Using Image Processing and Data Mining Techniques for the Automated Detection of *Drosophila* Embryo Gene Expression Patterns

Joseph May

Pine Crest School
University of Miami Data Mining, Database & Multimedia Research Group

Abstract:
This study employs image processing and data mining techniques for the development of *Drosophila* embryo-specific image-analysis software. In the initial stage of this research, the most effective method of segmentation for pre-processing of *Drosophila* embryo images was investigated. Several edge-detection based segmentation protocols were compared, demonstrating that Canny edge detection produced more favorable results than Sobel edge detection. After testing four different parameter sets, it was determined that the most accurate edge detection results were produced with a low Gaussian filter sigma value and gradient magnitude based threshold values. A bounding box was applied and resulted in an average segmentation success rate of 88.5%. In the second phase of this research, a visual dictionary, which can be used to identify the presence of specific gene expression patterns, was developed from extracted features. First, the SIFT (Scale-Invariant Feature Transform) algorithm was used to extract a feature matrix from each image. Next, a k-means clustering algorithm was run on a 10% subsample of the identified features, with the k number of means selected to be 10% of the subsampled features. The results show that a k-means clustering of a 10% subsample of identified features can be effectively used to develop a bag-of-words visual dictionary. Methods developed here for the accurate analysis of *Drosophila* embryogenesis images may help improve the speed of gene expression pattern studies using images gathered by the Berkeley Drosophila Genome Project.

Introduction

The genus *Drosophila*, specifically *Drosophila melanogaster*, is one of the most commonly used model organisms in the field of Biology. Because of genetic, historical, and behavioral traits, *Drosophila melanogaster* is well suited for use in both genomic and embryogenetic (developmental) studies. Though much of the genetic analysis of *Drosophila melanogaster* has been automated through bioinformatics software, embryogenetic analysis is still often conducted by hand. For large studies, this can be disadvantageous as it may be time consuming and prone to human error.

Recent advances in available computing power have led to the emergence of high content screening (HCS).[1] This technique uses large-scale image analysis to determine the effects of an environmental change on a living organism. HCS is most often used to identify new drug candidates, because it allows many chemical compounds to be tested on a large number of organisms in a short period of time. Though the image analysis could technically be performed by hand, HCS studies often produce images that, combined, contain hundreds of thousands of cells. Most often, automated computerized image analysis is performed.[1]

Through The Model Organism ENCyclopedia Of DNA Elements (modENCODE) project by the National Institute of Health and the US National Human Genome Research Institute, there is an ongoing effort to identify all of the functional genes in the *Drosophila melanogaster* genome. The Berkeley Drosophila Genome Project has, as part of modENCODE, amassed a large database of images containing gene expression patterns during embryogenesis. Tomancak at al. of The Berkeley Drosophila Genome Project used microarray analysis and *in situ* hybridization to label and observe the spatial patterns of mRNA expression.[9] The annotation of

these expression patterns has previously been conducted by hand, however annotating by hand is no longer practical as the database now contains over 100,000 images, many of which contain multiple embryos.[2] Research using the data from The Berkeley Drosophila Genome Project, in addition to future studies employing *Drosophila melanogaster* gene expression pattern imaging, would benefit from software that has been specifically designed for the analysis of these images.

Although multiple research groups have previously attempted to solve this problem, there is still no publically available open-source framework for *Drosophila melanogaster* gene expression pattern image analysis. Jieping Ye at al. of Arizona State University have developed the FlyExpress program with a 78% gene expression pattern identification success rate[16], however their research relied on manually pre-processed images. By automating the entire process, this study aims to improve the speed and accuracy of gene expression pattern identification.

Segmentation

Before features can be extracted to identify gene expression patterns, images must first be pre-processed. Pre-processing allows image analysis to be performed on standardized images that are easily comparable with each other. Because the Berkeley Drosophila Genome Project images often contain large amounts of "empty space" surrounding the embryo, segmentation is a necessary pre-processing step. Not to be confused with biological segmentation, image-processing segmentation is the process of separately identifying the object of interest (in this case the embryo) from the background. Segmentation is an important process in automated image analysis because if the cell is not properly identified, later image processing may be applied to the wrong part of the image (i.e. elements not in the embryo). Previous research on HCS studies

has shown that improper segmentation can lead to improper image recognition results.[4]

Image segmentation can be accomplished in many different ways.[6] Most popular in biological research are edge-detection based methods and K-means clustering based methods.[5] In the first phase of this research, various edge-detection image segmentation methods of *Drosophila melanogaster* embryos in multiple development stages are investigated.

Feature Detection

The most accurate way for computers to identify and associate the type of visual patterns that are easily recognized by humans is to algorithmically isolate features (points of interest) and use machine learning to connect and trend these feature points. Although countless methods have been developed for isolating and describing these features, the "gold standard" is Scale-Invariant Feature Transform, or SIFT. First proposed by David Lowe in 1999, this patented method has the advantage of being able to describe features in a way that they can be matched even when objects are scaled, skewed, or partly occluded. [12]

SIFT extracts features by first applying Gaussian filters with increasing sigmas in subsequent octave scale spaces. After the Gaussian filter has been applied five times in an octave, the next octave is created by subsampling the original image. This process is completed two more times for a total of four octaves. Next, each Gaussian-filtered image is subtracted from the previous Gaussian-filtered image to create four differences of Gaussians per each octave (in this case approximately equivalent to the Laplacian of Gaussian). For the selection of feature keypoints, each pixel in each difference of Gaussian layer is compared to its 28 immediate neighbors (each surrounding pixel, plus each surrounding pixel in the difference of Gaussian layer above and below it). Any pixel that is the maximum or minimum of the surrounding pixels

is chosen as a keypoint. These pixels do not directly correspond to the pixels of the original image; rather they are pixels in the higher to lower resolution difference of Gaussian layers. This technique identifies features invariant to scale because each scale space in each octave is blurrier (because of the applied Gaussian filter) than the one below it and equivalent to an image taken at a further distance or lower resolution. [12]

Once keypoints have been chosen, various techniques are used to remove noise before the orientation and descriptor vectors are created. In order to be invariant to rotation, the orientation gradient magnitude and direction are found. This is calculated by creating a directional histogram of surrounding points, and weighting it by the gradient magnitude. The largest resultant is chosen and applied to that keypoint. Finally, the 128 dimensional descriptor vector is created by finding the orientation of the sixteen sixteen-pixel blocks surrounding the keypoint.[12]

Machine Learning

Both supervised and unsupervised machine learning algorithms are useful for image processing. The advantage of unsupervised algorithms is that they do not require a training set and can often better adapt to unfamiliar conditions. On the other hand, supervised algorithms, when trained properly, will often produce more accurate results and can be later applied to individual images.

Bag-of-words is a commonly used image processing feature classification technique that was adapted from textual data mining. This technique uses a vector "dictionary" of feature keypoints to identify which images are most likely of a related nature. After features are extracted from an image of interest, each feature is matched to its nearest word in the dictionary. A histogram is constructed to show the most common words in the image, and "sentences" of the

visual words are compared to identify what objects of interest (in this case gene expression patterns) are present in the image.[17] The advantages of the bag-of-words approach are that it is both simple and relatively robust. The main disadvantage is that it does not take into account spatial information, an important consideration when evaluating images.[16]

In order to construct the dictionary (or codebook) for the bag-of-words classification, k-means clustering, a simple unsupervised algorithm, was chosen. In k-means clustering, k "means" (reference points) are randomly placed within the dimensional space of the data. Each data point is mapped to the nearest mean, and then the mean is moved to the centroid of the associated points. This process repeats itself until no data point moves to another mean point.

Materials & Methods

1. Retrieval of Images

To remotely retrieve the images off of The Berkeley Drosophila Genome Project server, a Java program was written to call Wget for each image. To install Wget on the OSX system used for development, the Unix dev. tools and Apple Xcode were first installed from the Apple App Store. The Wget source file was downloaded using the cURL Terminal command. The file was configured with Open SSL, the source was built, and Wget was installed.

To download the images, a CSV file was first obtained containing the URLs of all of the images. A Java program was written to loop through the CSV file and call Wget for each URL. A number was recorded after each subsequent download to track the completion of downloads. Because of network connectivity issues and time/resource constraints, slightly less than half of the image set was downloaded.

Part A (Segmentation)

A1. Creating Data Sets for Testing

The Berkeley Drosophila Genome project classifies *Drosophila melanogaster* embryogenesis into six distinct developmental stages. To test the efficacy of segmentation and classification methods, a data set was created for each of these stages. The six data sets each contain approximately 100 images, however this varies as certain images had to later be removed. The criteria for choosing each image were low resolution, lateral embryo view, and the image being focused on a single embryo.

A2. Segmentation

Segmentation of *D. melanogaster* embryos was conducted using Matlab. The protocol used was a modified version of the *Detecting a Cell Using Image Segmentation* protocol, published on the Mathworks website.[19]

Initially, an unmodified version of the protocol was applied to a sample image to observe the results. This included converting the image to grayscale, applying a Sobel edge detection algorithm, dilating the image, infilling the image, diamond-eroding the image, and placing an outline of the resultant image over the original image. The same protocol was also attempted with a Canny edge detection algorithm being used instead of a Sobel edge detection algorithm.

The protocol was then applied to the first developmental stage image set. A program was created to loop through the data set and apply the segmentation protocol to it. This was originally accomplished using the Sobel edge detection algorithm. This was repeated again using Canny

edge detection with the default parameters. The protocol using Canny edge detection with the default parameters was applied to both the first and second developmental stage data sets.

To determine the parameter values to use for the Canny edge detection algorithm, two sample images were chosen at random, and the protocol was applied using various threshold and sigma values.

Once possible ideal threshold values and two possible sigma values were found, the protocol using segmentation with each of these values was applied to the first and second developmental stage image set.

Finally, a protocol for determining the tone of an image was created (see below), which was applied to the images using a program that segmented the image with a Canny sigma of 14 if the image had a dark tone or segmented the image with a Canny sigma of 30 if it had a lighter tone. The first stage and second stage data sets were run through this program.

A3. Determining The Tone of an Image

To determine whether an embryo had a light or a dark tone, an image was first cropped to a 160x160 pixel image that only contained the inside of the embryo. This image was then converted from a RGB to a HSV color scheme, which allowed the individual brightness of pixels to be measured. The brightness of all of the cropped images was then averaged, and if the average was less than .3, the embryo was considered light-toned, and if the average was above, .3, the embryo was considered dark-toned.

A4. Placing a Bounding Box Over the Segmented Embryo

To remove noise visible on direct segmentations displays, a program was written to draw

a bounding box over the segmented embryo. First, the size of each segmented area was measured, and they were compared to determine the largest segmented area. Next, the corner coordinates of the segmented area were determined and a bounding box was drawn from those coordinates.

A5. Determining Whether an Embryo Has Been Segmented

To determine whether an image of an embryo was successfully segmented, each image was observed, and the researcher noted whether the segmentation reasonably matched the boundary of the embryo and the background.

To determine whether an image was well segmented, images that had been previously marked as successfully segmented were listed, and images that had a double segmentation, large amount of noise, or shifted segmentation were then removed from this listing.

Part B (Feature Detection and Clustering)

B1.Installing Matlab Toolbox

In order to extract Scale-Invariant Feature Transform (SIFT) features for use in a feature vector, the VLFeat open source Matlab toolbox (Matlab implementation of a C library) was installed. This toolbox was installed using the protocol on the VLFeat website (vlfeat.org), however the root directory was modified.

B2.Extracting SIFT Features

SIFT features were extracted by calling the SIFT algorithm in Matlab from the VLFeat toolbox. The algorithm was applied to all pre-processed images in each of the six developmental stages from the data set created in part A1. Before SIFT features were extracted, the images were

pre-processed by cropping them to the segmentation box as specified in part A. A program was written to iterate through the data set for each developmental stage, extract the SIFT features, and combine both VLFeat outputs (the location/direction/magnitude vector and the descriptor vector) into a single feature vector, which was exported as a CSV file.

B3.Clustering for Gene Expression Pattern identification

A program was written to retrieve and combine the feature vectors of each image from a single developmental stage. Because the computer used for this research was not powerful enough to cluster the full matrix of data points produced by all of the images in each developmental stage, a subsample of every tenth keypoint was taken (resulting in $1/10^{th}$ the number of features being clustered as detected). Next, the k-means function from the Matlab Statistics Toolbox was run, with the default options changed to display the results of each iteration and to allow for a maximum iteration of 1000. The K number of means (clusters) used for each was set equivalent 10% of the number features clustered. This process was repeated for all six developmental stages.

Results

Part A (Segmentation)

The initial segmentation testing was conducted with two sample images. When the edge-detection results were compared, the Canny edge detection algorithm appeared to provide significantly better recognition than the Sobel edge detection algorithm. This was the expected result, as the Canny algorithm is considered the standard of edge detection.[7]

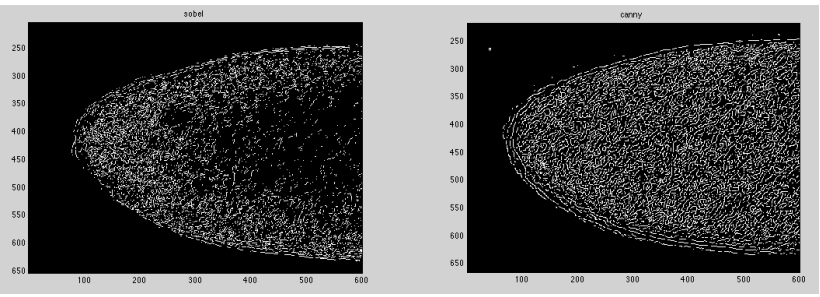

Figure 1: A comparison of the edge-detection results of a Sobel edge detection algorithm (left) and a Canny edge detection algorithm (right). Both images were created using default parameters and are magnified to show detail.[9]

Though the initial comparison demonstrated that Canny was the superior algorithm, these results were verified by applying the full segmentation protocol to the first developmental stage data set using the Sobel algorithm. Though a memory-usage error prevented the completion of this test, the results were so poor that it was not repeated.

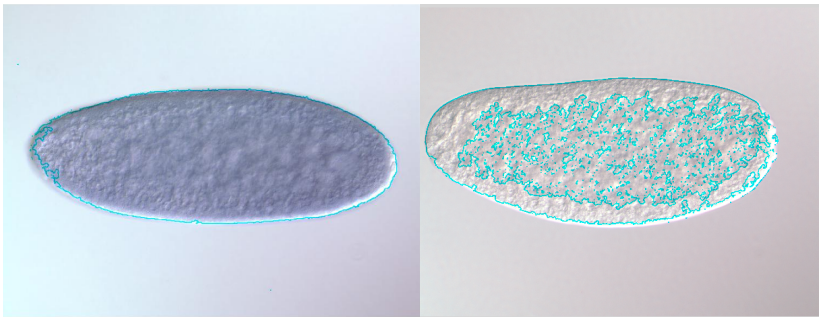

Figure 2: An example of a successful (left) and an unsuccessful (right) segmentation using the Sobel algorithm[9]

The first developmental stage data set was segmented using the Canny algorithm with default parameters, and the images were saved and analyzed. Though most embryos were successfully segmented, several images contained a large amount of noise, double segmentation, or non-embryo segmented areas. In addition, the segmentation edge of each embryo was very rough and often excluded small parts of the embryo. (Figure 3)

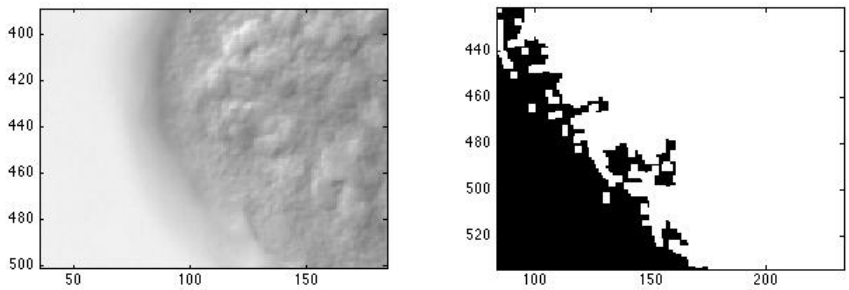

Figure 3: A close up of a "rough edge" result of Canny-based segmentation (right) and a close up of the original image (left)[9]

To attempt to improve on the results of the segmentation using the Canny algorithm with default parameters, especially the edges and double segmentation, the first developmental stage data set was segmented again using specific parameters. The two parameters that can be modified in Canny edge detection are threshold values and the sigma of the Gaussian filter. When detecting edges, the Canny algorithm includes all edges above the upper threshold value and only edges between the two threshold values that are adjacent to larger edges. The sigma of the Gaussian filter specifies the intensity of the Gaussian filter applied to the image before edge detection. The purpose of the Gaussian filter in edge detection is to "smooth out" the image and remove any subtleties that would be incorrectly interpreted as edges. A larger sigma of the Gaussian filter should result in less noise and smoother edges, but also fewer edges and edges at less exact positions. Two test parameter sets were chosen, both with a threshold range of .0 –.1 and with sigma values of 14 and 30 (see materials and methods for more details).

For the first developmental stage, the results for the sigma value of 14 were extremely promising. Embryos that were successfully segmented all had single-segmentation (versus the common double line for the default parameters), generally were low in noise, only segmented actual embryos, and had smooth edges. On the other hand, only certain embryos were segmented at all, and the embryos that were successfully segmented usually had darker interior colors that led to higher embryo contrast. Unsegmented embryos were usually ones with lower contrast. Lower contrast embryos that were segmented often had rougher segmentation edges or less perfect edges than other embryos.

By contrast, many more embryos were successfully segmented with a sigma of 30. However, these segmentations were less exact and often had a small fraction of the bottom of the embryo missing from the segmentation, with a sliver of the background added. These results are

logical, as a higher sigma of the Gaussian function leads to a blurrier image being processed by the Canny algorithm.

A program was run on the first developmental stage image set that determined the tone of the embryo images and then applied Canny edge detection with a variable sigma of either 14 or 30 to the image. As expected, the results were a larger number of embryos segmented than with a sigma of 14, and better segmentation on images with higher levels of contrast than with a sigma of 30.

The percentage of success for each segmentation method was calculated. These percentages only address whether an embryo was segmented; they do not address the quality of that segmentation. Using the default parameters, 97% of embryos were segmented successfully. Using a Gaussian filter sigma of 14, 70% of embryos were segmented. Using a Gaussian filter sigma of 30, 89% of embryos were segmented. Finally, the program that varies the Gaussian filter sigma depending on the tone of the embryo resulted in a segmentation rate of 78%.

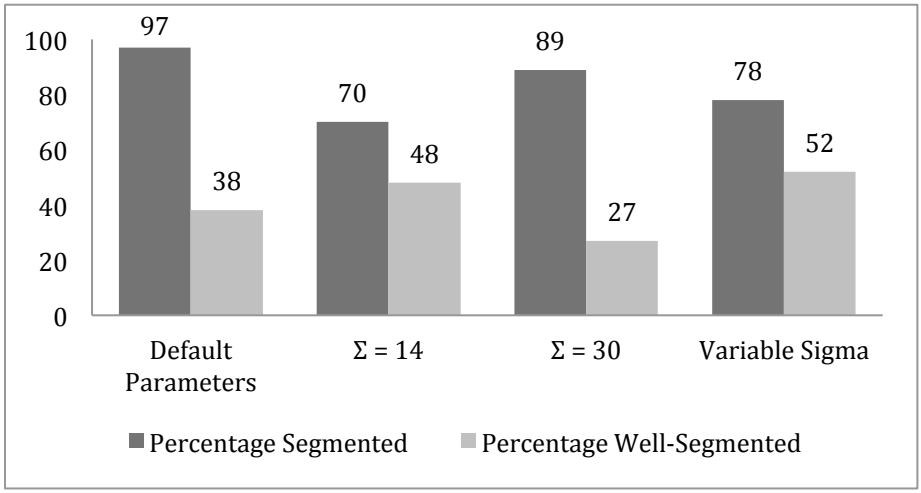

Figure 4: This graph shows the successfulness of each method of segmentation. Because the segmentation accuracy was analyzed manually, it is subject to human error and bias.

The second developmental stage data set was then segmented using the Canny algorithm with each of the four-parameter sets. With the exception of segmentation using default parameters, the results were strikingly different than the results for the first developmental stage data set.

Most of the embryos were successfully segmented using the default parameters. Like the first developmental stage data set, the segmented edges of many embryos were rough and did not exactly match the actual edges of the embryos. In addition there were many embryos that had multiple segmentation lines. This was less common in the second developmental stage data set than the first, however. Noise, though still present, was much less common in the second developmental stage data set. Many images not considered successfully segmented had segmentation outlines that were accurate but included multiple embryos.

The results of the embryos segmented using a Gaussian sigma of 14 were extremely poor. Almost none of the embryos were successfully segmented, and many that were segmented were missing small pieces of the embryo.

Though still poor, the quality of the segmentation using a Gaussian sigma of 30 was much better than that using a sigma of 14. Close to twice as many embryos were segmented with the higher sigma, though many embryos were on the borderline between properly and improperly segmented. Though there were many embryos with no segmentation at all, a more common problem with this method was under-segmentation.

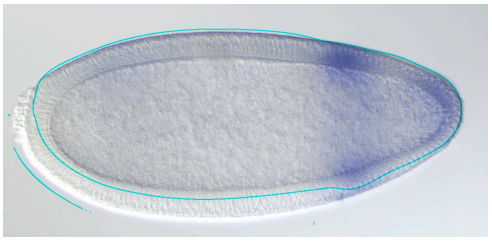

Figure 5: An example of an under-segmented embryo.[9]

Because the variable sigma segmentation used the same parameters as the either the sigma values 30 or 14 tests, results could not improve upon those two tests. As expected, the results of the variable sigma segmentation were very poor.

The percentage of success for each segmentation method was again calculated for the second developmental stage data set. These percentages only address whether an embryo was

segmented; they do not address the quality of that segmentation. Using the default parameters, 85% of embryos were segmented successfully. Using a Gaussian filter sigma of 14, 16% of embryos were segmented properly. Using a Gaussian filter sigma of 30, 30% of embryos were segmented. Finally, the program that varies the Gaussian filter sigma dependent of the tone of the embryo resulted in a segmentation rate of 19%.

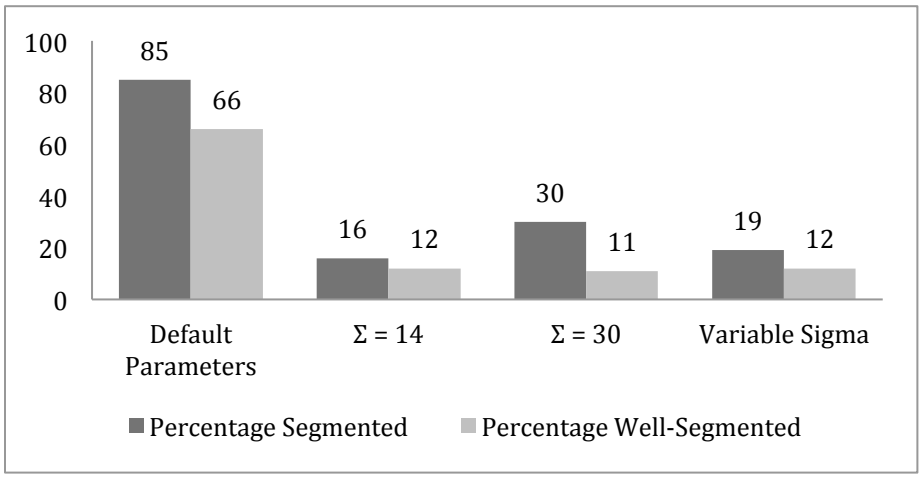

Figure 6: This graph shows the successfulness of each method of segmentation. Because the segmentation accuracy was analyzed manually, it is subject to human error and bias.

To determine if the quality of segmentation was affected by the threshold values, various threshold ranges were tested with both the default and the modified-Gaussian-sigma segmentation protocols. In all situations, there was no significant differences between the segmentation results when the threshold value was set at [.0 –.1] (the values used for the previous modified-Gaussian-sigma tests) as compared to when it was set at the default value, which is calculated by determining the gradient magnitude of the image

All six developmental stage data sets were segmented with Canny edge detection using the default parameters, and a bounding box was applied to each image to determine the accuracy of the segmentation once the factors of noise and double segmentation are removed. (See materials and methods for more details) There was very little variation between the results of each data set. All data sets had a correct segmentation percentage above 80%, and the average success rate was 88.5.

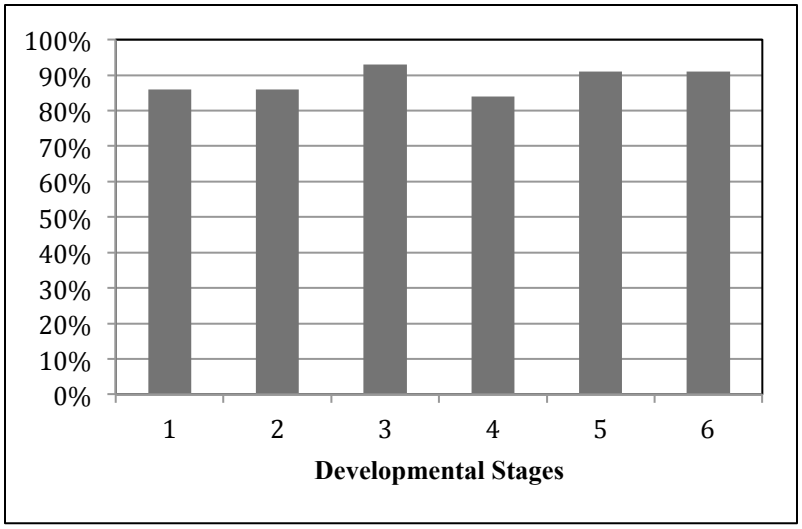

Figure 7: This graph shows the percentage of images in each developmental stage that had a properly placed bounding box after segmentation. Because the segmentation accuracy was analyzed manually, it is subject to human error and bias.

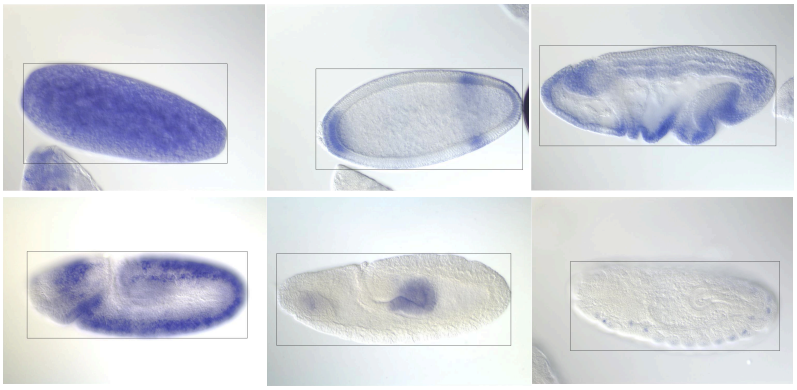

Figure 8: Examples of successful segmentation results with a bounding box from each developmental stage (top left – stage 1, bottom right – stage 6)[9]

Part B (Feature Detection and Clustering)

SIFT was applied to each image in each dataset. In total, 1.88 million features were identified. That averages out to 314 thousand features per developmental stage and around 3300 features identified per image. It is interesting to note that there were significantly fewer features identified in the developmental stage five and six datasets, though the lower number of features identified in stage six may be accounted for by the fact that there were fewer images in that data set. Each identified feature was signified by a 132 dimensional matrix, containing the coordinates of the points, the original gradient magnitude histogram data, and the 128 dimensional matrix created by the SIFT classifier (see background for more details).

K-means clustering was originally going to be run on the entire feature dataset for each developmental stage. However, this was not feasible because the computer used was not able to cluster data of that size. To get around this limitation without significantly degrading the quality of the results, the datasets were subsampled to contain 10% of the original features. Therefor, the average number of features clustered for each developmental stage was 31 thousand.

The number of clusters (K means) chosen was 1% of the feature datapoints. Although this may appear to be arbitrary, it provides a somewhat optimal tradeoff between accuracy and time. More accurate results could have been achieved by setting the number clusters to maximize the Bayesian Information Criterion[18], however this would have required time and computer resources beyond what was available for this research.

To determine the fitness of the clustered data for use in bag-of-words feature classification, a histogram was created for each developmental stage, showing the frequency of features per cluster. As the frequency was varied and most clusters contained a number of features in a reasonable range (around the mean of 100), the clustering appears to have been successful.

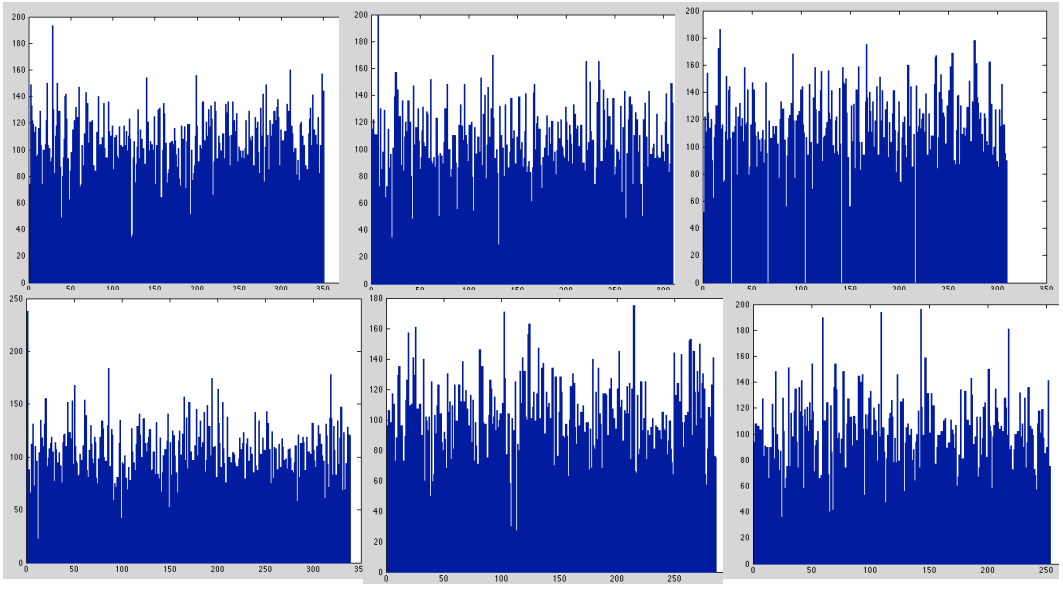

Figure 9: Histograms visualizing the clustering results of each dataset (top left – stage 1, bottom right – stage 6)

Discussion and Conclusion

In the first part of this research, it was found that Canny edge detection is superior to Sobel edge detection for embryo image segmentation applications. Various parameters of Canny edge detection for segmentation in multiple stages of embryo development were tested. Favorable results differed depending on stage of development and desired segmentation quality and traits. The most favorable parameters for one stage were not necessarily advantageous for another developmental stage.

The most consistent results arose from the use of the default parameters. This is a logical result, as the low Gaussian filter sigma values produced by the default parameters results in a greater validity of edges. Default parameter Canny edge detection had the advantage of more accurate segmentation, however it was also more conducive to noise and the joining of multiple embryos in a single segmentation.

The program that determined the tone of an embryo and then applied specific Gaussian filter values was unsuccessful at fulfilling its purpose, especially for the second developmental stage data set. This is likely because the "custom" Gaussian filter values, which were set based on the results of tests of the first developmental stage data set, are not accurate for later sets.

Applying a bounding box based on the results of default Canny edge detection was extremely successful and accurate for the vast majority of images. It appears that many of the problems that plagued segmentation with lower Gaussian filter values, such as rough edges, double segmentations, and noise are not present once a bounding box is applied to only the largest continuous segmented area.

The technique used in the first part of the study for determining the successfulness of each segmentation method is extremely subjective and is affected by both human error and bias.

Though a quantitative mathematical method would be ideal for this task, creating one would be extremely time consuming and would not provide much benefit to the overall goal of automated classification of *D. melanogaster* embryo developmental stages.

In the second part of this research, SIFT features were extracted and clustered in order to identify specific gene expression patterns in *Drosophila melanogaster* embryogenesis images. Although time constraints prevented the bag-of-words step of feature classification from being conducted, preliminary analysis of the clustering results suggest that gene expression pattern identification should be successful.

After gene expression patterns are identified using the bag-of-words method, future research will focus on adding spatial information in order to improve the accuracy of this identification. Another possible target for improving the accuracy of gene expression pattern detection is to add additional classification and feature data to the clustered feature matrix. Finally, accuracy may also be improved by using more effective machine learning/clustering algorithms.

References

1. Vivek C. Abraham, D. Lansing Taylor and Jeffrey R. Haskins, 2004, High Content Screening Applied To Large-Scale Cell Biology, *TRENDS in Biotechnology*, Vol.22 No.1, p. 15-22.

2. Jia-Yu Pan, Andre Guilherme, Ribeiro Balan, Eric P. Xing, Agma Juci, Machado Traina, and Christos Faloutsos, 2006, Automatic Mining of Fruit Fly Embryo Images, *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

3. Sudhir Kumar, Karthik Jayaraman, Sethuraman Panchanathan, Rajalakshmi Gurunathan, Ana Marti-Subirana and Stuart J. Newfeld, 2002, BEST: A Novel Computational Approach for Comparing Gene Expression Patterns From Early Stages of Drosophila
4. melanogaster Development, *Genetics*, Vol. 162(4) p. 2037-47.

5. Andrew A Hill, Peter LaPan, Yizheng Li and Steve Haney, 2007, Impact of image segmentation on high-content screening data quality for SK-BR-3 cells, *BMC Bioinformatics*, Vol. 340(8).

6. Dima AA, Elliott JT, Filliben JJ, Halter M, Peskin A, Bernal J, Kociolek M, Brady MC, Tang HC, and Plant AL, 2011, Comparison of segmentation algorithms for fluorescence microscopy images of cells, *Cytometry A*, Vol. 79(7), p. 545-59.

7. Lu´ıs Pedro Coelho, Aabid Shariff, Robert F. Murphy, 2009, Nuclear segmentation in microscope cell images: A hand-segmented dataset and comparison of algorithms, *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, p. 518-21.

8. Sarkar, S, Sanocki, T, and Bowyer, K, 1996, Comparison of edge detectors: a methodology and initial study, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* p. 143-48.

9. Tomancak P, Beaton A, Weiszmann R, Kwan E, Shu S, Lewis SE, et al. Systematic determination of patterns of gene expression during Drosophila embryogenesis. *Genome Biol*. 2002;3(12)

10. Jia, Xiaoguang, and Mark S. Nixon. Extending the feature vector for automatic face recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 17.12 (1995): 1167-1176.

11. Viikki, Olli, David Bye, and Kari Laurila. A recursive feature vector normalization approach for robust speech recognition in noise. Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on. Vol. 2. IEEE, 1998.

12. Lowe, David G. Object recognition from local scale-invariant features. Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee, 1999.

13. Dalal, Navneet, and Bill Triggs. Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

14. Ojala, Timo, Matti Pietikainen, and David Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on. Vol. 1. IEEE, 1994.

15. Wang, Xiaoyu, Tony X. Han, and Shuicheng Yan. An HOG-LBP human detector with partial occlusion handling. Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009.

16. Lei Yuan, Alexander Woodard, Shuiwang Ji, Yuan Jiang, Zhi-Hua Zhou, Sudhir Kumar and Jieping Ye. Learning Sparse Representations for Fruit-Fly Gene Expression Pattern Image Annotation and Retrieval. *BMC Bioinformatics* 2012, **13**:107 .

17. Shuiwang Ji1, Ying-Xin Li, Zhi-Hua Zhou, Sudhir Kumar and Jieping Ye, 2009, A bag-of-words approach for *Drosophila* gene expression pattern annotation. *BMC Bioinformatics*, **10**:119.

18. Chris Fraley, Adrian E. Raftery, 1998, "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. " *The Computer Journal*, 41:8.

19. “Detecting a Cell Using Image Segmentation.” Internet: http://www.mathworks.com/products/image/examples.html?file=/products/demos/shipping/images/ipexcell.html [October 13, 2012].