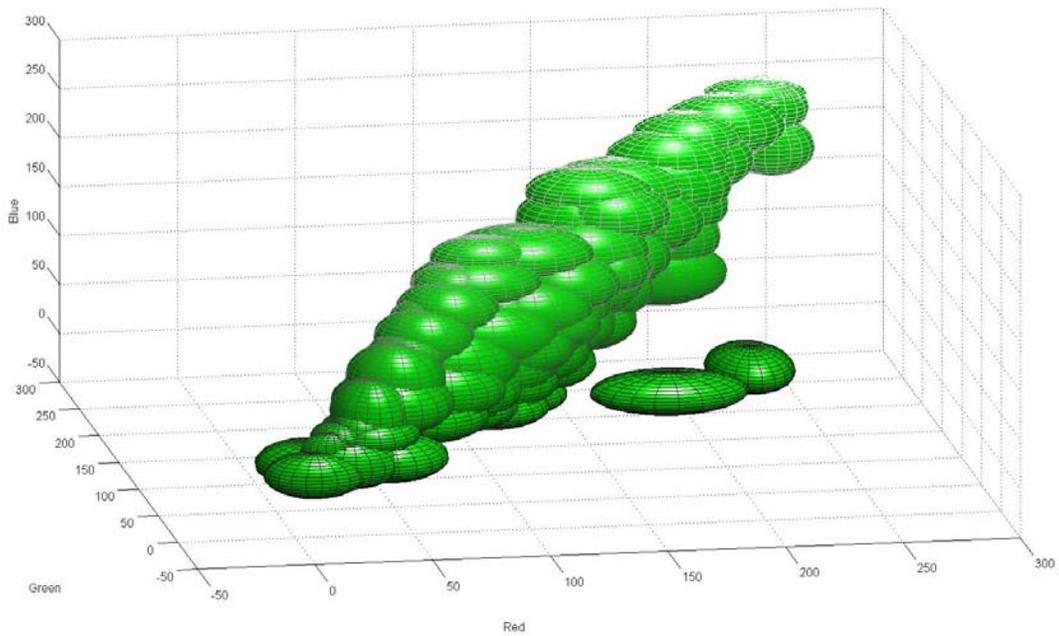


Figure 5. Illustration of the GMOG model space for Seq. 1. An ellipsoid with axes equal to the constant  $c (2.5) * \text{standard deviation}$  are drawn, with the center of these ellipsoids as the mean. This is done for all distributions. The RGB points contained in the ellipsoids represent the background class.

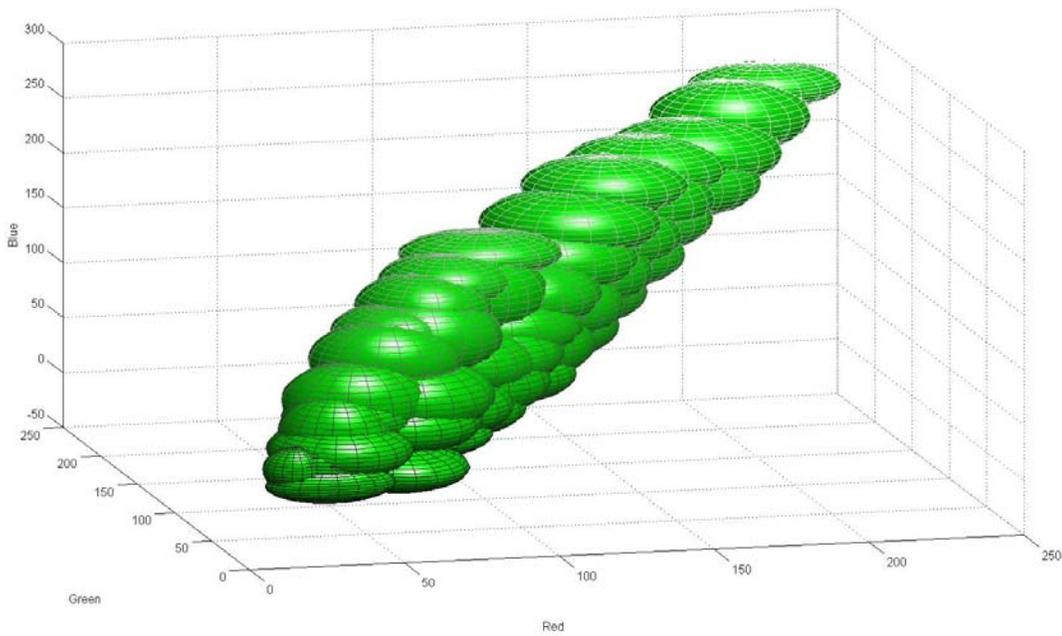


Figure 6. Illustration of the GMOG model space for Seq. 4.

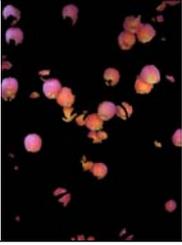
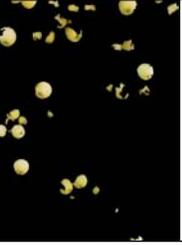
			
Test image for Seq. 1	Result of GMOG	Test image for Seq. 2	Result of GMOG
			
Test image for Seq. 3	Result of GMOG	Test image for Seq. 4	Result of GMOG

Figure 7. Examples of GMOG results.

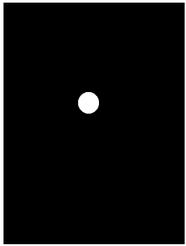
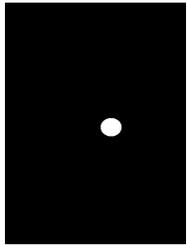
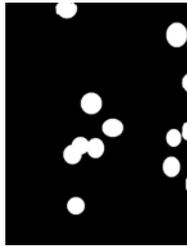
			
Test Image	Location of first apple in ground truth image	Location of second apple in ground truth image	Location of all apples in the test image

Figure 8. Example of Ground Truth generation, Dixie Red apples.