

Michael Perry

August 25, 2010

## Analysis of Video Data from Environmental Observing Systems

This project, done in collaboration with the Taiwan Forestry Research Institute, focuses on data analysis tools of bee images, to analyze accumulated data and data that will be acquired in the future. Web cameras, when supported by robust database and visualization systems, provide valuable data for ecological research that go beyond the traditional uses of imagery (Porter et al, 2009). With the usage of fast, sophisticated data acquisition tools, similarly sophisticated image analysis techniques are sought after. A major part of this image analysis, as far as my project is concerned, involves mechanically counting the number of bees in captured images, a task that is challenging but can greatly speed up the process of image analysis. The head of TFRI's bee project, Dr Shen-Shang Lu, would ultimately like to study the social behavior of bees and wasps and also the environmental effects such as weather and temperature on them. From my more specific point of view, my project involved testing and developing the bee counting program, with an emphasis on usability and allowing the program to automatically provide useful statistical data. It should also be noted that this project's purpose is not only to help TFRI with their research, but the ecoinformatics field as a whole by working on a technique that can be utilized by researchers of a wide variety interests.

This project was very different than what had been anticipated in the months leading up to the internship. I had received instruction and training regarding DataTurbine during the quarter before Taiwan, but my project required no work relating to DataTurbine. The first week was spent

trying to find a project relating to DataTurbine that could be done, but it was concluded that TFRI already had a sophisticated infrastructure using RBNB set up for video cameras, and that Dr Sheng-Shan Lu, the head of the bee project, was mainly interested in data analysis. After a week of meetings and brainstorming, I had a clearer goal, which involved development of the bee counting program. What was desired was results and features that could help Dr. Lu with his research, therefore, specifics of the development were according to my discretion.

The greatest component of this project is the bee counting program, which is a Java library developed by Michael Nekrasov and based on a blob-counting public domain library written by Julien Gachadoat. Usage and editing of the source code is permissible under the terms of the GNU Public License. The program comes in two forms: calibrate, an executable that displays a user interface, showing the image with a colored square around the recognized blob, and AutoDetect, another executable that runs on the command line. Both programs can process an image or a directory of images, and take the form of java executables (.jars) and can be run using the command console using command line arguments. I used the Eclipse IDE to manage and edit the source code of the program, which made coding much faster and more convenient.

According to the author of the blob-detection library, the library is aimed at doing computer vision by finding 'blobs' on an image, that is to say areas whose brightness is above or below a particular value (Gachadoat, 2006), and this program is indeed good at doing what it promises. However, at the state I received the program, it could benefit from some extra capabilities and documentation that would greatly help the user fully utilize the program, and provide useful information regarding the process. For example, the program did not provide any statistical information regarding the counting process itself. Keeping in mind the general user base of this

program – researchers who are interested in data and statistics, I included some features that were relatively easy to add but will be very useful: average bee count of a group of images, number of files processed, speed of the process, time it took for the process, and highest and lowest count in a directory and the respective images the counts came from. Additionally, the program lacked a simple, practical set of instructions that explained to the users how to use the program and what features the program has. I wrote this documentation, and passed it on to a TFRI worker, who has translated it into Chinese.

A major feature that I added, that can increase the robustness of the program, was the option add Rules. A Rule specifies the bounds in which the blobs will be recognized: physical bounds as well as bounds of brightness. Potentially, different rules allowed the detection of differently sized specimens. With the addition of this feature, a user may now, using a command line argument, add a new rule, specifying the rule name and description, minimum and maximum width and height, the light sensitivity, and color of the bounding boxes (applicable only for calibrate). The Rule is stored and will automatically be used for all future processes, unless the user deletes the Rule. With this feature, the program may be passed to other researchers who are interested in image detection but want to detect things of other sizes, or wish to detect two or three different specimens at the same time.

The preliminary stages of the project involved, other than learning how to use the program, testing it. I was provided with a large body of images of a bee's nest, acquired all the way back from 2006. An image was taken every minute, during the time of the day when it was light enough for bees to be visible. Accuracy tests were done, and figures were compared to hand-counted figures, with the assumption that hand-counted figures are accurate. Furthermore, frequency tests

were done. Dr. Lu was interested in discovering which snapshot frequency is permissible, without compromising accuracy. One, five, and ten-minute snapshots were chosen. The test results are outlined in the following table.

	<b>10 minutes</b>	<b>5 minute</b>	<b>1 minute</b>	<b>Hand Count</b>	<b>Time (program)</b>	<b>Time (by hand)</b>
<b>15-Jun-10</b>	33.22	33.06	33.7	31.06	23 m	6 h 49 s
<b>12-Mar-09</b>	21.7	21.71	21.28	9.92	26 m	5 h 15 m

Ideally, more samples would be needed for the testing to be statistically significant. However, this was not possible due to the time it took to hand-count the images. What could be concluded by not only this table but also common sense is that hand-counting takes significantly more time than machine counting. It is estimated that it would take almost a year of time to hand-count a year's worth of images. Clearly a faster solution is ideal. Also, it seems from the results that a lower frequency of snapshots could be taken, without compromising accuracy much. The most noteworthy observation is the loss of accuracy, especially for the March sample day. From further observations, the program often overcounts. This is due to a lack of contrast between the bees and the background, which is a result of either dust or poor lighting, two things that are difficult to control. Undercounting also occurs when bees cluster too closely together; this is something that is impossible to control.

There does exist a very promising solution to the program's accuracy problems. Images produced from infrared cameras, edited using image editing software to emphasize red, then converted to black and white, created perfectly contrasted images, eliminating contrast problems due to dust and poor lighting (but not eliminating bee clustering problems). However, usage of infrared cameras in the field is not possible right now due to their cost. In the future, if infrared cameras become more affordable, with the current algorithm, this would be the best solution to the

bee counting program. Using advanced Java image-editing libraries, it should not be much of a challenge to automatically edit hundreds of infrared photos to create the perfectly contrasted images.

It has been a great opportunity to be able to work with such a sophisticated program, and especially with such professional and wonderful co-workers. I have learned a great deal about Java, ecoinformatics, and statistics. By testing the program, I have learned of its capabilities and limitations. By adding features, I have learned how the exactly the program works, and have helped TFRI by filling their specific needs. Currently, there are still many problems with the program that I wish I could fix, more features that I wanted to add, and more image samples I would like to have tested, but all in all, I am pleased with the progress I've made. I also know enough about the program and know enough about what the program needs to improve, that I would like the opportunity to work with it some more. It has been a pleasure playing a small part in furthering the field of ecological informatics.

## References

Porter, J., Lin, C.C., Smith, D.E., Lu, S.S. 2009. Page 57. Ecological image databases: From the webcam to the researcher. Ecological Informatics. [www.elsevier.com/locate/ecolinf](http://www.elsevier.com/locate/ecolinf)

Gachadoat, J. 2009. Blob detection library. <http://v3ga.net/processing/BlobDetection/>