Journal of Animal Science and Biotechnology

RESEARCH Open Access

Evaluation of a laying-hen tracking algorithm based on a hybrid support vector machine

Cheng Wang[1], Hongqian Chen[2,3], Xuebin Zhang[1] and Chaoying Meng[1*]

Abstract

Background: Behavior is an important indicator reflecting the welfare of animals. Manual analysis of video is the most commonly used method to study animal behavior. However, this approach is tedious and depends on a subjective judgment of the analysts. There is an urgent need for automatic identification of individual animals and automatic tracking is a fundamental part of the solution to this problem.

Results: In this study, an algorithm based on a Hybrid Support Vector Machine (HSVM) was developed for the automated tracking of individual laying hens in a layer group. More than 500 h of video was conducted with laying hens raised under a floor system by using an experimental platform. The experimental results demonstrated that the HSVM tracker outperformed the Frag (fragment-based tracking method), the TLD (Tracking-Learning-Detection), the PLS (object tracking via partial least squares analysis), the MeanShift Algorithm, and the Particle Filter Algorithm based on their overlap rate and the average overlap rate.

Conclusions: The experimental results indicate that the HSVM tracker achieved better robustness and state-of-the-art performance in its ability to track individual laying hens than the other algorithms tested. It has potential for use in monitoring animal behavior under practical rearing conditions.

Keywords: Computer vision, Laying hens, Locomotion tracking, Support vector machine

Background

The behavior of animals is an important indicator of their welfare [1, 2]. Animal behavior is typically monitored through manual observation which requires substantial manpower and cannot always guarantee accuracy [3]. The demand for methods to automatically monitor animal behavior and track their movement has recently been increasing thereby promoting the initiation of related research [4].

Previous studies of animal behavior have focused on two main objectives, namely the identification of specific behavior and the tracking of animal movement. With respect to behavioral identification, the appearance of animals varies widely depending on their location which renders image processing and interpretation very difficult [5]. Some researchers have identified the behavior of animal groups through visual techniques such as monitoring the weight distribution in poultry flocks [6, 7], the spatial distribution of pigs [8, 9], the distribution of broilers [10], and the trajectory of a flock of poultry [11].

Monitoring the behavior of a particular animal in a group requires information obtained from tracking the specific animal and this can be achieved by limiting the animal's activity to ensure that it remains in an appropriate location without other animals in its vicinity. This idea has been applied to monitor a pig's weight [12] and back fat levels [13] and to monitor a laying hen's activities [14].

With respect to motion tracking, Computer Vision Technology was first used in 1997 to track animal behavior [15]. In 1998, Sergean et al. [16] developed a tracking system using color information and segmented individual birds using contour information. Currently, Ellipse Fitting is the most common approach used to track laying hens. Fujii et al. [17] used a method based on particle filters for tracking multiple hens. However, the particle filters lost track of the hens when sudden

* Correspondence: mcy@cau.edu.cn
[1]College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
Full list of author information is available at the end of the article

quick movements were made. The method which was proposed by Kashiha [18] had a superior performance for tracking individual laying hens in an image area but was unable to identify and track an individual laying hen in a flock. To solve this problems, Nakarmi et al. [19] installed a RFID (Radio Frequency Identification) antenna array at the bottom of a cage and attached RFIDs to the feet of hens' to determine their location for further tracking in the distance image. Although this method can achieve suitable tracking results, it is very limited in its application. It is not conducive to practical application and wearing the RFID can lead to discomfort for the hens which in turn may alter their behavior.

To address the challenges discussed above, a new laying-hen tracking algorithm, based on the Hybrid Support Vector Machine (HSVM) model has been proposed as a method to track a single hen within a flock raised under a floor system in real time with high robustness. The objective of this experiment was to compare the ability of this method to track individual laying hens in a flock with 5 other commonly used algorithms.

Methods

Experimental pen design and setup

This study was approved by the Animal Care and Use Committee of China Agricultural University (Beijing, China). As tracking targets, six 20-week-old Hyline Brown laying hens weighing an average of 1.4 kg were selected for study. The hens were allowed a 2 wk acclimation period before commencing data collection.

A 1.2 m × 1.5 m pen (Fig. 1a, b) was constructed to house the birds (Fig. 1c). On two sides of the pen, LED lighting was used to illuminate the test area from 0500 h to 2100 h every day to ensure that the intensity of illumination in the pen region was approximately 15–20 lux. The hens were fed twice a day at 0900 h and 1700 h and their eggs were collected at 1700 h every day. Manure was removed daily and the barn temperature was maintained about 20 °C.

The height of the cameras used to collect video (Launch, LC5505E7-C83R) was set at 2.2 m. Videos were operated from 0500 h to 2100 h. Over 500 h of video were obtained during the subsequent 30 d. Ten 3-min fragments out of the 500 h of video were randomly chosen to validate the tracking algorithm and 778 images in the video fragments were randomly chosen and manually labeled.

Initialization

The tracking algorithm consisted of three steps including initialization, tracking and updating. For initialization, the contour area of the target was manually marked and the rotation method was used to obtain the size of the minimum outer rectangle of the contour area. This minimum outer rectangle was represented as $T_0\{w_0,h_0,a_0,c_0\}$, where w_0 corresponded to the width of T_0, h_0 represented the height of T_0, a_0 was the angle between T_0 and the x-axis, and c_0 was the center of T_0. This rectangle was the initial tracking rectangle and the width and height of each sample was consistent with it.

Binary HSVM model ($HSVM_b$)

The HSVM model consisted of a one-class model, a binary classification model and a regression model. Around the initial tracking rectangle, the three types of HSVM were sampled as follows. Firstly, the Binary Classification Support Vector Machine ($HSVM_b$) model was established [20]. The binary model is often used for the tracking-by-detection strategy [21, 22] used in object tracking. However, this method results in a fuzzy boundary between positive and negative samples. To handle

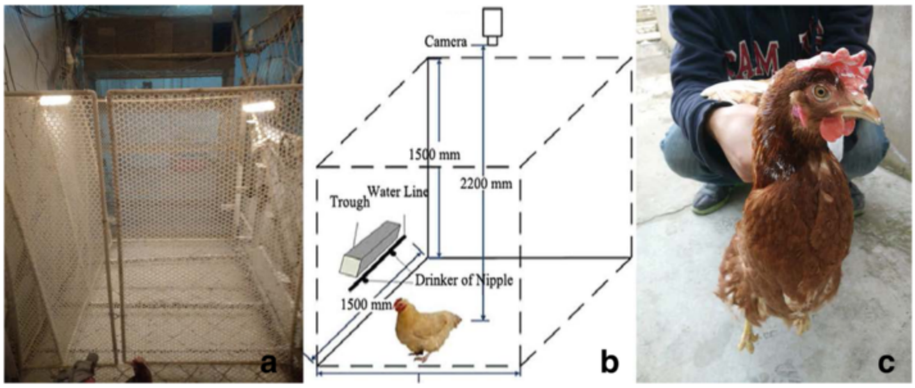

Fig. 1 A schematic drawing and photograph of the experimental pen and observation objects. (**a**) Photograph (**b**) Schematic (**c**) Observation objects

this problem, the regression model aids in locating the target more accurately to avoid drift.

For the HSVMb, the positive and negative samples were expressed as $\{x_i, y_i\}$, where $y_i \in \{+1,0\}$ was the label of sample x_i. If $y_i = 1$, x_i was a positive sample, and x_0 denoted the sample in the initial tracking rectangle. $l(x_i)$ denoted the location of sample x_i, and $l(x_0)$ denoted the location of T_0. The distance-based rule was used to select training samples [21, 23]. If $||l(x_i)-l(x_0)|| < d_1$, $y_i = 1$, and if $d_2 < ||l(x_i)-l(x_0)|| < d_3$, $y_i = 0$, (Fig. 2a) where

$$d_2 = \sqrt{W^2+H^2} \Big/ 2, d_1 = \sqrt{20}, \tag{1}$$

$$d_3 = 2\sqrt{W^2+H^2}(W = w_0,\ H = h_0)$$

To extract the histogram of orientation gradients, 50 positive and 50 negative samples were randomly selected according to the above rules. In the HSVM, the window size for the histogram of orientation gradient was 16 × 16 pixels and the cell size was 4 × 4 pixels. One block consisted of 4 cells and strided each cell once with 9 orientations. All of the samples selected for feature extraction were normalized to the size of the window. With the features and training pairs $\{x_i, y_i\}$, the binary HSVM model was obtained. The confidence score of a new candidate sample x_i was calculated by:

$$conf_b(x) = \sum_i a_i y_i k_b(x_i, x) \tag{2}$$

where a_i was the Lagrange Multiplier and $k_b(x_i, x)$ was the Kernel Trick [24].

Regression SVM model (HSVM$_r$)

For HSVM$_r$, all of the samples satisfying $d_1 < ||l(x_i)-l(x_0)|| < d_2$ were selected as training samples (Fig. 2b). The bounding box overlap area ratio was chosen to generate the regression function value y_i of sample x_i, which has been widely used to evaluate the accuracy of object detection [23]:

$$y_i = \frac{area(x_o \cap x_i)}{area(x_o \cup x_i)} \tag{3}$$

where x_0 denoted the initial tracking rectangle. Following this principle, 50 training samples were randomly selected to obtain the regression HSVM model. For any candidate in region x, its confidence score $conf_r(x)$ was calculated as follows:

$$conf_r(x) = \sum_i (a_i - a_i^*) k_r(x_i^T, x) \tag{4}$$

where a_i and a_i^* were the Lagrange Multipliers and $k_r(x_i^T, x)$ was the Kernel Trick [24].

One-class support vector machine (HSVM$_o$)

The one-class HSVM was the third model. The one-class model can be considered as an appearance model and can distinguish between individual layers [24]. Consequently, during the tracking stage, the confidence score of the candidate samples, chosen according to the tracking strategy used, was calculated using the HSVM model after feature extraction. The candidate region corresponding to the highest score was the tracking result of the current frame (Fig. 2c). After obtaining the tracking result for the current frame, we decided whether or not it was necessary to re-sample for model re-training in order to adapt to changes in target appearance.

One difference between the HSVM$_o$ and the first two models was that it used the entire tracking result region of each previous frame as the training sample. The confidence score of a candidate sample x_i was calculated as follows:

$$conf_o(x) = \sum_i a_i k_o(x_i, x) \tag{5}$$

where a_i was the Lagrange Multiplier and $k_o(x_i, x)$ was the Kernel Trick [24].

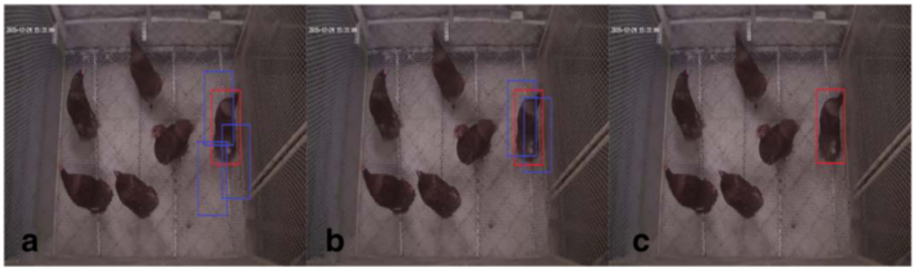

Fig. 2 Example of sampling by the three types of support vector machines. (**a**) SVM$_b$ (**b**) SVM$_r$ (**c**) SVM$_o$

After obtaining these three sub-models, the confidence score of a candidate sample x_i was calculated by

$$conf(x) = {(w_o * confn_o(x) + w_r * confn_r(x) + w_b * confn_b(x))} \Big/ {w_o + w_r + w_b} \tag{6}$$

where $confn_o(x)$, $confn_r(x)$, and $confn_b(x)$ were the results after normalizing $conf_o(x)$, $conf_r(x)$, and $conf_b(x)$ into the range [0,1]. w_o, w_r, and w_b, corresponded to the weights of each sub-model, respectively. The weights of each sub-model determined the relative contribution of each HSVM. HSVMb, adopted the binary classification, and was robust to changes in bird pose and therefore it worked the best for monitoring preening and flapping of wings for example. HSVMr effectively solved the drift problem. It had the best results for when the test hens were close to each other. HSVMo was not sensitive to a fast-changing background and therefore had good performance to monitor sudden movements from the hens [24]. Considering the adaptation of the different support vector machines to different scenarios and the results of repeated attempts, w_o, w_r, and w_b were set to 0.3, 0.6, and 0.1.

Tracking

In the tracking phase, the candidate samples were obtained around the tracking object. The model scoring was applied to select the best tracking results. The specific process was as follows:

(a) The tracking result of the previous frame was set as the initial target area $T_o\{w_o, h_o, a_o$, and $c_o\}$ of the current frame.

(b) c_o was set as the center of rotation. The target area was rotated h times in clockwise and counterclockwise directions, respectively. Each rotation was deflected by k degrees. If the coordinate of point X was (x,y) before the rotation, it became (x',y') after the rotation and the mapping formula was

$$y' = (x - x_0) \times sin\left(a_0 + (-1)^i h \times k\right) + \left(y - y_0\right) \times cos\left(a_0 + (-1)^i h \times k\right) + y_0 \tag{7}$$

$$x' = (x - x_0) \times cos\left(a_0 + (-1)^i h \times k\right) - \left(y - y_0\right) \times sin\left(a_0 + (-1)^i h \times k\right) + x_0 \tag{8}$$

where the coordinate of c_0 was (x_0, y_0). If the rotation direction was clockwise, i = 1; otherwise i =2.
There were a total of 2×h + 1 candidate regions. After the features were extracted from these regions, the HSVM model was used to calculate their confidence score. The candidate region with the highest score was chosen as the best tracking region $T_a\{w_a, h_a, a_a, c_a\}$, with respect to the angle. In the current experiment, h was set to 5 and k was set to 3;

(c) T_a was expanded m times to obtain the shift search area $T_m\{w_m, h_m, a_m, c_m\}$, where $w_m = m \times w_a, h_m = m \times h_a, a_m = a_a$, and $c_m = c_a$. The search box $T_s\{w_s, h_s,$

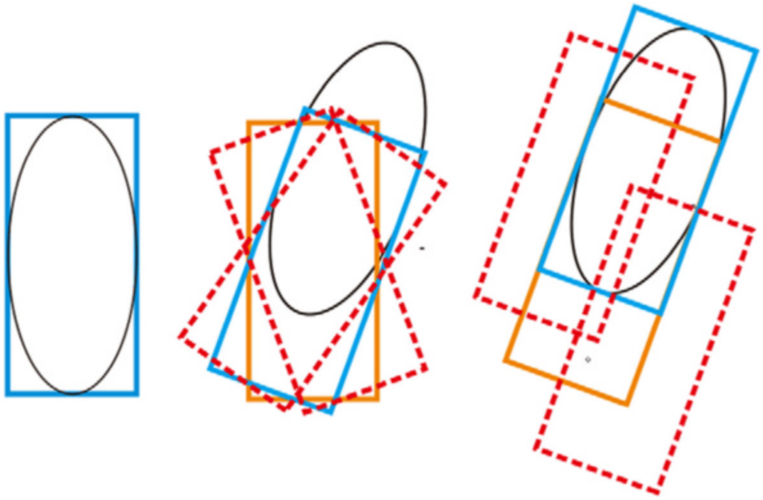

Fig. 3 Schematic diagram of the tracking process. The tracking object is indicated by an ellipse; the blue box represents the best tracking area of the current step; the orange box represents the location of the tracking box in previous steps; the red dashed boxes represent the candidate regions. The best region is selected from the candidate regions

a_s, c_s} was used to search the entire shift search area, where the initial value of the search box was $w_s = w_a$, $h_s = h_a$, and $a_s = a_a$. If the coordinate of c_a was (x_a,y_a) and the coordinate of c_s was (x_s,y_s), then

$$x_s = -0.1 \times W \times cos\alpha_a + 0.1 \times H \times sin\alpha_a + x_a \tag{9}$$

$$y_s = -0.1 \times W \times sin\alpha_a - 0.1 \times H \times cos\alpha_a + y_a \tag{10}$$

The search box maintained the same size and angle during the search process, while displacing it by M and N steps in the indicated direction along the width and height of the search area, respectively. When the search box was moved i times along the width and j times along the height, w_s, h_s, and a_s remained unchanged, and the coordinates of c_s were calculated as follows:

$$x' = \left(-(m-1) \times \frac{1}{2} \times W_a + {}^{(m-1)}\Big/_{M \times W_a \times i}\right) \times \cos a_a - \left(-(m-1) \times \frac{1}{2} \times H_a + {}^{(m-1)}\Big/_{N \times H_a \times j}\right) \times \sin a_a + x_a \tag{11}$$

$$y' = \left(-(m-1) \times \frac{1}{2} \times W_a + {}^{(m-1)}\Big/_{M \times W_a \times i}\right) \times \sin a_a + \left(-(m-1) \times \frac{1}{2} \times H_a + {}^{(m-1)}\Big/_{N \times H_a \times j}\right) \times \cos a_a + y_a \tag{12}$$

Thus, there were a total of M×N regions. After extracting the features of these regions and scoring them using the HSVM model, the candidate region with the highest score was selected as the best region, with respect to displacement (which was an initial target region of tracking). In this study, m = 1.2, M = 5, and N = 5.

The steps (b) and (c) were alternated until the two adjacent quasi-tracking areas coincided. At this time, the corresponding tracking box became the tracking area of this frame image (Fig. 3).

Because histogram of orientation gradient feature extraction is relatively time-consuming, the displacement and angle of laying hens were tracked separately. Firstly, the algorithm tracked the change in the angle and subsequently the change in the displacement, and was iterated until there was no more movement. In this way, the number of sampling iterations was effectively reduced. This method had no significant impact on the final results and effectively improved the real-time performance of the algorithm. For instance, in an iterative process, the number of sampling iterations of the tracking strategy was M×N + 2H + 1, while this number increased to (M×N)×(2H + 1) if the displacement and angle were tracked simultaneously.

Updating

Because a hen uses a non-rigid body motion, its appearance may change significantly during movement, especially if it turns, or if some of its body is partially obscured. To accommodate the hens' changing appearance during movement, the model must be updated.

The degree of change in appearance had to be calculated after the end of each frame of video tracking to determine if it required updating [24]:

$$d(x_{cur},, x_j) = 1 - \max \frac{\langle x_{cur},, x_j \rangle}{\|x_{cur}\| \bullet \|x_j\|} \tag{13}$$

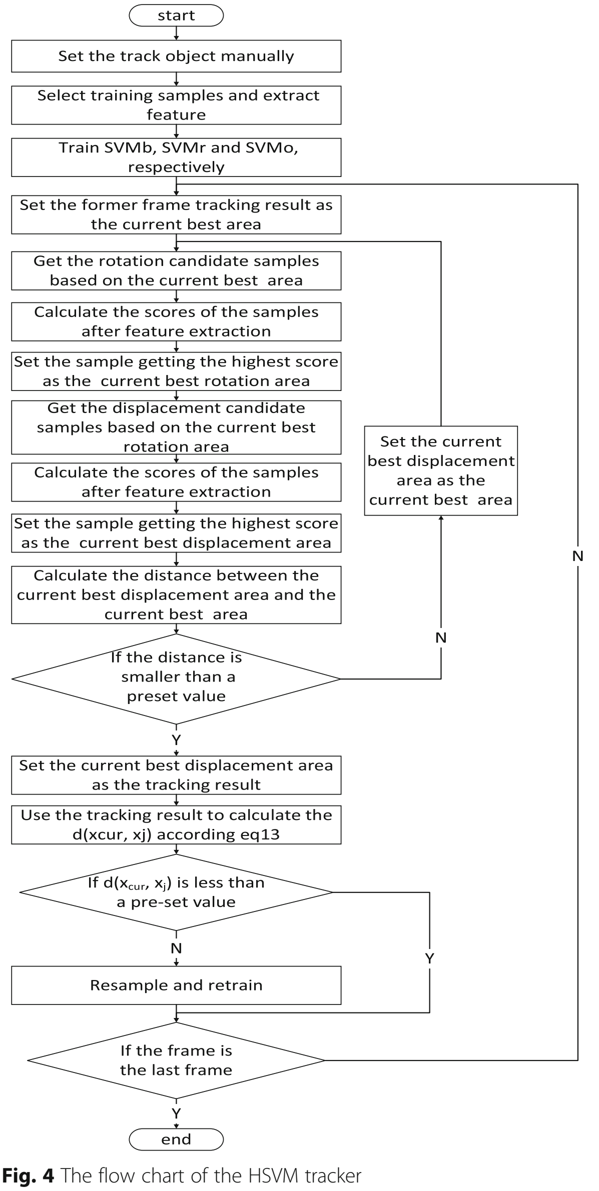
start
Set the track object manually
Select training samples and extract feature
Train SVMb, SVMr and SVMo, respectively
Set the former frame tracking result as the current best area
Get the rotation candidate samples based on the current best area
Calculate the scores of the samples after feature extraction
Set the sample getting the highest score as the current best rotation area
Get the displacement candidate samples based on the current best rotation area
Calculate the scores of the samples after feature extraction
Set the sample getting the highest score as the current best displacement area
Calculate the distance between the current best displacement area and the current best area
If the distance is smaller than a preset value
N
Set the current best displacement area as the current best area
Y
Set the current best displacement area as the tracking result
Use the tracking result to calculate the d(xcur, xj) according eq13
If d(x_cur, x_j) is less than a pre-set value
N
Resample and retrain
Y
If the frame is the last frame
N
Y
end

Fig. 4 The flow chart of the HSVM tracker

In the above formula, x_{cur} was the characteristic value of the tracking result of the current frame and x_j was the characteristic value of the previous tracking results of each frame.

If $d(x_{cur}, x_j)$ was less than a pre-set value (0.05 in our experiment), the data was re-sampled and then retrained for the model. The re-sampling rules were as follows:

(a) For the binary HSVM, the image area corresponding to the image tracking box was taken as a positive sample to be inserted into the queue of 40 positive samples using the first-in-first-out strategy. Sometimes the target was blocked for a considerable duration of time and all of the positive samples corresponded to the blocked target. This could have resulted in drift problems. This problem was solved by reserving the 10 initial positive samples. In addition, 50 negative samples were randomly selected to replace all of the former negative samples.

(b) For the regression HSVM, the sampling method for positive samples was the same as the method for the binary HSVM. For negative samples, 20 negative samples were randomly selected to replace the original negative samples.

(c) For the one-class HSVM, the sampling method for positive samples was the same as the method for the binary HSVM. The whole algorithm process is shown in (Fig. 4).

Results and discussion

The two most important criteria for the evaluation of algorithm tracking methods are real-time operation and robustness. The HSVM was implemented in OpenCV on a personal computer with a 3.50GHz Intel® Core™ i2-4150. It achieved an average speed of about 9.1 frames per second.

One Hyline Brown hen was chosen from the 6 observation objects as the tracking target. HSVM was compared with 5 other algorithms including Frag [25], TLD [26], PLS [27] (these three algorithms can all be downloaded from the homepage of the original author), MeanShift, and the Particle Filter Algorithm (these two are widely used classical algorithms). Each of these algorithms were used to track the target hen in the experimental video. Three experiments with 3 different randomly-selected tracking targets were conducted and the 6 algorithms were compared in these 3 experiments. The results are shown in (Fig. 5).

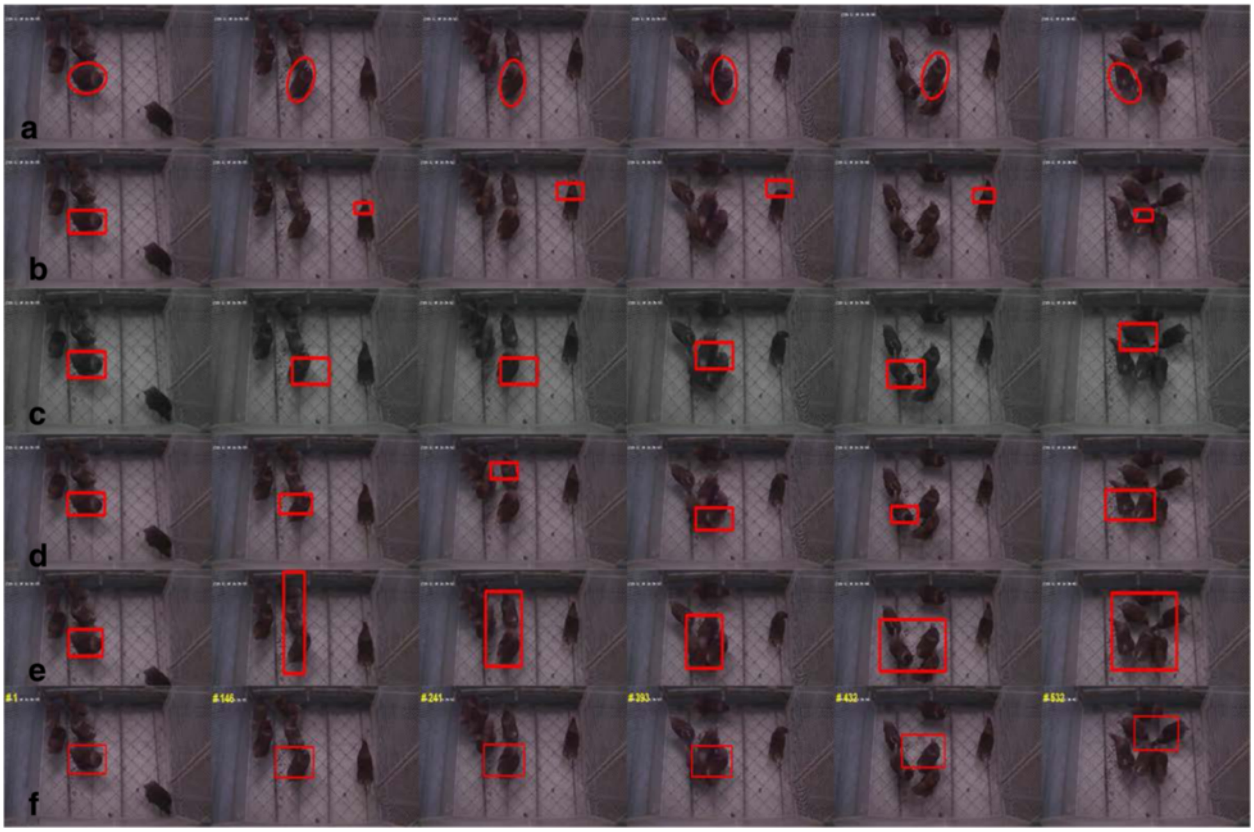

Fig. 5 Experimental results of the six algorithms: (**a**) HSVM; (**b**) TLD; (**c**) Frag; (**d**) Particle filter; (**e**) MeanShift; (**f**) PLS

To assess the robustness of the algorithm, the overlap rate (OR) was used to quantify the tracking accuracy. The overlap rate was calculated as:

$$OR = \frac{area(R_t \bigcap R_l)}{area(R_t \bigcup R_l)} \qquad (14)$$

where R_t represented the results of the tracking and R_l represented the ground truth.

The overlap rates were calculated for the 6 aforementioned algorithms (Fig. 6). The vertical axis of the statistical graph represented the overlap rate. Higher overlap scores indicated more accurate tracking while an overlap rate of 0 indicated that the algorithm completely lost the tracking targets. Figure 6a shows that for most frames, HSVM maintained an overlap rate of approximately 0.8.

An aggregation of the laying hens occurred during the 430th–600th frames. The hens' mutual occlusion sent the overlap rate on a downward trend but the algorithm

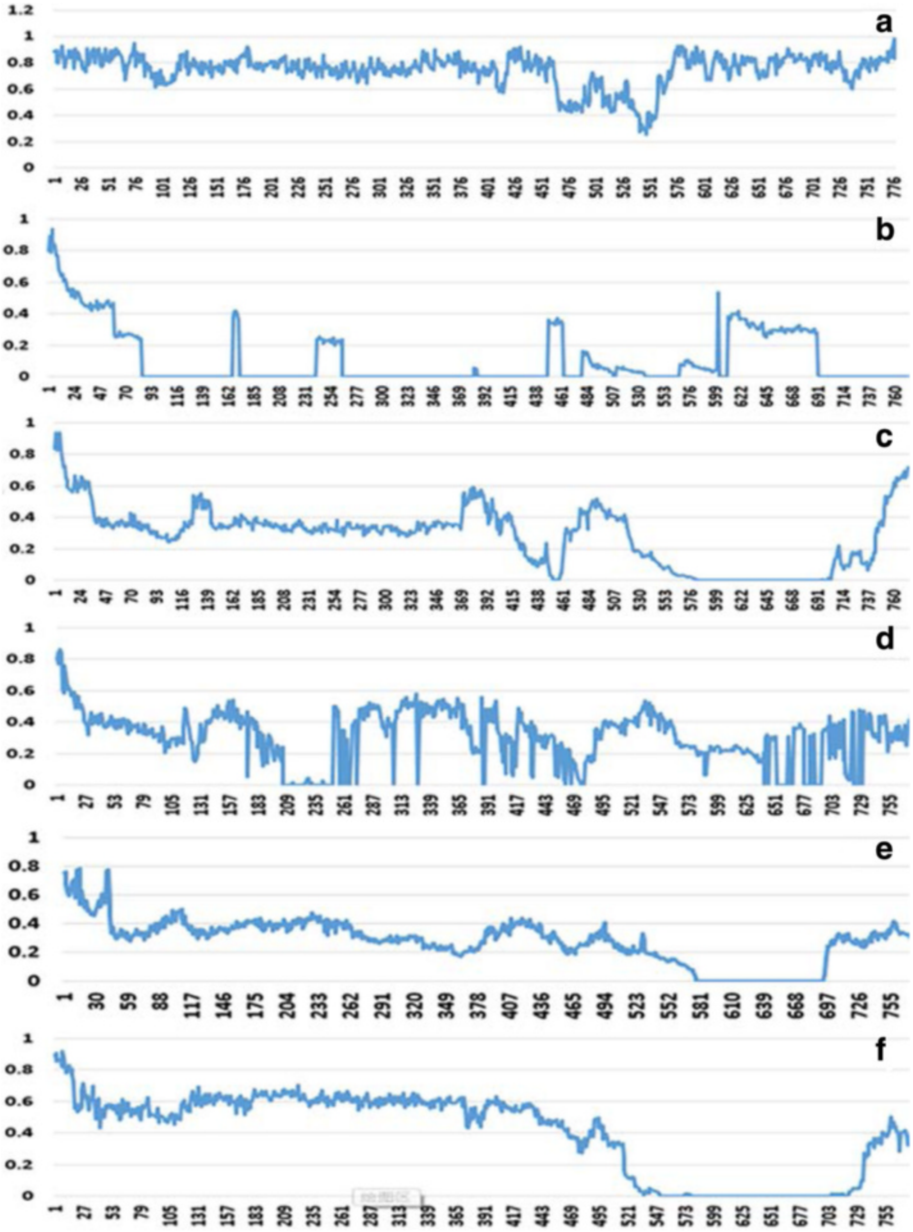

Fig. 6 Overlap rate of the six algorithms: (**a**) HSVM; (**b**) TLD; (**c**) Frag; (**d**) Particle filter; (**e**) MeanShift; (**f**) PLS. The vertical axis of the statistical graph represents the overlap rate. Higher overlap rate scores indicate more accurate tracking, while an overlap rate of 0 indicates that the algorithm completely lost the tracking targets. The horizontal axis of the statistical graph represents the frame number of the images which have been labeled

self-adjusted to recover an overlap rate of approximately 0.8. Figure 6b shows the statistical graph for the TLD algorithm.

The overlap rate curve dropped significantly at the beginning, indicating that the drift of the tracking box increased until the tracking box missed the target. The tracking box only rebounded to the target for a short period of time in the middle part of the frames. Figure 6c shows the graph for the Frag algorithm. The overlap rate curve decreased until the overlap rate was approximately 0.4 because the target hen kept changing its direction of movement. The curve then maintained this value for some time. After the 430th frame, the overlap rate curve declined again until the tracking box missed the target because of the aggregation of hens.

The Particle Filter Algorithm lost and retrieved the target frequently during the tracking process. As a result, the value of its overlap rate varied between 0 and 0.5, as shown in Fig. 6d, but it quickly recovered the target hen each time it lost it. Figure 6e shows that the MeanShift Algorithm tracking boxes expanded easily when the target hen got close to other laying hens resulting in the decline of the overlap rate curve. When the hens aggregated around the 430th frame, the tracking box simply expanded instead of losing the target. Therefore, after the 430th frame, the overlap rate curve did not suffer an obvious drop. The tracking box lost the target and stayed on the flock of hens when the target hen left the flock. Subsequently, the tracking box was transferred to other laying hens until the target hen and tracking box coincided again. The overlap rate curve of the PLS algorithm showed relatively stable performance, overall, and the value of overlap rate was approximately 0.6. Even so, the curve began to decline around the 430th frame until the tracking box lost the target.

From the figures described above, each algorithm adapted to different situations in the movement of laying hens. The average overlap rate is shown in Table 1 according to the different scenarios in the 778 images.

Table 1 shows that HSVM obtained a higher average overlap rate than the other algorithms both with respect to the total average overlap rate and for the different particular scenarios. The value of the overall average overlap rate was 35 % higher than the highest value among the other algorithms. When tracking a single target in a multi-hen mutual occlusion situation (the most challenging scenario), HSVM's average overlap rate was 68 %, which was 41 % higher than the highest value attained for the other algorithms. HSVM was relatively stable with the average overlap rate maintained between 68 and 79 % across the specific cases and the overall average. The PLS algorithm attained the best performance among the contrast algorithms because the PLS was able to model the correlation of target appearance and class labels due to its capacity for both dimensionality reduction and classification [27]. The value of the average overlap rate for the changing of direction, two hens' mutual occlusion, and preening scenarios was 55, 61 and 62 % respectively. However, PLS performed poorly in handling the heavy occlusion, which can easily and quickly change the appearance of targets [28]. In the situation of multiple hens' mutual occlusion, PLS lost the target hen for some frames resulting in a drop in the average overlap rate to 23 %. For the situation of multiple hens' mutual occlusion, the best performance (excluding that of HSVM) was achieved by the Particle Filter Algorithm, whereby the average overlap rate only reached 27 %.

Table 1 Average overlap rate for the six algorithms conducted for different scenarios

Average overlap rate	HSVM	TLD	Frag	Particle Filter	MeanShift	PLS
Change of direction	0.79	0.23	0.36	0.36	0.40	0.55
Two hens mutual occlusion	0.78	0.03	0.39	0.37	0.37	0.61
Preening	0.75	0.05	0.33	0.21	0.36	0.62
Multi hens mutual occlusion	0.68	0.09	0.17	0.27	0.18	0.23
All frames	0.75	0.11	0.28	0.30	0.28	0.40

The TLD algorithm used the optical flow method to track the object, which meant the following three conditions had to be satisfied. First, the change of luminance in the different frames should be very small. Secondly, the content of two adjacent frames should change very slowly. Finally, the projections of nearby image points were nearby points and shared similar speed [29]. The lighting in our hen house was not uniform and could not be kept stable. Moreover, hens often made sudden and quick movements such that the average overlap rate of TLD was only 11 %. The reason is that the true target was blurred, and it was difficult for the TLD to distinguish it from the background [30].

The MeanShift tracker had the advantage of low complexity, but it also failed with fast motion, illumination changes, cluttered background and occlusion [31, 32]. The average overlap rate of the MeanShift tracker was only 17 % higher than that of TLD. The Particle Filter Algorithm tracked the object by predicting its location in the next frame. It worked well when the object was briefly blocked. However, if the occlusions lasted for a longer duration, the tracking was more likely to fail [33]. Furthermore, the Particle Filter Algorithm lost the target during quick or sudden movements [17]. Thus, the average overlap rate of the Particle Filter Algorithm was similar to that of the MeanShift tracker. The Frag can cope with many different situations due to the use of local appearance models [34]. But Frag performed poorly

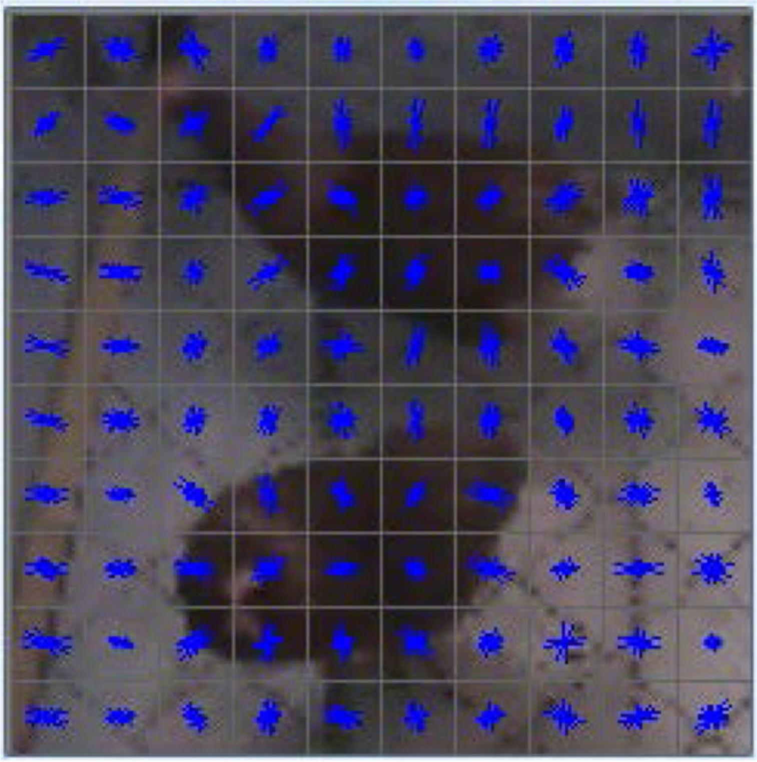

Fig. 7 Visualization results of the histogram of orientation gradient feature used in the HSVM track

in this experiment because it could not handle drastic appearance changes [35–37], so the average overlap rate of Frag was only 28 %.

Figure 6 and Table 1 demonstrate that our HSVM tracker was superior to the classical methods and existing state-of-the-art methods, with respect to better coverage and robustness on the testing sequences.

HSVM owes its success to the following aspects. First, the algorithm used histogram of orientation gradient features to detect laying hens and this effectively described the contour of the laying hens. Secondly, a new type of tracking strategy that accounted for the laying hens' displacement and their body angle improved the tracking accuracy. Third, although the histogram of orientation gradient feature extraction was time-consuming, the algorithm still had a good real-time performance by optimizing the tracking process and reducing the number of sampling iterations.

Although the HSVM algorithm showed impressive potential, there are still areas that need improvement. The histogram of orientation gradient feature was based on the object edge gradient (Fig. 7). Thus, if the tracking object is significantly occluded for a long time, the HSVM algorithm may also lose track of the object. In this experiment, the stocking density was not too high, and this situation happened only a few times in the videos. In further research, the stocking density will be increased to explore approaches to improve the robustness of the algorithm.

Conclusions

In this paper, a laying hen tracking algorithm based on the HSVM was developed to track a single hen within a flock of hens under a floor system. The experimental results showed that the algorithm achieved better robustness and real-time performance than other comparable algorithms, indicating that HSVM has a substantial practical value in the field. Because it does not require the support of a sensor, the HSVM had better application prospects. With the tracking approach, we can classify the laying hens' behavior to achieve automatic recognition. To improve the average overlap rate in future work, we will investigate a method to adjust the size of the tracking box based on the size change of the moving tracking targets.

Acknowledgments

The authors sincerely thank Dr. Phil Thacker (University of Saskatchewan) for his help in polishing this manuscript.

Funding

We greatly appreciate the financial support provided by the Key Projects in the National Science & Technology Pillar Program during the Twelfth Five-year Plan Period (No. 2014BAD08B05). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Availability of data and materials

The datasets during and/or analyzed during the current study available from the corresponding authors on reasonable request.

Authors' contributions

MC conceived and designed the experimental plan. CW and XZ performed the experiments, and HC analyzed the data. CW wrote the paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China. [2]Network Center, China Agricultural University, Beijing 100083, China. [3]College of Water Resources and Civil Engineering, China Agricultural University, Beijing 100083, China.

Received: 8 April 2016 Accepted: 23 September 2016
Published online: 12 October 2016

References

1. Duncan I. Animal rights-animal welfare: A scientist's assessment. Poult Sci. 1981;60(3):489–99.
2. Dawkins M. Using behaviour to assess animal welfare. Anim Welfare. 2004; 13(1):3–7.
3. Abrahamsson P, Tauson R. Effects of group size on performance, health and birds' use of facilities in furnished cages for laying hens. Acta Agric Scand. 1997;47(4):254–60.
4. Noldus L, Jansen RG. Measuring broiler chicken behaviour and welfare: Prospects for automation. In: Weeks CA, Butterworth A, editors. Measuring and Auditing Broiler Welfare. Wallingford: CABI Publishing; 2004. p. 277–300.
5. Stuyft EVD, Schofield CP, Randall JM, Wambacq P, Goedseels V. Development and application of computer vision systems for use in livestock production. Computers & Electronics in Agriculture. 1991;6(3):243–65.
6. Wet LD, Vranken E, Chedad A, Aerts JM, Ceunen J, Berckmans D. Computer-assisted image analysis to quantify daily growth rates of broiler chickens. Brit Poult Sci. 2003;44(4):524–32.
7. Chedad A, Aerts JM, Vranken E, Lippens M, Berckmans JZD. Do heavy broiler chickens visit automatic weighing systems less than lighter birds? Brit Poult Sci. 2003;44(5):663–8.
8. Shao J, Xin H, Harmon JD. Comparison of image feature extraction for classification of swine thermal comfort behavior. Computers & Electronics in Agriculture. 1998;19(97):223–32.
9. Hu J, Xin H. Image-processing algorithms for behavior analysis of group-housed pigs. Behav Res Methods Instrum Comput. 2000;32(1):72–85.
10. Kashiha M, Pluk A, Bahr C, Vranken E, Berckmans D. Development of an early warning system for a broiler house using computer vision. Biosyst Eng. 2013;116(1):36–45.
11. Vaughan R, Sumpter N, Henderson J, Frost A, Cameron S. Experiments in automatic flock control. Robot Auton Syst. 2000;31(s1-2):109–17.
12. Schofield CP, Marchant JA, White RP, Brandl N, Wilson M. Monitoring pig growth using a prototype imaging system. J Agric Eng Res. 1999;72(3):205–10.
13. Frost AR, French AP, Tillett RD, Pridmore TP, Welch SK. A vision guided robot for tracking a live, loosely constrained pig. Computers & Electronics in Agriculture. 2004;44(2):93–106.
14. Leroy T, Vranken E, Van Brecht A, Struelens E, Sonck B, Berckmans D. A computer vision method for on-line behavioral quantification of individually caged poultry. Trans ASABE. 2006;49(3):795–802.
15. Sumpter N, Boyle R D, Tillett R D. Modelling collective animal behaviour using extended point-distribution models. Durham University: Proceedings of the British Machine Vision Conference; 1997. p. 110–117.
16. Sergeant D, Boyle R, Forbes M. Computer visual tracking of poultry. Computers & Electronics in Agriculture. 1998;21(1):1–18.
17. Fujii T, Yokoi H, Tada T, Suzuki K. Poultry tracking system with camera using particle filters. IEEE Trans Power Electron. 2009. doi:10.1109/ROBIO.2009.4913289.
18. Kashiha MA, Green AR, Sales TG, Bahr C, Berckmans D, Gates RS. Performance of an image analysis processing system for hen tracking in an environmental preference chamber. Poult Sci. 2014;93(10):2439–48.
19. Nakarmi AD, Tang L, Xin H. Automated tracking and behavior quantification of laying hens using 3D computer vision and radio frequency identification technologies. Trans ASABE. 2014. doi:10.13031/trans.57.10505.
20. Vapnik VN. The nature of statistical learning theory. Neural Networks IEEE Transactions. 1995;10(5):988–99.
21. Avidan S. Support vector tracking. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2001;26(8):1064–72.
22. Tang F, Brennan S, Zhao Q, Tao H. Co-tracking using semi-supervised support vector machines. Proceedings of the IEEE International Conference on Computer Vision(ICCV). 2007. doi:10.1109/ICCV.2007.4408954.
23. Everingham M, Gool L, Williams C, Zisserman A. Pascal Visual Object Classes Challenge Results. 2010. http://www.pascal-network.orgS. Accessed 15 Jan 2015.
24. Zhang S, Sui Y, Yu X, Zhao S, Zhang L. Hybrid support vector machines for robust object tracking. Pattern Recognit. 2015;48(8):2474–88.
25. Agassiz A. Robust fragments-based tracking using the integral histogram. IEEE Computer Society Conference on Computer Vision & Pattern Recognition. 2006;1:798–805.
26. Kalal Z, Matas J, Mikolajczyk K. P-N learning: Bootstrapping binary classifiers by structural constraints. IEEE Conference on Computer Vision & Pattern Recognition. 2010;238(6):49–56.
27. Wang Q, Chen F, Xu W, Yang MH. Object tracking via partial least squares analysis. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society. 2012;21(10):4454–65.
28. Yan J, Chen X, Deng D, Zhu Q. Structured partial least squares based appearance model for visual tracking. Neurocomputing. 2014;144:581–95.
29. Sun C, Zhu S, Liu J. Fusing Kalman filter with TLD algorithm for target tracking. Control Conference (CCC). 2015;3736–41. doi:10.1109/ChiCC.2015.7260218.
30. Zhong W, Lu H, Yang MH. Robust object tracking via sparsity-based collaborative model. IEEE Conference on Computer Vision & Pattern Recognition. 2012:1838–1845. doi:10.1109/CVPR.2012.6247882.
31. Phadke G. Robust multiple target tracking under occlusion using Fragmented Mean Shift and Kalman Filter. International Conference on Communications & Signal Processing. 2011;517–21. doi:10.1109/ICCSP.2011.5739376.
32. Dinh TB, Vo N, Medioni G. Context tracker: Exploring supporters and distracters in unconstrained environments. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE Trans Power Electron. 2011;2011:1177–84.
33. Abramson H, Avidan S. Tracking through scattered occlusion. IEEE Computer Society Conference on Computer Vision & Pattern Recognition Workshops. 2011:1–8. doi:10.1109/CVPRW.2011.5981674.
34. Lu H, Jia X, Yang M H. Visual tracking via adaptive structural local sparse appearance model. IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2012:1822–1829. doi:10.1109/CVPR.2012.6247880.
35. Babenko B, Yang MH, Belongie S. Robust object tracking with online multiple instance learning. IEEE Trans Pattern Anal Mach Intell. 2011;33(8):1619–32.
36. Zhong W, Lu H, Yang MH. Robust Object Tracking via Sparse Collaborative Appearance Model. IEEE Trans Image Process. 2014;23(5):2356–68.
37. Wang D, Lu H, Yang MH. Online object tracking with sparse prototypes. IEEE Trans Image Process. 2013;22(1):314–25.