Research Note

Automatic analysis of Camera Image Data: an Example of Honey Bee (*Apis cerana*) Images from Shanping Wireless Sensor Network

Sheng-Shan Lu,[1)] Michael Perry,[2)] Michael Nekrasov,[2)] Tony Fountain, [2)] Peter Arzberger,[2)] Yu-Huang Wang,[1)] Chau-Chin Lin[1,3)]

【**Summary**】

Under the international collaborative program between TFRI and PRIME of San Diego University USA in 2010, we extended an image analysis package and applied it to honey bee observations. In this article we describe the result of this collaboration. A tool suitable for routine measurements and counting tasks was developed to perform automatic process. We applied blob-detecting of computer vision technique to develop this package. We then tested the tool by using different bee presence number images collected from Shanping wireless sensor network of TFRI. We compared the time consumed between the automatic process and manual process. Result shows that the low number of bee presence image (average bee number <30 individuals per image) analysis between automatic process and manual process are 9 and 315 mins. A similar result shows that in the high number of bee presence image (average bee number >30 individuals per image) analysis between automatic process and manual process are 23 and 409 mins. Although the automatic process overestimates 2%-21% of bee counting, the tool shows significantly reducing the process time. We conclude that the program provides a convenient process to determine the target and thus facilitate the examination of large volume honey bee images from wireless sensor network of the field.

Keywords: bee image, image analysis, automated identification, blob detection.

Lu SS, Wang YH, Perry M, Nekrasov M, Funtain T, Arzberger P, Lin CC. 2011.

Automatic analysis of camera image data: an example of honey bee (*Apis* *cerana*) images from Shanping wireless sensor network. Taiwan J For Sci 26(x):xxx-xxx.

Short running title: Automatic analysis of camera image

---

[1] Forest Protection Division, Taiwan Forestry Research Institute, 53 Nanhai Rd., Taipei 10066, Taiwan. 林業試驗所森林保護組，10066台北市南海路53號。

[2] California Institute for Telecommunications and Information Technology, UCSD, CA 92093, USA. 聖地牙哥加州大學電信與資訊技術學院。

[3] Corresponding author, email: chin@tfri.gov.tw 通訊作者。

Received January 2011, Accepted xxxx 2011. 2011年1月送審 2011年xx月通過。

研究簡報

自動化影像分析：以扇平無線感測網之東方蜜蜂(*Apis cerana*)影像爲例

陸聲山[1)] Michael Perry[2)] Michael Nekrasov[2)] Tony Fountain[2)] Peter Arzberger[2)]

王豫煌[1)]　林朝欽[1, 3)]

摘　要

林業試驗所於2010年，透過人才交流計畫(PRIME)的國際合作，與美國聖地牙哥大學合作，針對無線感測器網所獲得的生態研究影像，開發自動電腦分析工具，並將其應用於東方蜜蜂行爲觀測之研究。此工具利用無線感測器網的攝影機所獲得的影像，以影像斑點偵測技術，執行自動化辨識並計算東方蜜蜂出現的數量。本研究以670張東方蜜蜂出現低隻數(平均<30隻)及與800張東方蜜蜂出現高隻數(平均>30隻)之影像，分別以人工處理與電腦自動處理分析測試此工具之可用性。結果顯示：人工處理分析低隻數蜜蜂影像耗時315分；電腦自動處理分析則僅需9分。高隻數蜜蜂影像之人工處理分析需409分；電腦自動處理分析則僅花費23分。雖電腦自動處理分析高估蜜蜂數2%-21%，但顯著降低處理時間。本研究獲得初步結論爲：經由無線感測器網獲取的大量影像數據，可透過電腦自動處理分析獲得快速與正確的辨識結果。

關鍵詞：蜜蜂影像、影像分析、自動化辨識、斑點偵測。

陸聲山、Perry M、Nekrasov M、Fountain T、Arzberger P、王豫煌、林朝欽。2011。自動化影像分析：以扇平無線感測網之東方蜜蜂(*Apis cerana*)影像爲例。台灣林業科學26(x):xxx-xxx。

A beehive is an enclosed structure that some honey bee species live and raise their young. Beehives (simply as nests) are occupied by honey bee colonies. The beehive's internal structure is a densely packed matrix of hexagonal cells. The bees use the cells to store food (honey and pollen), and to house the eggs, larvae, and pupae. It turns that the beehive is a food source of other organisms that are the predators of honey bees. Many wasps are predatory, using other bees as food for their larvae (Abrol 1994). For instance, the Asian giant hornet (*Vespa mandarinia*) is a relentless hunter that preys on other insects such as honey bees (Lu *et al*. 2009). The hornets often attack honey bee hives with the goal of obtaining the honey bee larvae. A single Asian giant hornet can kill as many as 40 honey bees per minute using their large mandibles which can quickly strike and decapitate a bee. Although the hornets can easily defeat the defenses of many individual honey bees, the honey bees also possess a collective defense against them (Abrol 2006). When a hornet scout locates and approaches to honey bee hive it will emit specific pheromonal hunting signals. When the honey bees detect these pheromones, a hundred or so will gather near the entrance of the nest and set up a trap, keeping it open apparently to draw the hornet further into the hive or allow it to enter on its own (Abrol 1994, 2006). Entomologists in Taiwan Forestry Research Institute (TFRI) are interested in understanding the interactions of predatory wasps and honey bees.

Thus, a wireless sensor network that includes a web camera has been deployed in the field since 2007 to monitor the honey bee defensive behavior. The Bee Camera allowed hornet attacks on a bees' nest to be monitored every minute over the course of an entire year, something that would have been impossible by a human observer (Porter *et al.* 2010). Taking advantage of the camera's ability, images representing unobtrusive observations of a honey bee nest have been captured over long time periods in the southern forest of Taiwan for 3 years. However, image data each obtained by 1 min frequency requires high time consuming manual processing and analysis of images in the laboratory. Efficient analysis of image data has long been a dream of many biologists, especially for taxonomists working on automatic species identification (Larios et al. 2007, MacLeod 2007, Mayo and Watson 2007, Francoy et al. 2008, Salle 2009). But automated image recognition and analysis is a vast and complex project which requires a lot of technical background. Fortunately, technological advances in cyberinfrastructure and computing ecology such as image analysis programs have provided the opportunity to make the automatic analysis possible. 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. 2010). With the use of fast, sophisticated data acquisition tools, similarly sophisticated image analysis techniques are sought after. Under the collaborative program between TFRI and PRIME (Pacific RIM undergraduate Experiences, Arzberger et al. 2010) in 2010, we addressed the extension of observing system on honey bee image identification and analysis.

The purpose of the collaboration is to develop a tool which is suitable for routine measurements and the counting task. The honey bee image source is from the wireless sensor network in Shanping, Taiwan which has been set up since 2007 (Lu et al. 2009, Porter et al. 2010). The tool is expected to perform automated calculation by using image acquisition and image analysis techniques. In addition, the tool is also to be accurate in identifying the target (here is the example of bees) and thus to facilitate the examination of large volume of image data. In this article we describe the result of our collaboration by presenting the development of an efficient and accurate computer program for the routine analysis of bee images.

The bee counting tool is a Java library developed and based on a blob-detecting public domain library. In the area of computer vision, blob detecting refers to visual modules that are aimed at detecting points and/or regions in the image that are either brighter or darker than the surrounding. According to this rule, 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 2009), and this technique is indeed good at doing what it promises. By searching the target area where the bees are easily identified, we see that the image of the bee is different from the background (lighter or darker). Using this difference we can automatically calculate the number of bees. Usage and editing of the source code of this collaboration is permissible under the terms of the GNU Public License. The result welcomes ecologists who are interested in the similar kinds of analysis to use and modify for their specific purposes.

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 interfaces can process an image or a directory of images, and take the form of java executables (.jars) and can be run on the command console using command line arguments.

Users can perform bee counting by either Calibrate or AutoDetect. Calibrate is a GUI (Graphical User Interface) containing controls and the processed picture, with the boxed bees used to run the job of counting (Fig. 1). Since the program calibrate is relatively slow, it is not recommended for processing many pictures at the same time. The interface provides users the ability to adjust the box sizes of bees we want to count, and to control the brightness for recognition of the bees. Alternatively, users can choose another program called AutoDetect which is a non-GUI program. This program is very fast and outputs results on the command line. AutoDetect can also compute the average count, time elapsed, and the speed of the process when processing many pictures at the same time (Fig. 2). When processing many pictures at the same time, it is better to use the AutoDetect program.

In order to increase the robustness of the program, a rule option is added to specify the bounds in which the blobs will be recognized: physical as well as brightness bounds. Potentially, different rule allows for the detection of different size of 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 of box, brightness sensitivity, and color of the bounding boxes. The rule is stored and will automatically be used for all future processing, unless the user deletes or modifies the rule. If there is little difference between sequential images, the detection rule will naturally shorten the processing time. 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 sizes of specimens at the same time.

We tested the tool developed by comparing the automatic and manual processes using the initial 670 images with low number of bee presence (average bee number <30 individuals per image) and 800 images with high number of bee presence (average bee number >30 individuals per image). Results show that the time consumed by manual processes took about 315 and 409 mins. But, the time was reduced to 9 and 23 mins, respectively, by our two versions (one without rule and the other to add rules) of the program (Table 1). Both low number and high number of bee presence counting show that automatic analysis significantly reduces the processing time for large number of images.

With the assumption that manual processes are accurate, we tested the accuracy of automatic counting result. The test results show that the of the automatic process was overestimate 21% in the low number of bee presence images, but only overestimated 2% in the higher number bee presence images. Reasons of the high overestimation in the low number bee presence images might be the dusts on the background of the images. The dusts in too low number of bee presence cause the high misinterpretation by machine.

Furthermore, we tested the frequency of image sampling on bee number counted by choosing 1, 5 and 10 min intervals of captured images. The test shows that numbers of bee counted averagely are 33.7, 33.06 and 33.22 (Table 2). The results indicate the overestimations are not higher than the highest frequency sampling. Therefore scientists can adjust their sampling frequency without worrying about overestimation. Obviously, researchers could adjust their sampling frequency to create a more efficient counting program to facility their research.

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 manual process huge number of images. What could be concluded, not only from the results in Table 1, but also from common sense is that manual processing takes significantly more time than machine counting. It is estimated that it would almost take one year to manual process whole year’s sensor images. Clearly, a faster solution is needed. From the results it seems that a lower frequency of snapshots could be taken, without compromising accuracy much.

In terms of accuracy of our program, the most noteworthy observation is the loss of accuracy, especially for the rainy sample days. From further observations, the program often over-counts. 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 conditions that are difficult to compensate after the photos being taken. It might be possible to improve the field condition before taking photos, such as re-painting the background or adjusting the camera position. Furthermore, under-counting also occurs when bees cluster too closely together, this is something that is more difficult to control.

There is a very promising solution to improve the program’s accuracy. Images from thermal infrared cameras processed by image editing software to enhancing higher thermal region, and then transformed into black and white perfectly contrasted images (Fig. 3), eliminate 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.

A major part of this image analysis involves computerized counting the number of bees in the captured images, a task that is challenging but can greatly speed up the process of image analysis. This 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 also should be noted that this project’s purpose is to help not only scientist with their research, but the ecoinformatics field as a whole by working on a technique that can be used by researchers with various interests.

This tool is designed in the spirit of open source software, to provide researchers easy to use software and an interface to make the program more flexible to use and adapt, thus providing the opportunity to adjust operations to other species with similar characteristics. Image processing is the most difficult part of the feature extraction; by pre-treating images to get better image quality, one can also greatly improve the performance and accuracy of computing, making post-process of a large number of images an easy operation.

ACKNOWLEDGEMENTS

This material is based upon the work supported by the Taiwan Forestry Research Institute under Grant 2010 Ecoinformatics study and PRIME 2010 and associated Pacific Rim Applications and Grid Middleware Assembly, both funded by the US National Science Foundation and supported by the California Institute of Telecommunications and Information Technology.

LITERATURE CITED

Abrol DP. 1994. Ecology, behaviour and management of social wasp, *Vespa velutina* Smith (Hymenoptera: Vespidae), attacking honeybee colonies**.** Korean J Apiculture 9:5-10.

Abrol DP. 2006. Defensive behaviour of *Apis cerana* F. against predatory wasps. J Apicultural Sci 50:39-46.

Arzberger P, Wienhausen G, Abramson D, Galvin J, Date S, Lin FP, Nan K, Shimojo S. 2010. PRIME: an integrated and sustainable undergraduate international research program. Advances in Engineering Education. Vol 2, No 2 Available at http://advances.asee.org/vol02/issue02/05.cfm. 34p.

Francoy TM, Wittmann D, Drauschke M, Muller S, Steinhage V, Bezerra-Laure MAF, De Jong D, Goncalves LS. 2008. Identification of Africanized honey bees through wing morphometrics: two fast and efficient procedures. Apidologie 39:488-94.

Gachadoat J. 2009. Blob detection library. Available at:

http://v3ga.net/processing/BlobDetection/.

Larios N, Deng H, Zhang W, Sarpola M, Yuen J, Paasch R, Moldenke A, Lytle DA, Ruiz-Correa S, Mortensen E, Shapiro LG, Dietterich TG. 2007. Automated insect identification through concatenated histograms of local appearance features. Mach Vision Appl 19:105-23.

Lu SS, Lin WC, Chen YH, Lin CC. 2009. Application of wireless sensor network to study the defensive behavior of *Apis cerana* (Hymenoptera: Apidae). J of Nat Park 19:1-8. [Chinese with English abstract].

MacLeod N. 2007. Automated taxon identification in systematics: theory, approaches and applications. New York: CRC press. 339 p.

Mayo M, Watson AT. 2007. Automatic species identification of live moths. Knowl-Based Syst 20:195-202.

Porter J, Lin CC, Smith DE, Lu SS. 2010. Ecological image databases: from the webcam to the researcher. Ecol Inform: 5:51-8.

Salle JL, Wheeler Q, Jackway P, Winterton S, Hobern D, Lovell D. 2009. Accelerating taxonomic discovery through automated character extraction. Magnolia Press Zootaxa 2217:43-55.

Table 1. Comparison of the number of bee count between manual process and automatic process at two differeent bee density

Bee counting method	Manual process	Automatic process (ver. 1)	Automatic process (ver. 2 rule added)
Low number of bee presence (Mar 12 2009)			
Photo processing number	670	670	670
Average number of bees counted per image	9.92(A)(B)(C)[a]	21.28(A)(B)(C)	12.03(A)(B)(C)
Processing time (mins)	315	26	9
	F=235.28, df=2, p=0.0002		
High number of bee presence (Jun 15 2010)			
Photo processing number	800	800	800
Average number of bees counted per image	31.06(A)[a]	33.14(A)	33.70(A)
Processing time (mins)	409	31	23
	F=1.128, df=2, p=0.3237		

[a] Means with the same letter within a row are not significantly different at the 0.01 level, using Tukey's mean separation test.

Table 2. The number of bee count through automatic process under three differeent smapling frequency

Sampling frequency	1min	5 mins	10 mins
Photo processing number	800	160	80
Average number of bees counted per image	33.70	33.06	33.22
Processing time (mins)	23	6.6	2.9

Fig. 1. A graphical user interface showing the image with a colored square around the recognized blob and the bee number of automatic process.

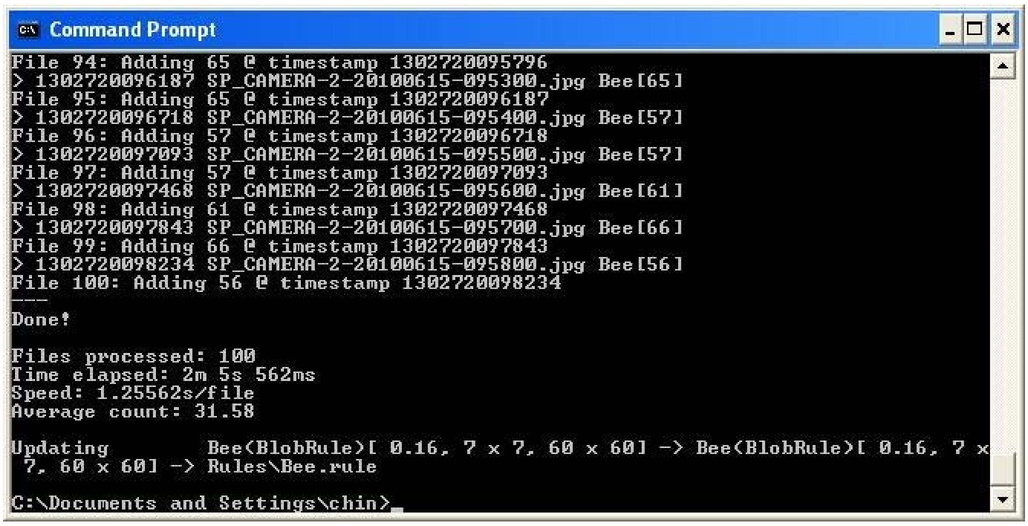

Fig. 2. AutoDetect with a custom rule, output the average count number on the command line.

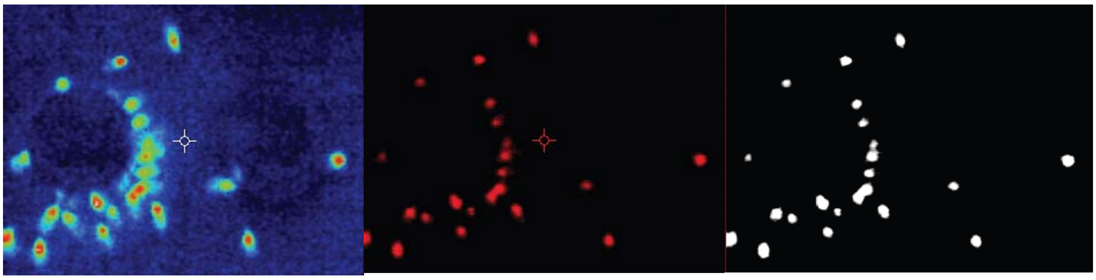

Fig. 3. Images produced from infrared cameras (left), edited using image editing software to emphasize red (middle), and then converted to black and white (right), this approach is very promising solution to the program's accuracy problems.