Automated Detection of Cattle Mounting using Side-View Camera

Yongwha Chung[1], Dongwhee Choi[1], Heesu Choi[1], Daihee Park[1]*, Hong-Hee Chang[2], and Suk Kim[3]

[1]Dept. of Computer and Information Science
Korea University, Sejong, Korea
[e-mail: ychungy, dongwhee, chlgmltn420, dhpark@korea.ac.kr]
[2]Dept. of Animal Science
Gyeongsang National University, Jinju, Korea
[e-mail: hhchang@gnu.ac.kr]
[3]Col. of Veterinary Medicine
Gyeongsang National University, Jinju, Korea
[e-mail: kimsuk@gnu.ac.kr]
*Corresponding author: Daihee Park

Abstract

Automatic detection of estrus in cows is important in cattle management. This paper proposes a method of estrus detection by automatically checking cattle mounting. We use a side-view video camera and apply computer vision techniques to detect mounting behavior. In particular, we extract motion information to select a potential mount-up and mount-down motion and then verify the true mounting behavior by considering the direction, magnitude, and history of the mount motion. From experimental results using video data obtained from a Korean native cattle farm, we believe that the proposed method based on the abrupt change of a mounting cow's height and motion history information can be utilized for detecting mounting behavior automatically, even in the case of fence occlusion.

Keywords: IT-Agriculture, Estrus detection, Video monitoring, Mounting behavior

1. Introduction

$\mathbf{A}$utomated activity monitoring in livestock management is important because it can significantly reduce a manager's required working time [1,2]. In particular, failure to detect estrus within 20 h is a limiting factor in cattle management. The detection of estrus in cattle management is essential as it can determine an optimum artificial insemination time and thus lead to increased reproductive performance. A fundamental sign of estrus in a cattle farm is mounting behavior [3,4]. However, the traditional method of mounting detection using manual observation entails excessive cost for a large-scale farm that is supervised by minimal farm workers.

There are many commercially available "attached" sensors used to detect estrus. For example, recording-the-cow-activity method is used in livestock with an electronic pedometer attached to the cow's leg [5-7]. Cattle in estrus walk more than cattle not in estrus. To determine a walking pattern, the system analyzes the recorded walking data. An activity meter attached to the cow's neck can also be used to measure the amount of mounting activity associated with estrus [8,9]. However, these attached sensors can cause false alarms when the sensors are bumped into fences or other cows. Moreover, these sensors are relatively expensive for large-scale farms. We must reduce the managing cost in large-scale farms that employ individual sensors and eliminate the requirement of farm workers having to verify individual sensors.

In [10], we proposed a "non-attached" method to detect estrus with an audio surveillance system. In this paper, we apply computer vision techniques to mounting detection for a large-scale farm to further improve the accuracy of estrus detection. It is difficult to segment the body of a Korean native cow from an input scene captured with a "tilted-down" camera because the color of the cow's body in the scene is similar to the color of the background (*i.e.*, ground). Thus, we propose a video surveillance system with a "side-view" camera to detect mounting behavior based on motion analysis techniques.

The main idea of the proposed system is that we use a side-view camera and check the abrupt change (*i.e.*, move-up by more than 40 cm within 0.5 seconds) of a mounting cow's height with computer vision techniques. Typically, a mounting activity can be observed at 1.5~1.9 m above the ground. Therefore, we install a side-view camera at 1.7 m above the ground and check the abrupt change. That is, the checking process identifies both spatial (*i.e.*, a mounting can be observed at a height higher than that of a normal activity) and temporal (*i.e.*, a mounting can be observed at a speed faster than that of a normal activity) abnormalities with a side-view camera. At the start (end) of a mounting behavior, a mounting cow moves up (down) abruptly creating upward (downward) motion vectors with significant magnitude. These types of motion vectors can be typically detected using the Optical Flow method [11, 12], a motion estimation technique. When a mounting is partially occluded by a fence structure, however, the required motion vectors may not be fully detected by the Optical Flow method because of the occlusion. A video surveillance system using a side-view camera must be capable of resolving this *fence-occlusion problem*.

In this paper, we use the Motion History Image (MHI) method [13, 14], which can provide rich motion information. Although some portion of a mounting behavior may be missed by a partial occlusion, we can utilize the motion history information to resolve the fence-occlusion problem. That is, we define rules to detect "true" mounting behavior even in the fence-occlusion case by carefully considering the direction, magnitude, and history of the

mounting motion. To the best of our knowledge, this is the first report of a cattle mounting detection system (*i.e.*, our main contribution) using a side-view camera with computer vision techniques. The experimental results confirm that the proposed system can detect mounting behavior without the use of attached sensors. We also measured the effect of the background subtraction (*i.e.*, Gaussian Mixture Model, GMM [15,16]) for the mounting detection accuracy (*i.e.*, with/without GMM) and the effect of the Region of Interest (RoI) for the mounting detection speed (*i.e.*, with/without RoI).

The remainder of the paper is structured as follows. Section 2 describes the background knowledge for motion detection. Section 3 explains the proposed system for mounting detection, and Section 4 discusses the experimental results. Section 5 summarizes the paper.

2. Related Works

For moving-object detection in stream video data, *background subtraction* (also known as foreground detection) and *motion vector extraction* are widely used. These two techniques can detect moving objects from the difference of each pixel between the previous frame and the current frame. For background subtraction, we use the Gaussian Mixture Model (GMM) method [15] that uses a parametric probability density function represented as a weighted sum of Gaussian component densities. To extract motion vectors, we select feature points such as the corners from each image frame. These feature points can then be used to calculate two-dimensional vectors with spatial coherence between the previous frame and the current frame. However, the type of motion vectors extracted using the Optical Flow (OF) method [11, 12] may not resolve the fence-occlusion problem. The additional information necessary for resolving the fence-occlusion problem can, nonetheless, be extracted using the MHI method [13, 14].

2.1 Gaussian Mixture Model (GMM)

Typically, the frame difference technique is widely used to detect moving objects. This is a simple but efficient background subtraction algorithm identifying the difference of each pixel between the previous frame and the current frame. To detect moving objects in our experiments, open source GMM [16] was employed. GMM is used for background modeling to subtract the background using each pixel modeled as mixed Gaussian distributions.

As a brief description of GMM, it is a parametric probability density function represented as a weighted sum of Gaussian component densities. It is a probabilistic model for density estimation using a convex combination of multivariate normal densities. GMMs are estimated from training data using a convex combination of multivariate normal densities and using the iterative Expectation Maximization (EM) algorithm. GMM is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. They model the probability density function of observed data points using a multivariate Gaussian mixture density. Computation of conditional probabilities can be calculated for given test input patterns only when a model is once generated. The details of GMM can be found in [15].

2.2 Motion History Image (MHI)

MHI [13] is an efficient technique for action representation. MHI is presented as a relatively new concept where the moving parts of a video sequence can be engraved in a single image allowing the prediction of motion flow and the moving portions of the video action. This action/motion representation technique has become popular owing to its simple algorithm and

ease of implementation. Consequently, MHI representations for numerous applications and MHI technique for action recognition have become fashionable in the computer vision community [14]. MHI can describe motion in a sequence of continuous frames. The *silhouette* sequence is condensed into a gray scale image; the dominant motion information is preserved. MHI is considered in many applications related to action recognition and video analysis. MHI is calculated from motion silhouettes that can be generated with a background subtraction algorithm. The gray value of MHI encodes the temporal information of the motion. The most recent motion in a frame is encoded as the brightest gray value; older motion is represented as a darker gray value (compared to the gray value of the latest motion). MHI represents a gray value as black when there is no motion in the frame. The details of MHI can be found in [13].

3. Mounting Detection System

3.1 Requirements of Mounting Detection for Korean Native Cattle

- *Accuracy Requirement*

A mounting detection system must distinguish the mounting activity from other activities. Because the color of Korean native cattle and the ground color of a farm are similar, segmenting each individual cow from a background image (*i.e.,* ground or non-moving object) is difficult. Thus, we must consider a mounting detection system based on motion analysis techniques rather than color-based segmentation techniques.

- *Response Time Requirement*

A mounting detection system must detect a mounting in real-time from a 24-h video stream to determine an optimum artificial insemination time. Typically, computer vision techniques require intensive computational workload. To detect a mounting in real-time, the Region of Interest (RoI) must be set to a minimum to reduce the computational workload. Reducing the computational workload can also reduce the total cost for a large-scale farm.

3.2 Proposed Mounting Detection System

The principal idea (*i.e.*, our main contribution) of the proposed system is that we utilize a side-view camera and check the abrupt change of a mounting cow's height with computer vision techniques. As depicted in **Fig. 1**, a typical tilted-down camera has difficulty satisfying the accuracy and response time requirements. For some occlusion cases, it is a challenge to identify mounting and non-mounting activities. For example, in **Fig. 1**, because of the occluding cow, it is difficult to identify the activity of the occluded cow. In the left image, it is difficult to distinguish the "actual" forward walking from the "possible" forward mounting of the occluded cow. In the right image, it is problematic to discriminate the "actual" backward mounting from the "possible" backward walking of the occluding cow. Furthermore, setting the RoI in a scene captured from a typical tilted-down camera may not reduce the computational workload significantly because the possible area for a mounting in the scene is relatively wide. If we use a side-view camera and exploit the height change of the mounting cow, we can avoid many of the difficulties mentioned.

Fig. 1. Scenes captured with a typical tilted-down camera
(picture taken from the Sacheon farm)

Fig. 2 presents an overview of the proposed detection system. The 24-h/365-day video stream data are transmitted to a server via a LAN cable. The server determines if a scene has motion. The mounting detection system consists of three modules: motion detection (Module 1), motion information extraction (Module 2), and mounting verification (Module 3).
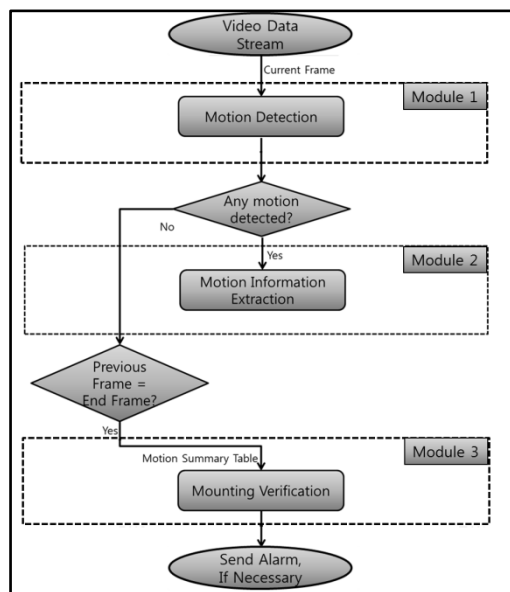

Fig. 2. Overview of the mounting detection system based on a side-view camera

In Module 1, we set an RoI to reduce the computational workload and then analyze only the moving objects within the RoI using GMM. In Module 2, we extract the motion vector and the number of motion pixels using MHI. For the purpose of explanation, we denote the *start frame* as the first frame containing motion in the RoI (*i.e.*, an object appears in the RoI). We denote the *end frame* as the last frame containing motion in the RoI (*i.e.*, the object disappears from the RoI). Note that our goal is to detect a mounting and any motion within the RoI is

detected continuously during the entire duration of the mounting (*i.e.*, from the start frame to the end frame). We denote an *activity* as the motion from the start frame to the end frame collectively. Module 2 accumulates the motion information of the activity such that Module 3 can determine if the activity is a mounting based on specific rules.

3.2.1 Motion detection

Applying computer vision techniques in a straightforward fashion to 24-h/365-day visual stream data generated from a large-scale farm could entail excessive implementation cost. Therefore, to detect a mounting activity cost-effectively, we set an RoI in the possible location of a mounting scene captured with a side-view camera. Typically, a mounting activity can be observed at 1.5~1.9 m above the ground. Therefore, we installed a side-view camera at 1.7 m above the ground, and we set the RoI (*i.e.*, a center area of 640 × 80 pixels from the acquired resolution 640 × 480 pixels) as illustrated in **Fig. 3**. Note that a scene captured with a side-view camera may include a fence structure; the system must be able to detect the mounting accurately, regardless of the fence structure.

Fig. 3. RoI for a scene captured with a side-view camera
(picture taken from the Milyang farm)

A scene from a cattle farm may have a complex background and various illuminations. Furthermore, the illumination can change gradually or abruptly based on the weather conditions. To extract the motion information accurately, we first use GMM, a background modeling technique [15]. GMM updates the training data by adding new frames and discarding old frames. Therefore, a system based on GMM can enhance performance against illumination change caused by weather conditions and efficiently detect a moving area. As indicated in **Fig. 4**, we can identify the fence structure from the detected moving area.
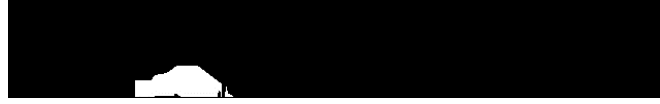
Fig. 4. GMM output of a mounting activity

3.2.2 Motion information extraction

As mentioned, the fence-occlusion problem may not be resolved by typical motion extraction techniques such as OF [11, 12]. To resolve the fence-occlusion problem, we require additional information regarding the motion. In this paper, we use the motion information generated by the MHI method [13, 14] that can provide the temporal change of a motion as illustrated in **Fig. 5**.

frame #10

frame #30

frame #80

Fig. 5. MHI output of a mounting activity

For each input frame, MHI calculates the current motion silhouette with the frame difference or background subtraction method. Then, the time-stamp is recorded in a current motion silhouette. MHI classifies each motion silhouette using the time-stamp as indicated in **Fig. 6** and computes the highest pixel coordinates of each silhouette. Using this information, we can extract the motion vector with the highest pixel positions of the oldest motion silhouette (*i.e.*, previous motion silhouette) and the latest motion silhouette (*i.e.*, current motion silhouette). In **Table 1**, for example, the first silhouette is the latest motion silhouette and the tenth silhouette is the oldest motion silhouette. With this information, the magnitude and the angle of the motion vector are 34.5 and 60.4°, respectively.
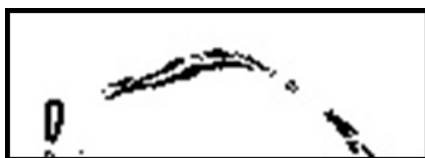

(a) Pixels of the latest silhouette   (b) Pixels of the oldest silhouette

Fig. 6. MHI output for frame #6

Table 1. Information generated by MHI for frame #6

| Silhouette in the current frame	Time stamp (sec)	Highest pixel position (x, y)
1st Silhouette	13.088	(176, 279)
2nd Silhouette	12.979	(178, 276)
3rd Silhouette	12.870	(171, 272)
4th Silhouette	12.762	(145, 265)
5th Silhouette	12.651	(143, 264)
6th Silhouette	12.541	(149, 259)
7th Silhouette	12.433	(153, 259)
8th Silhouette	12.325	(152, 256)
9th Silhouette	12.215	(151, 255)
10th Silhouette	12.104	(159, 249)
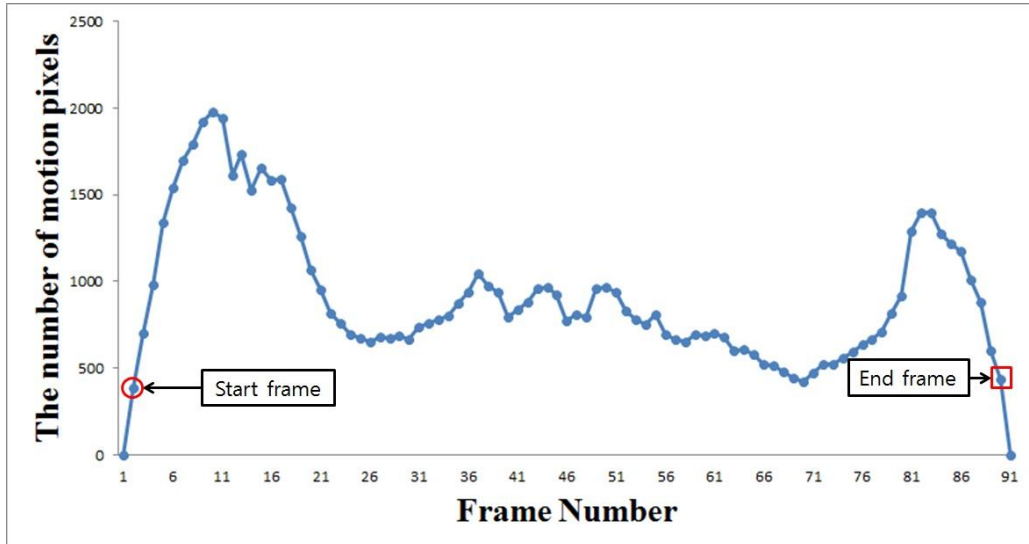
Fig. 7. Number of motion pixels of the mounting activity presented in **Fig. 5**

For each input frame, we can also compute the number of motion pixels. For example, with the MHI output of the mounting activity presented in **Fig. 5**, **Fig. 7** displays the number of motion pixels for the mounting activity.

Finally, from the motion vector and the number of motion pixels, we prepare a *motion summary table* for each activity as presented in **Table 2** (frame #2 is the start frame and frame #90 is the end frame). With this motion summary table, Module 3 can determine if this activity is a mounting activity. Note that we focus on the height change of a mounting cow and the direction of the motion vector can be determined using a tangent graph as indicated in **Fig. 8**. For example, a motion vector can have an *upward* direction (*i.e.*, between $1/4\pi$ and $3/4\pi$ such as Up, Up-Left, and Up-Right) or a *downward* direction (*i.e.*, between $5/8\pi$ and $7/10\pi$ such as Down, Down-Left, and Down-Right). Because the start (end) frame is the first (last) frame of any motion within the RoI, the start (end) frame must have the direction upward (downward).

Table 2. Motion summary table for the mounting activity presented in **Fig. 5**

Frame number		2		6	7		90
The number of motion pixels		388	•••	1,539	1,702	•••	438
Motion vector	magnitude	9.8		34.5	32.6		7.4
	direction	up ward		up ward	up ward		down ward
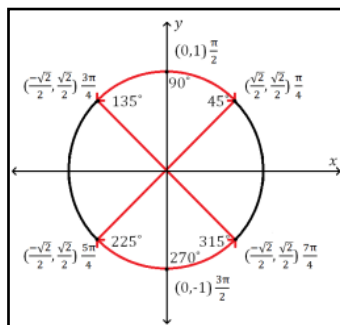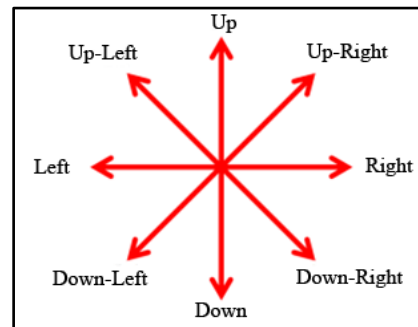
(a) Tangent graph   (b) 8-direction graph

Fig. 8. Tangent and 8-direction graph

3.2.3 Mounting verification

We verify the mounting activity with the motion summary table. The basic assumption is that a mounting activity includes both a sudden mount-up motion and a sudden mount-down motion. We first select the representative frame for the "potential" mount-up (mount-down) motion. Then, using this frame, we determine how the mount-up (mount-down) motion abruptly changes. The mount-up or mount-down motion is faster than non-mount motions.

For example, with **Table 2**, the representative frame of a potential mount-up motion (denoted as *mount-up frame*) can be selected by checking the motion summary table from left to right. We select the mount-up frame initially as the start frame (as explained, the start frame should have an upward direction). Then, we check the next frame to the right. If frame #7 has an upward direction and the number of motion pixels (*i.e.*, 1,702) is greater than that of the current mount-up frame (*i.e.*, 1,539 for frame #6), we update the mount-up frame as frame #7. This checking process continues until the frame number of the current frame being checked is greater than (the frame number of the mount-up frame $+ \Delta$). That is, if the current frame being checked differs from the mount-up frame by more than Δ, we regard the motion of the current check frame as a different motion than the mount-up motion. The threshold value Δ can be determined experimentally. Similarly, we can select the representative frame for the potential mount-down motion (denoted as *mount-down frame*) by checking the motion summary table from right to left. With **Table 2**, the mount-up (mount-down) frame is selected as frame #10 (#83).

The next step is to determine the change abruptness of the potential mount-up (mount-down) motion. Even with a side-view camera, we must resolve the *object-scale problem* (*i.e.*, a mounting cow can be captured as a small-scale or large-scale object, depending on the distance between the camera and where the mounting is occurring). We determine the maximum value of the number of motion pixels of an activity and divide the

number of motion pixels of each frame with this maximum value. With this "normalized" number of motion pixels (*i.e.*, ranged from zero to one in **Fig. 9**), we can determine the *change abruptness* of the potential mount-up motion. If (the frame number of the mount-up frame - the frame number of the start frame) is less than θ_h and the normalized number of motion pixels of the mount-up frame is greater than θ_v, we can consider that the mount-up motion changes abruptly and regard this "potential" mount-up motion as a "true" mount-up motion. The threshold values θ_h and θ_v can be determined experimentally. Similarly, we can determine the change abruptness of the potential mount-down motion by checking that (the frame number of the end frame - the frame number of the mount-down frame) $< \theta_h$ and the normalized number of motion pixels of the mount-down frame $> \theta_v$.

Finally, if the current activity has both "true" mount-up motion and "true" mount-down motion, we regard this activity as a "true" mounting activity. This "true" mounting activity can be classified further. If the mounting duration (*i.e.*, *mount_duration* = (the frame number of the end frame - the frame number of the start frame)/frame rate) is longer than 1.5 s, we regard this mounting as "mount-accept" type (*i.e.*, the mounted cow allowed the mounting activity). This is the case that we want to detect, and denote as "mounting behavior." Otherwise, we regard this mounting as "mount-reject" type (*i.e.*, the mounted cow refused the mounting activity).
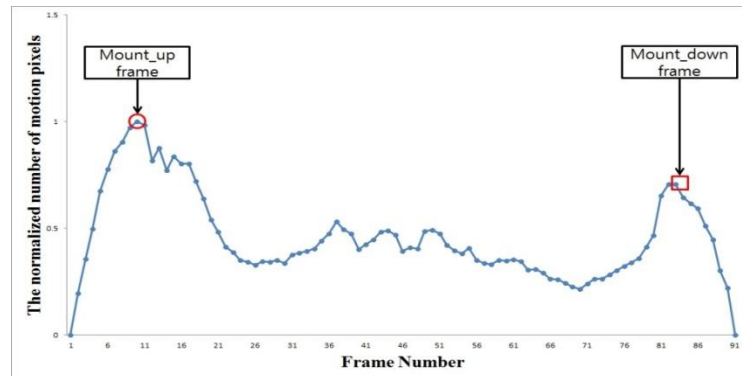

Fig. 9. Normalized number of motion pixels of the mounting activity presented in **Fig. 5**

Algorithm 1. Mounting verification

```
INPUT: motion summary table
OUTPUT: "true" or "false" mounting

true_mount = true_mount_up = true_mount_down = NO;

compute the "normalized" number of motion pixels for each frame
from motion summary table;
select mount-up frame;
select mount-down frame;
if (change abruptness of the mount-up motion=YES) {
   true_mount_up = YES;
   if (change abruptness of the mount-down motion=YES) {
      true_mount_down = YES;
      true_mount = YES;
      if (mount_duration> 1.5 s) mount_type = mount-accept;
      else mount_type = mount-reject;
   }
}
```

4. Experimental Results

The experiments were conducted using an Intel Core i5-750 2.67 GHz 4-core processor with 4GB RAM. We set the resolution to 640 × 480 pixels and the frame rate to 24 frames/s (fps). The camera was located 2 m from the fence and 1.7 m above the ground, at Milyang and Sacheon farms. The RoI was set as a center area of 640 × 80 pixels from the acquired resolution 640 × 480 pixels (See **Fig. 10**). With these settings, we acquired 25 mounting datasets from the observed field of size 5 m × 10 m. There were ten (Milyang farm) and five (Sacheon farm) cows in the field. From the experiment with 25 mounting datasets (mounting verification threshold $\triangle = 4$, $\theta_h = 16$, $\theta_v = 0.5$), all the mounting activities were detected successfully using the proposed method.
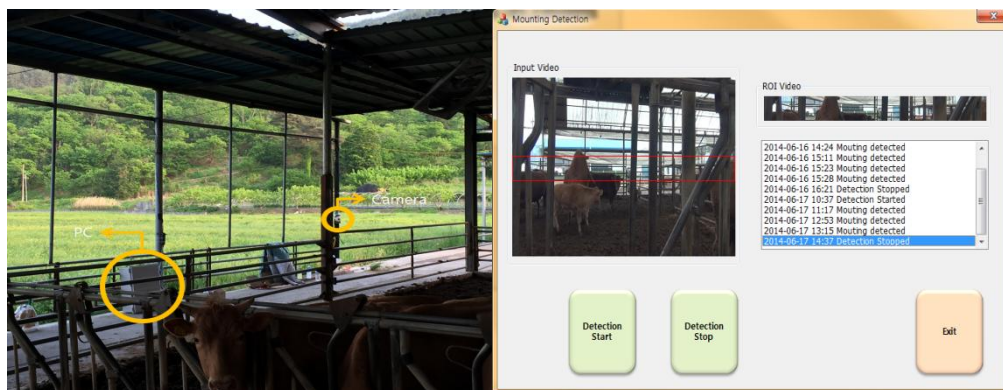

Fig. 10. Picture of a cattle farm with a video camera installed (picture taken from the Sacheon farm) and the snapshot of the implemented system

4.1 Small-Scale Mounting

As mentioned, we must resolve the object-scale problem (i.e., a mounting cow can be captured as a small-scale object or a large-scale object depending on the distance between the camera and the mounting). **Fig. 11** displays the case where a mounting cow is captured as a "small-scale" object. Because size-normalization (i.e., the normalized number of motion pixels explained in Section 3) can provide a similar motion pattern as a typical-scale mounting, the small-scale mounting can be detected successfully (see **Fig. 14** (d)).

Fig. 11. Captured scene of a small-scale mounting

4.2 Fence-Occlusion Mounting

Fig. 12 presents the case where a mounting cow is captured as an "occluded" and "large-scale" object. Because MHI can provide sufficient information even with the fence-occlusion, the portion of the mounting cow captured within the RoI can be used to detect the mounting (see **Fig. 14** (e)). This large-scale mounting must be processed in a similar fashion to the small-scale mounting, in addition to the partial-occlusion processing. With the total information for the mounting cow (*i.e.*, the circle indicated in **Fig. 12**), the information captured only within the RoI (*i.e.*, the rectangle indicated in **Fig. 12**) is sufficient to detect a large-scale mounting.

Fig. 12. Captured scene of a fence-occlusion mounting
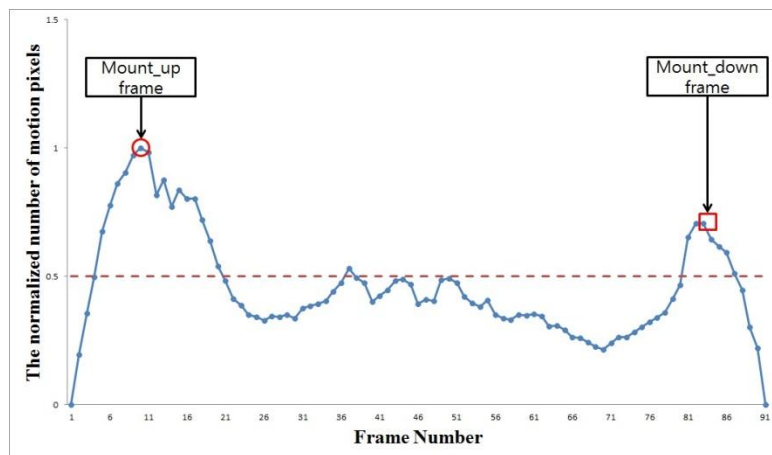
4.3 Non-Mounting (Head-Up)

Fig. 13 presents the case where a mounted cow moves her head up during a mounting. Because the head-up can also be captured within the RoI, the head-up (actually, non-mounting) motion of the mounted cow could be confused with the mount-up motion of the mounting cow. To evaluate the effect of this head-up motion, we measured the activities of the mounting cow and the mounted cow separately. Although the head-up motion might be fast (*i.e.*, the mounted cow was surprised by the mounting), the head-down motion was adequately slow to be distinguished from the actual mount-down motion (see **Fig. 14** (f)).


Fig. 13. Captured scene of a head-up motion of a mounted cow

4.4 Summary

The mounting activities captured typically had five similar motion patterns (*i.e.*, sideward, forward, backward, small-scale, and fence-occlusion). From the mounting activity for the five cases, we confirmed that the proposed method could automatically detect the activity with motion pattern analysis. The head-up activity, denoted as a non-mounting motion, could be discriminated from the mounting activities owing to the smoothness of the head-down motion compared to the abruptness of the mount-down motion. **Fig. 14** presents the normalized motion patterns of the typical mounting and non-mounting cases. In particular, to verify the effect of the head-up motion, **Fig. 14** (f) indicates only the head-up motion of the mounted cow, excluding the motion of the mounting cow. A mounting can also include a motion with right or left direction in addition to upward/downward directions. Although **Fig. 14** includes all these motions, the mount-up and mount-down frames are selected with upward/downward motions only (*i.e.*, the normalized number of motion pixels of the mount-up/down frame may be lower than that of an adjacent frame having right or left motion direction).
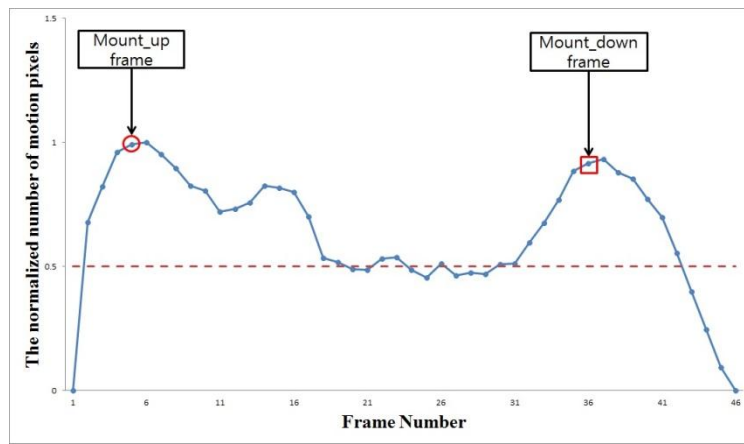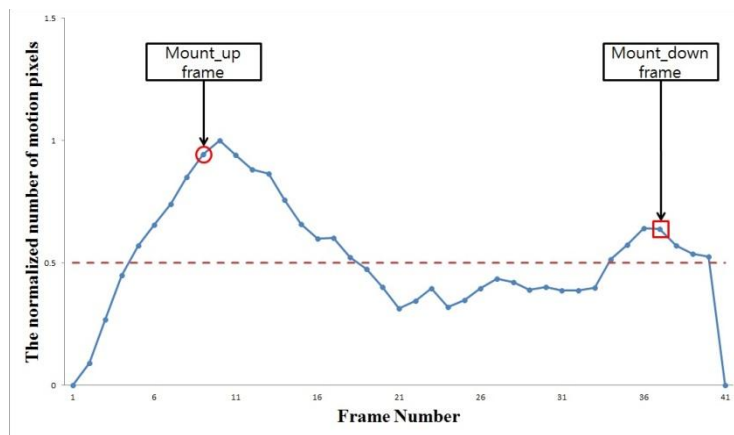

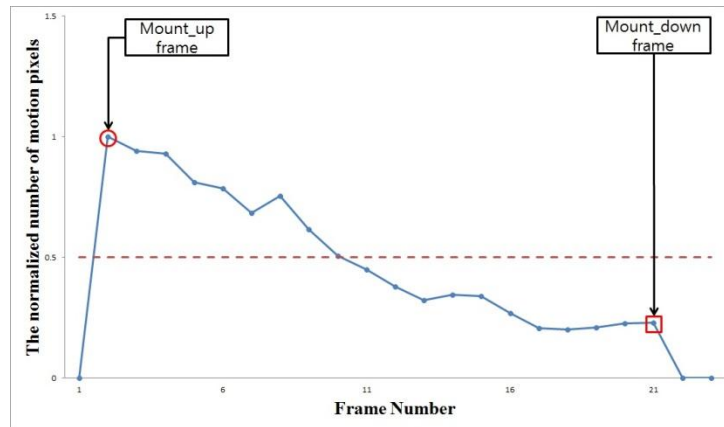(a) Case of sideward mounting
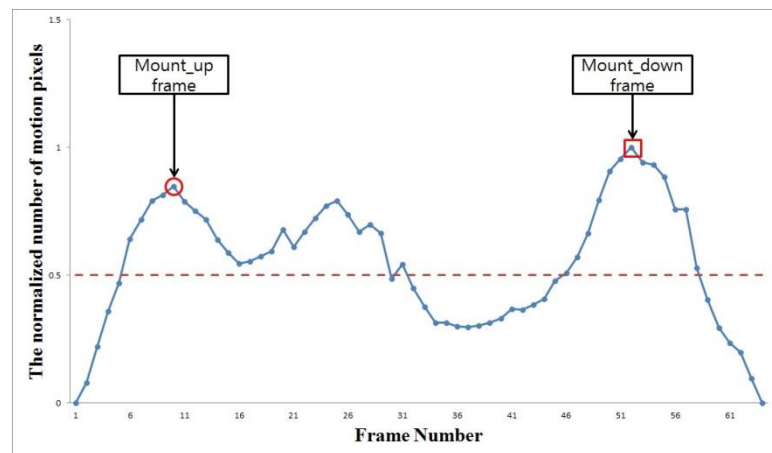
(b) Case of forward mounting
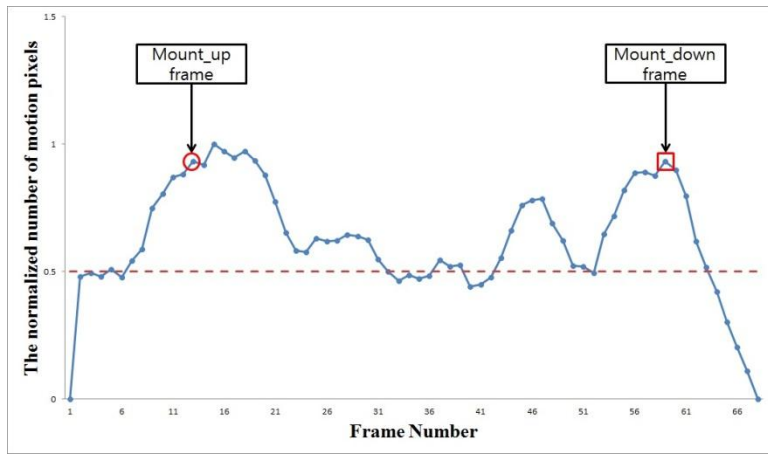
(c) Case of backward mounting

(d) Case of small-scale mounting

(e) Case of fence-occlusion mounting

(f) Case of head-up motion of the mounted cow (*i.e.*, non-mounting)

Fig. 14. Normalized number of motion pixels of typical mounting and non-mounting cases

Table 3 indicates the effect of open source GMM [16] in terms of detection accuracy. The GMM/MHI method provided higher accuracy of mounting detection than the MHI-only method because the additional GMM technique could handle the outdoor condition. Moreover, the proposed GMM/MHI method did not generate any "false alarms" with the video data captured.

Table 3. Comparison of accuracy

	Detection Accuracy
MHI-only	76%
GMM/MHI	100%

We also implemented a mounting detection system with a combination of GMM and OF using open source [16]. Although this GMM/OF method could extract some motion vectors, it could not detect a mounting in many fence-occlusion cases. The proposed GMM/MHI method may not detect the total-occlusion case (*i.e.*, a mounting is totally occluded by another mounting), although the proposed method did resolve the partial-occlusion case. With a second camera installed on the other side of the cattle shed, this total-occlusion problem could be mitigated (see **Fig. 15**).
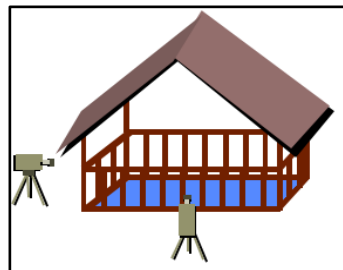

Fig. 15. Camera position for total-occlusion case

As explained in Section 3, we applied the RoI to the visual stream data to reduce the execution time of the mounting detection. **Table 4** presents the effect of RoI in terms of the average execution speed of the mounting dataset. We confirmed that the mounting detection based on RoI could be executed in real-time.

Table 4. Comparison of execution speed

	Average Execution Speed
Without RoI	6.9 fps
With RoI	29.5 fps

5. Conclusion

Automated activity monitoring in cattle management is important because it can save a significant portion of a manager's working time. In this paper, we proposed a mounting detection method for Korean native cattle from video stream data with computer vision techniques and mounting detection rules. This study focused on detecting both spatial and temporal abnormalities of a mounting with a side-view camera. From the experiments, we confirmed that the proposed method could detect a mounting by considering the direction, magnitude, and history of the mounting motion, even in the case of a fence occlusion.

References

[1] D. Berckmans, "Automatic On-line Monitoring of Animals by Precision Livestock Farming," *Animal Production in Europe: The way forward in a changing world, in between congress of the ISAH*, pp. 27-30, 2004. Article (CrossRef Link).

[2] Y. Chung, H. Kim, H. Lee, D. Park, T. Jeon, and H. Chang, "A Cost-Effective Pigsty Monitoring System based on a Video Sensor," *KSII Tr. Internet and Information Systems*, vol. 8, no. 4, pp. 1481-1498, 2014. Article (CrossRef Link).

[3] A. Yager, M. Neary, and W. Singleton, "Estrus Detection in Farm Animals," 2003. Article (CrossRef Link).

[4] G. Perry, "Detection of Standing Estrus in Cattle," 2004. Article (CrossRef Link).

[5] U. Brehme, U. Stollberg, R. Holz, and T. Schleusener, "ALT Pedometer – New Sensor-Aided Measurement System for Improvement in Oestrus Detection," *Computers and Electronics in Agriculture*, vol. 62, pp. 73-80, 2008. Article (CrossRef Link).

[6] J. Roelofs, F. Eerdenburg, N. Soede, and B. Kemp, "Pedometer Readings for Estrous Detection and as Predictor for Time of Ovulation in Dairy Cattle," *Theriogenology*, vol. 64, pp. 1690-1793, 2005. Article (CrossRef Link).

[7] J. MacKay, J. Deag, and M. Haskell, "Establishing the Extent of Behavioral Reactions in Dairy Cattle to a Leg Mounted Activity Monitor," *Appl. Anim. Behav. Sci.*, vol. 139, pp. 35-41, 2012. Article (CrossRef Link).

[8] P. Lovendahl and M. Chagunda, "On the Use of Physical Activity Monitoring for Estrus Detection in Dairy Cows," *J. of Dairy Science*, vol. 93, pp. 249-259, 2010. Article (CrossRef Link).

[9] C. Hockey, J. Norman, and M. McGowan, "Evaluation of a Neck Mounted 2-Hourly Activity Meter System for Detecting Cows About to Ovulate in Two Paddock-based Australian Dairy Herds," *Reproduction in Domestic Animals*, vol. 45, pp. 107-117, 2010. Article (CrossRef Link).

[10] Y. Chung, J. Lee, S. Oh, D. Park, H. Chang, and S. Kim, "Automatic Detection of Cow's Oestrus in Audio Surveillance System," *Asian-Aus. J. Anim. Sci.*, vol. 26, pp. 1030-1037, 2013. Article (CrossRef Link).

[11] B. Horn and B. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185-204,

1981. Article (CrossRef Link).

[12] S. Tamgade and V. Bora, "Motion Vector Estimation of Video Image by Pyramidal Implementation of Lucas Kanade Optical Flow," in *Proc. of IEEE ICETET*, pp. 914-917, 2009. Article (CrossRef Link).

[13] A. Bobick and J. Davis, "The Recognition of Human Movement using Temporal Templates," *IEEE PAMI*, vol. 23, pp. 257-267, 2001. Article (CrossRef Link)

[14] A. Ahad, "Motion History Images for Action Recognition and Understanding," *Springer Briefs in Computer Science*, 2013. Article (CrossRef Link).

[15] A. Utane and S. Nalbalwar, "Emotion Recognition through Speech Using Gaussian Mixture Model and Support Vector Machine," *International Journal of Scientific & Engineering Research*, vol. 4, no. 5, pp. 1439-1443, 2013. Article (CrossRef Link).

[16] Open Source Computer Vision, *OpenCV*, http://opencv.org.

Yongwha Chung received the BS and MS degrees from Hanyang University, Korea, in 1984 and 1986, respectively. He received the PhD degree from the University of Southern California, USA, in 1997. He worked for ETRI from 1986 to 2003 as a Team Leader. Currently, he is a professor in the Dept. of Computer and Information Science, Korea University. His research interests include video surveillance, parallel processing, and performance optimization.


Dongwhee Choi received the BS and MS degrees in computer science from Korea University, Korea, in 2013 and 2015, respectively. His current research interests include video surveillance, parallel processing, and performance optimization.

Heesu Choi received the BS degree in computer science from National Institute for Lifelong Education, Korea, in 2013. She is working on a master's degree at Korea University, Korea. Her current research interests include pattern recognition, machine learning, and data mining.

Daihee Park received the BS degree in mathematics from Korea University, Korea, in 1982, and his PhD degree in computer science from Florida State University, USA, in 1992. He joined Korea University in 1993 where he is currently a professor in the Dept. of Computer and Information Science. His research interests include data mining and intelligent database.
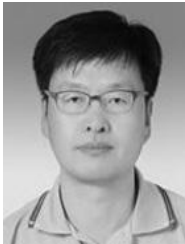
Hong-Hee Chang received the BS, MS, and PhD degrees in agricultural engineering from Chungnam National University, Korea, in 1989, 1994, and 1998, respectively. He is currently a professor in the Dept. of Animal Science, Gyeongsang National University. His research interests include livestock housing and environment control.

Suk Kim received the BS and MS degrees in veterinary medicine from Chungnam National University, Korea, in 1999 and 2002, respectively. He received his PhD degree in veterinary medicine from Gifu National University, Japan, in 2004. He is currently a professor in the College of Veterinary Medicine. His research interests include infectious diseases and preventive medicine of livestocks.