Research note

Automatic Analysis of Camera Image Data: An Example of Honey Bee (*Apis cerana*) Images from the 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 an international collaborative program between the Taiwan Forestry Research Institute (TFRI) and Pacific RIM undergraduate experience (PRIME) of San Diego University, San Diego, CA, USA in 2010, we extended an image analysis package and applied it to honey bee observations. In this article, we describe the results of this collaboration. A tool suitable for routine measurements and counting tasks was developed to perform an automatic process. We applied blob-detecting of a computer vision technique to develop this package. We then tested the tool using images with different numbers of bees present collected from the Shanping wireless sensor network of TFRI. We compared the times consumed between the automatic and manual processes. Results showed that analysis of images with a low number of bees present (with an average bee number of < 30 individuals per image) between the automatic process and manual process respectively required 9 and 315 min. A similar results showed that analysis of images with a high number of bees present (with an average bee number of > 30 individuals per image) between the automatic process and manual process respectively require 23 and 409 min. Although the automatic process overestimated bee counts by 2~21%, the tool shows significant reductions in processing times. We concluded that the program provides a convenient way to determine the target and thus facilitate the examination of a large volume of honey bee images from a wireless sensor network in the field.

Key words: bee image, image analysis, automated identification, blob detection.

Lu SS, Perry M, Nekrasov M, Fountain T, Arzberger P, Wang YH, Lin CC. 2011. Automatic analysis of camera image data: an example of honey bee (*Apis cerana*) images from the shanping wireless sensor network. Taiwan J For Sci 26(3):305-11.

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, Univ. of California at San Diego, San Diego, CA 92093, USA. 聖地牙哥加州大學電信與資訊技術學院。

3) Corresponding author, e-mail:chin@tfri.gov.tw 通訊作者。

Received January 2011, Accepted May 2011. 2011年1月送審 2011年5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*)影像為例

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為觀測之研究。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為：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*)影像為例。台灣林業科學26(3):305-11。

A beehive is an enclosed structure in which 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 house eggs, larvae, and pupae. It turns that the beehive is a food source of other organisms that are 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 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 a 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 apparently open to draw the hornet further into the hive or allow it to enter on its own (Abrol 1994, 2006). Entomologists of the Taiwan Forestry Research Institute (TFRI) are interested in understanding interactions of predatory wasps and honey bees.

Thus, a wireless sensor network that includes a web camera was deployed in the field since 2007 to monitor the honey bee defensive behavior. The Bee Camera allows hornet attacks on a bees' nest to be monitored every minute over the course of an entire year, something that would be impossible with a human observer (Porter et al. 2010). Taking advantage of the camera's ability, images representing unobtrusive observations of a honey bee nest were captured over long time periods in a forest in southern Taiwan for 3 yr. However, image data obtained with a 1-min frequency is highly time consuming to manually process and analyze 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 are vast and complex project which requires broad technical background. Fortunately, technological advances in cyberinfrastructure and computing ecology such as image analysis programs have provided the opportunity to make automatic analyses possible. Web cameras, when supported by robust database and visualization systems, can 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 being sought. Under a collaborative program between TFRI and PRIME (Pacific RIM undergraduate Experiences, Arzberger et al. 2010) in 2010, we created an extension of an observation system for honey bee image identification and analysis.

The purpose of the collaboration was to develop a tool which is suitable for routine measurements and counting tasks. The honey bee image source is from the wireless sensor network at Shanping, Taiwan which was set up in 2007 (Lu et al. 2009, Porter et al. 2010). The tool is expected to perform automated calculations using image acquisition and image analysis techniques. In addition, the tool can also be used to accurately identify the target (here in the example is bees) and thus facilitate the examination of large volumes of image data. In this article, we describe the results 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 detection refers to visual modules that are aimed at detecting points and/or regions on an image that are either brighter or darker than the surroundings. According to this rule, the library is aimed at doing computer vision by finding "blobs" on an image, that is to say areas the brightness of which 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, one can see that the image of the bee differs from the background (lighter or darker). Using this difference, we can automatically calculate the number of bees. Using and editing the source code of this collaboration are permissible under the terms of the GNU Public License. We welcome ecologists who are interested in similar kinds of analysis to use and modify this tool for their own specific purposes.

The program comes in 2 forms: Calibrate, an executable program that has a user interface, showing the image with a colored square around the recognized blob, and Auto-

Detect, another executable program 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 using 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 Calibrate program is relatively slow, it is not recommended for processing many pictures at the same time. The interface provides users with the ability to adjust the box sizes of bees they want to count, and control the brightness for recognizing the bees. Alternatively, users can choose another program called AutoDetect which is a non-GUI program. This program is very fast and displays results on the command line. AutoDetect can also compute the average count, time elapsed, and 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 was added to specify the bounds in which the blobs will be recognized: there are physical and brightness bounds. Potentially, a different rule allows for the detection of different-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 of the box, brightness sensitivity, and color of the bounding boxes. The rule is stored and is automatically used for all future processing, unless the user deletes or modifies

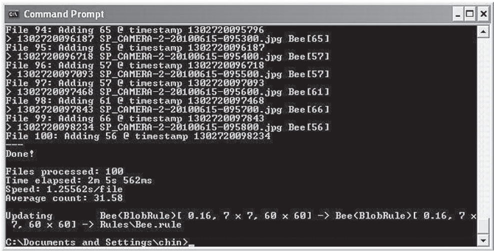

Fig. 2. AutoDetect with a custom rule. The average count number is displayed on the command line.

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

it. 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 2 or 3 different-sized specimens at the same time.

We tested the tool developed by comparing the automatic and manual processes using an initial 670 images with a low number of bees present (with an average bee number of < 30 individuals per image) and 800 images with a high number of bees present (with an average bee number of > 30 individuals per image). Results showed that the times consumed by the manual processes took about 315 and 409 min, respectively. But the times were reduced to 9 and 23 min, respectively, by our 2 versions (one without rules and the other to which rules can be added) of the program (Table 1). Counting both low and high numbers of bees present showed that the automatic analysis significantly reduced the processing time for a large number of images.

With the assumption that the manual processes were accurate, we tested the accuracy of automatic counting results. The test results showed that the automatic process overestimated by 21% the number of bees on images with a low number of bees present but only overestimated by 2% that on images with a higher number bees present. Reasons for the high overestimation on images with a low number of bees present might have been the dust on the background of the images. The dust on images with a very low number of bees present may have caused high misinterpretation by the machine.

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

Table 1. Comparison of the number of bee counts between the manual process and automatic processes at 2 differeent bee densities

Bee-counting method	Manual process	Automatic process (ver. 1)	Automatic process (ver. 2 rule added)
Low number of bees present (March 12, 2009)			
No. of photos processed	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 (min)	315	26	9
	$F = 235.28$, df = 2, $p = 0.0002$		
High number of bees present (June 15, 2010)			
No. of photos processed	800	800	800
Average number of bees counted per image	31.06(A)[a)]	33.14(A)	33.70(A)
Processing time (min)	409	31	23
	$F = 1.128$, df = 2, $p = 0.3237$		

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

Table 2. Number of bees counted by the automatic process under 3 differeent sampling frequencies

Sampling frequency (min)	1	5	10
No. of photos processed	800	160	80
Average number of bees counted per image	33.70	33.06	33.22
Processing time (min)	23	6.6	2.9

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 manually process a 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 take almost 1 yr to manually process an entire 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 greatly compromising the accuracy.

In terms of the accuracy of our program, the most noteworthy observation was the loss of accuracy, especially on rainy sampling days. From further observations, the program often over-counts. This was due to a lack of contrast between the bees and background, which was a result of either dust or poor lighting—two conditions that are difficult to compensate for after the photos are taken. It might be possible to improve the field conditions 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, 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 enhance higher thermal regions, and then transformed into black and white perfectly contrasting images (Fig. 3) eliminate contrast problems due to dust and poor lighting (but do not eliminate bee-clustering problems). However, the use of infrared cameras in the field is not possible at this time 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 perfectly contrasting images.

A major part of this image analysis involves computerized counting of 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

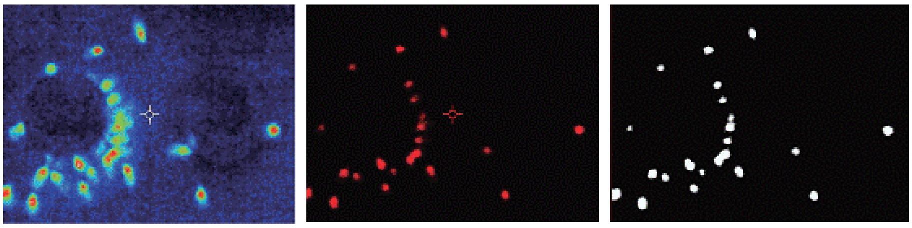

Fig. 3. Images produced from an infrared camera (left), edited using image-editing software to emphasize red (middle), and then converted to black and white (right). This approach is a very promising solution to the program's accuracy problems.

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 was to help scientists with their research, and the ecoinformatics field as a whole by working on a technique that can be used by researchers with various interests.

This tool was designed in the spirit of open source software, to provide researchers with 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 feature extraction; by pretreating images to obtain better image quality, one can also greatly improve the performance and accuracy of computing, making the post-processing 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 the 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. Kor J Apicult 9:5-10.

Abrol DP. 2006. Defensive behaviour of *Apis cerana* F. against predatory wasps. J Apicult 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. 34 p.

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 Natl Park 19:1-8. [in 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.