An Online Computer Vision Prototype for the Early Detection of Infestation in Greenhouses

in the framework of Integrated Pest Management Methods

Ikhlef Bechar
Ikhlef.bechar@sophia.inria.fr
INRIA Sophia Antipolis Méditerranée, PULSAR TEAM

in collaboration with
INRA Avignon, Unité de Pathologie Végétale
CREAT, Chambre d'Agriculture des Alpes Maritimes

ISSTB SEMINAR 26/03/2010
Outline

• Context and related work
• BioSerre Project : Goals and Methods
• Proposed Computer Vision Approach
  – Extraction of potential insect locations in static frames
    ➢ Transformation of a video from RGB space into Gray space
    ➢ Some related work on Extraction of low contrasted spots in unspecific backgrounds
    ➢ Joint space-intensity model of an insect of interest
    ➢ Derivation of the object recognition criterion
    ➢ Some results
  – On-line Video Processing Solution
    ➢ Background Subtraction approach
    ➢ Object presence detection approach
    ➢ Frame-2-Frame Object tracking
• Conclusion
Context (1/2)

- **Intensive** greenhouse cultures (roses, tomato, strawberry, pepper, etc.)
- A greenhouse is an input/output system with constant evolution
- Temperature and hygrometric conditions inside a greenhouse favor:
  - frequent and rapid attacks of bio-aggressors (insects, acarids, mites, fungi, etc.)
  - Drastic proliferation of bio-aggressors
- Difficult to guess exactly starting **time and focus regions** of the attacks
- Such attacks cause severe agricultural/economical damages
Context (2/2)

- **Solution**: Integrated Pest Management Methods for (greenhouse) cultures which:
  - Consist in integrating various natural means like:
    - Biological (Prophylactic) fighting methods (e.g. auxiliaries)
    - Acting on climate factors inside a greenhouse
    - Agricultural solutions
    - Parsimonious use of chemicals
  - Guarantee:
    - A sustainable agriculture
    - Good quality for the consumer and profitability for the cultivator
    - Less harmful solutions to the agents
    - Less polluting of the environment through reduction of pesticide use (i.e. ecological)
Some related work on plant/insect monitoring

• Approaches for plant disease monitoring
  – Hudelot; C., 2003: Automatic plant image interpretation to early pathology detection
  – Moya; E.A. et al., 2005: Assessment of disease severity caused by powdery mildew
  – Granitto; P.M. et al., 2005: Identification of weed seeds by machine vision & classification (Naive Bayes classifier versus neural networks)
  – Skaloudova; B. et al., 2006: Estimation of leaf damages caused by spider mites

→ Use static images, not in situ generally
  - Giacomelli; G.A. et al., 1996: Real time plant monitoring for stress detection in a plant growth chamber (not a greenhouse context)

• Measure of whitefly density by image analysis
  - Bauch, C., et al., 2005: Mechanical/Vision system (aspiration mechanism, filter, image analysis, etc.) : invasive

• Video monitoring of honey bee colonies
  - Campbell; C. et al., 2008
BioSerre Project : Towards an e-Greenhouse?

**Goal** : Design of a Non Invasive Automatic Vision System called “DIViNe” (Detection of Insects by a Video camera Network) to:

- *In situ* autonomous monitoring of greenhouse cultures
- Early detection of infestation
- Reduction of pesticide use and human intervention
- Produce interesting data and vision algorithms for insects behavior mining

**Methods** :

- **Resources** : video sensing network in a greenhouse + PCs
- **Processing tools** :
  - Video processing
  - Machine learning
BioSerre Steps

- Video Acquisition
  - Image sequences, spatiotemporal sampling

- Video Processing Chain
  - Insects identification
  - Insects trajectories

- Systems Validation
  - Pest counting results

- Behavior Analysis and Recognition
  - Scenario Recognition (laying, predation...)

Done               Ongoing work               Future work
Video Acquisition system (1/2): Prototype

- Network of 5 wireless video cameras (protected against sprinkle of water and sunshine)
- In a 130 m² greenhouse at CREAT planted with 3 varieties of roses
- Principle: Continuously observing sticky traps during daylight (method already used but manually...) + Statistical analysis of results
Video Acquisition system (2/2): GUI

- Axis Camera technologies
- Video acquisition platform (Keeneo)
- Socket programming

- End-users can:
  - Acquire Images from remote cameras:
    - Tune camera parameters
    - Schedule acquisition time
  - Configure / launch video processing:
    - Tune processing parameters
    - Enable actions (e.g. alarms)
Video processing challenges

- Quasi real-time video processing
  - necessary both from a practical and processing viewpoints
- Big image size (~1,300,000 px.)
- Low resolution of objects (30 ~ 100 px.)
- Low color contrast (PSNR)
- Unclearly defined object borders
- Absence of shape information
- In-situ constraints:
  - Non-stationary background (light changes, shadows, etc.)
  - Reflectances, intensity saturation
  - Presence of outliers (other insects, leaves, dust, humans, etc.)
Automatic extraction of the trap

• Principle: Extract the sticky trap once and for all from the first frame of a camera movie

• Hypotheses: Color constancy + Compactness

• Algorithm:
  - Sample the background intensities around the image center
  - Estimate 1\textsuperscript{st} and 2\textsuperscript{nd} order color statistics
  - Predict membership of other image pixels by means of a Chi2 statistical test
  - Post-process:
    - Extract them by using the Connected Components Algorithm
    - Fill in Holes
  - Finally, encode the trap in a sparse data structure to speed up subsequent video-processing
RGB-into-Gray Video Transformation

• Linear transformation from RGB space into gray-scale space that best separates insects from the trap's background => Maximize SN ratio between a sample of insect RGB intensities \( S_I = \{ (R_i, G_i, B_i); i = 1, \ldots, N_I \} \) and a sample of background RGB intensities \( S_B = \{ (r_j, g_j, b_j); j = 1, \ldots, N_B \} \) (done off-line)

\[
\frac{\sum_{i=1}^{N_I} (t_r R_i + t_g G_i + t_b B_i)^2}{\sum_{j=1}^{N_B} (t_r r_j + t_g g_j + t_b b_j)^2} \rightarrow \max_{(t_r, t_g, t_b) \in \mathbb{R}^3}
\]

• Amounts to solving a generalized eigen value problem:

• Linear transformation coefficient

vector “t” is found as the generalized eigen vector corresponding to the greatest generalized eigen value

• Cheap from a computational point of view

Show movie

Original RGB frame  
Transformed gray frame
Approaches for Automatic Extraction of Contrasted Spots from Unspecific Backgrounds in images and videos

- A problem very well studied in many application contexts:
  - **Direct approaches**: fluorescence videomicroscopy, microarray imaging, astronomic images
  - **Undirect approaches**: Feature points extraction in active vision, Hough transform, SIFT, SURF, etc.

- Commonly used spot extraction methods:
  - Global thresholding (naive)
  - Gaussian Fitting (naive)
  - Adaptive Filtering
  - Wavelets
  - Local maxima extraction by using the Laplacian

- We developed a **new** simple, fast and more suitable technique for on-line video-processing
Spot detection by pattern recognition (1/4)

- Joint space-intensity model of a spot: Contrasted Rectangular Pattern

  with half-width “w”, half-length “r”, tilt angle “theta” and

  height (intensity) “h”:

  \[ f(x, y) = \begin{cases} 
  h, & \text{if } |x \cos(\theta) + y \sin(\theta)| \leq w \text{ and } |-x \sin(\theta) + y \cos(\theta)| \leq r \\
  0, & \text{otherwise} 
\end{cases} \]

- Obtain a continuous model: Convolve with a Gaussian Kernel:

  \[ K_\sigma(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x^2 + y^2)}{2\sigma^2}} \]

  to find:

  \[ f_\sigma(x, y) = h \left( \Phi_\sigma(x \cos(\theta) + y \sin(\theta) + w) - \Phi_\sigma(x \cos(\theta) + y \sin(\theta) - w) \right) 

  \left( \Phi_\sigma(-x \sin(\theta) + y \cos(\theta) + r) - \Phi_\sigma(-x \sin(\theta) + y \cos(\theta) - r) \right) \]

- Change of variable

  \[ g_\sigma(u, v) = h g_{\sigma, w}(u) g_{\sigma, r}(v) \]

  Where:

  \[ g_{\sigma, w}(u) = \left( \Phi_\sigma(u + w) - \Phi_\sigma(u - w) \right) \]

  \[ g_{\sigma, r}(v) = \left( \Phi_\sigma(v + r) - \Phi_\sigma(v - r) \right) \]
Spot detection by pattern recognition (2/4)

• **Choice of sigma**: In such a way that $f_\sigma(x, y)$ shows a clearly-defined maximum at the centroid of any rectangular object of interest.

• Amounts to maximizing with respect to $\sigma$ the filter response (to be addressed later).

• One then derives the following conditions must hold true simultaneously

\[
\left( \frac{w^2}{\sigma^2} - 3 \right) (2\Phi_\sigma(r) - 1) - \frac{2r}{\sqrt{2\pi}\sigma} e^{-\frac{r^2}{2\sigma^2}} \leq 0
\]
\[
\left( \frac{r^2}{\sigma^2} - 3 \right) (2\Phi_\sigma(w) - 1) - \frac{2w}{\sqrt{2\pi}\sigma} e^{-\frac{w^2}{2\sigma^2}} \leq 0
\]

• **Sufficient condition**:

\[
\sigma \geq \frac{\max(w, r)}{\sqrt{3}}
\]

• $\sigma$ is learned from a dataset of 500 sampled objects off-line.
Spot detection by pattern recognition (3/4)

• Rectangle centroid extraction as local maxima extraction

• We use Karush-Kuhn-Tucker local maxima extraction criteria:

\[ \nabla g_\sigma(0, 0) = 0 \]
\[ \nabla^2 g_\sigma(0, 0) < 0 \]

• Some results:
  • Video 1: Background with slight light reflectance (Recorded at evening time)

  • Video 2: Background with heavy light reflectance (Recorded at noon time)

• Such a criterion is useless mainly due to noise and light reflectance and one needs unavoidably a more robust one
Spot detection by pattern recognition (4/4)

- Saliency of a rectangular pattern is given by the absolute value of the least negative eigen value of the Hessian matrix $s^*$

- Robust criterion: Absolute value of $s^*$ should exceed some threshold so

- Principle:
  - Start by finding the analytical expressions of the eigen values of the Hessian

\[
s_1(\sigma, w, r) := \frac{\partial^2 g_\sigma(0, 0)}{\partial u^2} = \frac{-2hw}{\sqrt{2\pi}\sigma^3} e^{-\frac{w^2}{2\sigma^2}} (2\Phi_\sigma(r) - 1)
\]

\[
s_2(\sigma, w, r) := \frac{\partial^2 g_\sigma(0, 0)}{\partial v^2} = \frac{-2hr}{\sqrt{2\pi}\sigma^3} e^{-\frac{r^2}{2\sigma^2}} (2\Phi_\sigma(w) - 1)
\]

- so estimated off-line from 500 sampled imagettes of insects of interest as the minimum value

- Such a robust criterion is much more useful over all when combined with the Connected Component Algorithm in order to eliminate too small CCs, however many false positives have been observed in the recorded videos

Show movie
On-line Video Processing Chain

- Trap Extraction (First frame)
- Background Subtraction
- Insect Presence Detection
- Image Patch Voting
- Local Insect Tracking
- Insect Recognition

Statistical Analysis of the Counting Results
A. Subdivision of a video frame into small patches (1/2)

• Insects arrive in a **sparse** way (i.e. as rare events) in the sticky trap

• Hence, **parsimonious** use of the insect detector via:
  - Subdivision of the extracted trap into $k \times n$ patches
  - Account for border effects via patch overlapping
  - At most one image patch is processed by the detector to detect any recently trapped insect of interest
A. Subdivision of a video frame into small patches (2/2)

- **Principle of image patch voting**

  - **Background subtraction:** Pixel based, Motion detection
  - **Insect presence detection:** Pixel based, Model of color insect intensity learned from sample data
  - **Construction of binary image of votes:** All pixels that passed the insect presence detection test
  - **Patch election:** Patch with maximum votes if it exceeds some pre-defined threshold
B. Background subtraction (1/4)

- Standard method for real time video-processing in video-surveillance
- Used for motion detection, object segmentation and object tracking
- Challenges:
  - Noise, acquisition artifacts (e.g. image compression)
  - Robustness (adaptiveness) to light changes:
    - Gradual light changes: daylight changes
    - Relatively rapid lightening: Motion of clouds
  - Shadows: Motion of plant leaves, humans, insects
  - Quickness:
    - In order to adapt to frame acquisition rate
    - Solution: generally pixel based, post-processing
B. Background subtraction (2/4)

• Some existing background subtraction techniques:

1. Thresholding techniques
   • Simple image differentiation + Global thresholding

   $$|frame_i - frame_{i-1}| > Th$$

   • Running average algorithm (Wren et al. 1997): Adaptive estimation of a mean image using a running average scheme, and global thresholding of the difference between a new frame and the mean image

   $$|frame_i - background_i| > Th$$

   $$B_{i+1} = \alpha * F_i + (1 - \alpha) * B_i$$

   • Advantage: Quickness
   • Disadvantage: Don’t handle multi-modal backgrounds (e.g. in presence of shadows due to the swaying of the neighboring plant leaves)
B. Background subtraction (3/4)

1. Probabilistic techniques: Estimate a pdf of the intensity of a pixel over recent history, and foreground corresponds to “pdf<t”

2.a. Kernel density estimators (El Gammal et al. 2000):
   - Use a kernel density estimator for estimating the pdf of a pixel
   - Efficient but: too slow + big memory requirements

2.b. Mixture of Gaussians (MoG) (Stauffer et al. 1999)
   - Model a pixel as a mixture of m Gaussians \( \mu_i, \sigma_i, \omega_i \)
   - Very quick adaptive update schemes for each \( \mu_i, \sigma_i, \omega_i \)
   - Classify pixel as Foreground its intensity lies far by more than 2.5 \( \sigma_i \) times from each comp. “i”
   - Advantage: handle multi-modal backgrounds (e.g. in presence of shadows due to the swaying of the neighboring plant leaves)
   - Disadvantages: Choice of m, many magic parameters to tune
B. Background subtraction (4/4)

- A novel background subtraction algorithm
  - To handle 2 gray states of the background model (periodic shadows, etc.)

\[
I_t(x) = \begin{cases} 
\mu_1(x) + \zeta_t(x) \\
\mu_2(x) + \zeta_t(x)
\end{cases}
\]  

or

\[
I_t(x)^2 + a_t(x)I_t(x) + b_t(x) = \zeta_t(x)
\]

- Quick update scheme of pixel background state equation:

\[
\begin{align*}
\alpha_t(x) & := \frac{\mu_t^2(x) - \mu_1^2(x)\mu_t^2(x)}{\mu_t^2(x) - (\mu_t^1(x))^2} \\
\beta_t(x) & := \frac{\mu_1^1(x)\mu_1^2(x) - (\mu_t^1(x))^2}{\mu_t^2(x) - (\mu_t^1(x))^2}
\end{align*}
\]

\[
\sigma_t^2(x) := (1 - \alpha)\sigma_{t-1}^2(x) + \alpha(I_t(x)^2 + \alpha_{t-1}(x)I_t(x) + \beta_{t-1}(x))^2
\]

- Foreground pixel detection test
- Foreground detection threshold “t” is learned from videos
- Extension into RGB intensity is straightforward (one model w.r.t. each channel)
- Proved to be satisfactory when combined with presence detection test
C. Pixel-based insect presence detection:

- **Principle:**
  - Learn off-line from insects sampled the color space of the insects of interest
  - Principal Components Analysis: Estimate the statistical model of insect RGB colors

- **Apply presence detection test for all pixels where significant change has been observed by the background subtraction algorithm**

- **Insect presence detection test**
  \[
  \frac{|\rho_{v_1}(d)|}{\sigma_1} \leq t_1 \quad \text{and} \quad \frac{|\tau_{v_1}(d)|}{\sigma_2} \leq t_2
  \]
D. Insect local Frame-to-Frame tracking (1/2)

• Motivation: An insect might displace from its initial location
• Goal: Detect an insect only once and maintain its location in subsequent “T” frames using a Track-Before-Detect Strategy
• Requirements:
  ➢ Accuracy: To not count the same insect many times if one misses its track in subsequent frames
  ➢ Quickness: To be able to run it on-line
• Choice of the feature vector
  ➢ Intensity based: compute a square bounding box around insect and sort gray values in ascending order, subsample into N points
  ➢ Advantages: easy to compute, invariance under rigid transformations (translation and rotation)
D. Insect local Frame-to-Frame tracking (2/2)

• Tracking algorithm: A slight modification of correlation based tracking

1. For each pixel in the neighborhood of the insect (typically a window of some fixed size), compute a correlation coefficient between initial feature vector and feature vector extracted from a square window of same size as the initial bounding box

2. Acceleration: Use information gained from the background subtraction algorithm: Only windows containing a minimal number of pixels detected as foreground are considered

Finally, choose the new insect location as the window which maximizes the correlation coefficient
Conclusion

• We showed the feasibility of a video-surveillance system for pest monitoring and implemented a full prototype (video-acquisition, GUI, Video-Processing)

• A novel on-line vision solution combining video-processing technique combining some original image and video-processing algorithms along with learning from sample videos

• System currently tested off-line and showed good accuracy in terms of false positive rate (less than 1%) and false negative rate (less than 5%)

• Main inconvenient : Slowness under changing light conditions

Future work :

• Short term :
  – Large scale validation (mainly on the whitefly species )
  – Deploy the system in a new experimental site at INRA Avignon (tomato culture)
    ➢ An enigneer is currently working on this issue since almost two months
  – Extension of the approach to other insect species and cultures (e.g., greenfly)
    • A new intern is hired for this
  – Improvement of the video-quality (a new camera has been purchased in this purpose)
  – Embedding of the low-level video-aglorithms inside the cameras