Implementation of Noise Reduction Methods for Rear-View Monitoring Wearable Devices

Huy Toan Nguyen¹, Seung You Na^{1,*}, Jin Young Kim¹
and Gwon Seok Sun²

¹*School of Electronics and Computer Engineering, Chonnam National University, Gwangju, Republic of Korea*

²*Korea Polytechnics College, Gwangju, Republic of Korea*

E-mail: nguyenhuytoantn@gmail.com; beyondi@jnu.ac.kr; mtrsk@hanmail.net

**Corresponding Author: syna@jnu.ac.kr*

Received 15 September 2016; Accepted 15 November 2016;
Publication 24 November 2016

Abstract

This study suggests effective noise reduction methods for wearable neckband devices, which are able to monitor users' rear-view areas. The wearable neckband device helps the user to monitor rear-view areas in which he/she is unable to see in normal ways (without turning back). Unlike general computers or supercomputer systems, the neckband devices have some particular constraints such as small size, lightweight and low power consumption. In a general vision system, there are many kinds of noises, which significantly decrease system quality such as impulse noise, random noise, motion noise, etc. These noises also affect wearable devices, which use cameras as the system input. Moreover, when the user walks or runs, the neckband device moves accordingly. The changing position of the neckband device causes many other noises such as camera motion (ego-motion) noises. Furthermore, when the user walks from indoors to outdoors or vice versa, the illumination dramatically changes, which also affects the device performance. Effective noise reduction methods to deal with these noises are proposed in this study. Random noise and other small noises are removed by using a Gaussian filter

Journal of Software Networking, 113–136.

doi: 10.13052/jsn2445-9739.2016.007

© 2016 River Publishers. All rights reserved.

and adaptive color threshold techniques. We propose to use feature detection and the homography matrix estimation method to reduce ego-motion noise. Remaining noises are cancelled out by a morphology technique. Finally, we apply Local Binary Patterns (LBP) descriptor and Adaboost classifier to classify whether there are people or not in the moving foreground object regions. The experiments demonstrate that our proposed noise reduction methods have achieved successful results in the different environments and users' walking speeds.

Keywords: Noise reduction, Rear-view monitoring, Wearable device.

1 Introduction

Surveillance systems are used to monitor potentially dangerous objects in region of interest [1]. A wide range of applications have been buttressed upon the advances in monitoring techniques such as search and rescue [7], video surveillance [8], vehicle driving assistance [9], reconnaissance [10] and robotics [11]. The reliability and robustness of such systems strictly depend on the environmental conditions, for instance, light intensity. There have been many studies attempting to eliminate the unwanted effects of unpredictable incoming light sources [14, 15, 24–26].

Nowadays, with the advent of semiconductor and electronic technologies, electronic components are becoming more and more affordable. Thanks to the development of digital cameras, video monitoring systems based on computer vision are becoming popular in daily life. Due to that reason, various approaches to building autonomous systems based on image processing have been published [12, 13]. Their works were based on computers or supercomputers to achieve high accuracy and reduce processing time. We propose the design and implementation of a small wearable device to detect and classify dynamic objects in a rear-view area of the user based on microprocessors.

Developing a small wearable device with high accuracy and short processing time is a challenge due to low resolution, background changes, and hardware constraints. In case of wearable devices, there are strong variations in illumination, background, shadows and other random types of noise such as impulse noise [14], Gaussian noise, Poisson noise [15], Speckle noise, Salt and Pepper noise, which can make the situation even more complicated. To address this problem, various effective noise reduction methods for small wearable devices are studied and exploited in this paper.

2 Related Work

2.1 Video Monitoring System

There has been increasing interest in building monitoring systems. However, a video monitoring system has many constraints such as low processing time, high accuracy, affordability and ease for the user. To address these problems, a lot of research work has been proposed. O. Jafari *et al.* [2] and D. Mitzel *et al.* [3] suggested a system, which used a single CPU core. They used a head-mounted camera based on Kinect RGB-D input data to detect pedestrians in a close-range and achieved a high processing speed up to 18 fps. An application using IP cameras to send and receive data via network and Internet was proposed by R. Rashmi *et al.* [4]. They used Motion Detector application to warn the user via email or text messenger. The image data were transferred from mobile to PC to store data for a long time. Lefang *et al.* [5] came up with a remote video monitoring system based on ARM processing chip using Linux operating system combined with GPRS technologies. X. Jiangsheng [6] introduced a video monitoring system based on TMS320DM642 DSP, local network and GPRS for large railway maintenance machinery. However, these previously mentioned systems were mainly based on computers or supercomputers for processing acquisition videos. The high cost and inconvenience for a single user limit the popularity of these systems in real life. The design and implementation of a wearable device, which is small and lightweight as SenseCam device [16] with real-time processing, are currently hot topics in the computer vision and robotics communities.

2.2 Wearable Device

The design and development of wearable devices have accumulated lots of attention from the scientific community and industry for the last few years [17, 18]. There are various applications based on wearable devices, especially for health care problems. In [19], A. Pantelopoulos *et al.* designed and developed wearable biosensor systems for health monitoring. The wearable biosensor system included many parts such as physiological sensors, transmission modules and processing capabilities for health monitoring. An overview of state-of-art wearable technologies for remote patient monitoring was presented by Hung *et al.* [20]. Their research work suggested developing tele-home healthcare systems, which used wearable devices for monitoring remotely. J. Hernandez *et al.* [21] proposed to use sensors embedded in

the Google Glass. In [22], C. Setz *et al.* introduced the wearable device for detecting early warning signs at the workplace. They took interest in developing a method to analyze the discriminative power of electrodermal activity (EDA) to detect stress signals. On the other side, wearable devices were adopted to make navigation systems to assist impaired people. In [23], a survey about wearable devices, which were used for obstacles avoidance purposes, was presented. The author introduced many wearable devices for assisting impaired people. Three main categories of wearable navigation systems were electronic travel aids (ETAs), electronic orientation aids (EOAs), and position locator devices (PLDs). Most of the approaches introduced so far consider wearable devices for healthcare and obstacles avoidance purposes with active sensors such as distance sensors or other sensors for gathering input signals. Due to the limitation of active sensors, wearable monitoring systems based vision sensors have been increasing. However, only few approaches provided experimental results for monitoring at human blind spots based on vision sensors. In this paper, we develop noise reduction methods, which can be applied to the small wearable device based on microprocessor using camera as an input sensor.

2.3 Noise Reduction Method

The output quality of a wearable neckband device with a camera can be significantly decreased by various components of noise. There are some sources of noise such as camera sensor, random noise, and motion noise. Recently, there has been growing interest in noise reduction and enhancement of the output quality. In [24], M. Kim *et al.* proposed an approach to reduce noise and enhance quality of extremely low-light video. They used adaptive temporal filtering based on Kalman filter to reduce noise. Another method for denoising color video sequences was introduced by V. I. Ponomaryov *et al.* in [25]. They selected fuzzy filtering to obtain better results. The image denoising method based on local average gray values and gradients, pre-classifying neighbourhoods and thereby reducing the original quadratic complexity to a linear one and the influence of less-related areas in the denoising of a given pixel was presented by M. Mahmoudi *et al.* in [26].

Different from the aforementioned methods, we propose a noise reduction method to eliminate both static and motion noise for rear-view monitoring wearable devices. We rely on Gaussian Filter and morphology technique to overcome static noise. The ego-motion noise caused by the camera variation is compensated by feature based and homograph decomposition method.

We propose adaptive color threshold method and apply Adaboost classification using LBP features to achieve useful information from input captured video.

We organize our paper as follows. In Section 3, the system architecture including system hardware and system software is first explained. The experiment setup is fruitfully represented in Section 4. Results and discussions are in Section 5. Finally, conclusion and future works are mentioned in Section 6.

3 System Architecture

Unlike a general computer or supercomputer, which can execute multiple tasks, the proposed wearable neckband device system only performs monitoring tasks with the specific hardware constraints. The system hardware includes a set of constraints such as small sizes, light weight, low costs, and high processing ability. The proposed algorithms for noise reduction have been designed and implemented by the system software. In the next parts, the system hardware and software are discussed in detail.

3.1 The System Hardware

In general, the system performance depends on the system hardware. The proposed system hardware is divided into several parts. The schematic diagram is presented in Figure 1.

3.1.1 Camera module

The camera module is one of the most important and vital elements in any vision system. The camera's quality can directly affect the system performance. Due to the system constraints, a small, inexpensive, and acceptable

Figure 1 Hardware system.

quality camera is selected. Captured images are transmitted to the processing board via a lightweight cable.

3.1.2 Processing board

A small, lightweight and user-friendly Processing Board (PB) is selected. The CPU on PB takes the video data input from the camera module, processes all logical and computational tasks and gives the output video result on the LCD monitor. In our case, CPU and memory are integrated on the same PB. In the selection of PB, low power consumption for the long time operation is considered as one of vital properties.

3.1.3 Memory

Another basic element of system hardware is the memory. The memory sizes can significantly affect the system performance and time consumption. In the general case, the memory is separated into two parts: Random Access Memory (RAM) and storage memory. All the temporary data and variables are stored in the RAM, while the system is turned on. Another memory is used for storing the operating system and software algorithms. All memories are physically integrated into PB.

3.1.4 Power supply

When designing small devices such as wearable devices, low power consumption is highly considered. For the long time operation, the proposed system components are selected to reduce power consumption. The system software is also optimized for the same purpose. For convenient and long time operation, a small size, high output capacity and suitable shape battery is selected.

3.1.5 LCD screen

After processing steps, the results are displayed on the LCD screen. To make the system more user-friendly and interactive, the touch screen LCD is installed. The display screen not only provides certain information to the user but also works as the interface between the user and the system device. All interaction tasks are performed on the touch LCD screen. The LCD screen, which is almost the same size as the PB, is chosen to decrease the device's physical size.

3.2 The System Software

The system software describes all performing logical and mathematical calculations on the input video data. The output of the system is presented on

the LCD screen. With the specific system hardware, the good system software tends to achieve real-time and high accuracy with low power consumption. In this section, an algorithm for monitoring a moving object in a rear-view range is proposed.

The input image data is captured from the mounted camera on the wearable neckband device when the user walks. The distance from the user to objects is in a close-range of 1 to 10 meters. The proposed method has three main steps: pre-processing, detection of moving foreground objects and object classification. The methods of reducing noise in a captured image are proposed in the pre-processing step. Noises, which mainly come from the illumination changes and the camera sensor, are reduced in this step. The primary purpose of the second step is to detect moving foreground objects. Moreover, in the second step, methods of decreasing camera motion noises and the computation time are considered. Finally, object feature extraction and object classification within the moving object regions are introduced in the third section. The flowchart of the system software is presented in Figure 2.

3.2.1 Pre-processing step

The input captured image is distorted by many kinds of noise such as impulse noise, random noise, Gaussian noise and so on. The quality of the system is directly affected by these noises. Due to hardware limitations and real-time constraints, a simple filter is selected to remove these noises. At the beginning, when the camera module is activated, the input captured image $I(x, y, t)$ immediately is resized to 320×240 pixels and converted to grayscale $I^{gray}(x, y, t)$. The Gaussian Filter is adopted for noise reduction on the converted grayscale image. The Gaussian Filter can be presented by Equation (1):

$$G_0(x, y) = Ie^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}} \quad (1)$$

where μ is the mean and σ represents the variance (for each of the variables x and y).

We selected the kernel size equal to 5×5 . After applying Gaussian filter, small noises such as salt and pepper noise and random noise are cancelled out.

Figure 2 Software system.

3.2.2 Moving foreground objects detection

The camera is mounted on the neckband for detection task. The main reason of background instability is the moving action of the user when he/she uses the wearable system. The camera ego-motion and the motion of moving objects are considered as two fundamental motions in the detection problem. In case of non-static background, the camera motion has to be estimated to know which parts of the picture are changed due to the camera motion and the other parts of the picture are changed independently. In [27], D. Szolgay *et al.* adopted Hierarchical Block-Matching algorithm to estimate the strong camera motion. The main problem with Hierarchical Block-Matching algorithm is the computation cost. In this paper, we propose a method to overcome the weakness of the above method.

Let's consider two grayscale consecutive images $I^{gray}(x, y, t - 1)$ and $I^{gray}(x, y, t)$. In the ideal case, the relationship between two grayscale consecutive images can be presented by Equation (2) or Equation (3):

$$I^{gray}(x, y, t) = HI^{gray}(x, y, t - 1) \quad (2)$$

or

$$\begin{pmatrix} x_t \\ y_t \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ 1 \end{pmatrix} \quad (3)$$

where H is a homography matrix.

It is necessary to extract moving foreground objects in order to reduce calculation time for object classification step. Through experiments with different features, Good Feature to Track [28] is suitable for our system in comparison with other methods [29, 30, 31, and 32]. In order to track key points from the previous grayscale image $I^{gray}(x, y, t - 1)$ to the current grayscale image $I^{gray}(x, y, t)$, we propose to use the tracking method [33]. The projective transformation matrix between the two consecutive images, a homography matrix H , is determined by the RANSAC method [34]. From the homography matrix, we figure out the reference motion-compensated image by Equation (4).

$$I_{\text{Ref}}^{gray}(x, y, t) = HI^{gray}(x, y, t - 1) \quad (4)$$

Error image $E^{gray}(x, y, t)$, which is determined by subtraction of $I^{gray}(x, y, t)$ from $I_{\text{Ref}}^{gray}(x, y, t - 1)$, presents the differences between an ideal image result and a real image result. The error image $E^{gray}(x, y, t)$

presents moving foreground objects independently from the camera motion as follows:

$$E^{gray}(x, y, t) = |I_{Ref}^{gray}(x, y, t - 1) - I^{gray}(x, y, t)| \quad (5)$$

In an ideal situation, the error image $E^{gray}(x, y, t)$ consists of only moving foreground objects. However, in our real experiments, the camera motion, in some cases, is even stronger than movements of an object itself. As a result, the error image also contains false positive pixels, which significantly degrade the system quality. Using normal methods, which are based on gray scale images to get moving foreground objects, is impossible in this case. Thus, we propose to use color information from the original image $I(x, y, t)$. The proposed error image is presented by the following equation:

$$E(x, y, t) = \begin{cases} 0 & \text{if } E^{gray}(x, y, t) = I^{gray}(x, y, t) \text{ or } E^{gray}(x, y, t) < th_E \\ I(x, y, t) & \text{otherwise} \end{cases} \quad (6)$$

Finally, the threshold error image th_E is calculated by the mean and variance of error image in each R, G, and B channel separately. This threshold is adaptive for each specific error image. Finally, we apply a morphology method to remove small noise and get moving foreground object regions.

3.2.3 Object classification

After the pre-processing step, the random noise, illumination noise and other motions noises should have been removed already in the video dataset. The output of Section 2 is regions of interest (ROI), which include only moving foreground objects. In order to get more detail information, we propose the object classification step to determine whether moving objects are human or non-human.

In our experiments, moving objects are in a rear-view range in the distance of 1 to 10 meters. We realize that when the user wears the neckband device, in a rear-view and close range, the camera is unable to capture the full-body images. Due to this reason, the full-object detector methods [35, 36] are not suitable for our cases. In this study, for the object classification purpose, we only concentrate on human face detection in a rear-view area and close range distance.

Feature selection and classification have demonstrated promising results for this task. Among many feature descriptors for the object classification, we propose to use a Local Binary Pattern descriptor and an AdaBoost classifier. The original LBP [37] forms labels for the image pixels by using the threshold

Figure 3 The basic LBP operator [30].

of the 3×3 neighborhood of each pixel with the center value and then by considering the result as a binary number.

The basic methodology for LBP based on face description was proposed by Ahonen *et al.* [38]. The facial image is divided into local regions, and LBP texture descriptors are extracted from each region independently. The basic LBP is shown in Figure 3.

Due to the necessity of the fast response, we select an Adaboost algorithm [39] for classification tasks.

4 Experiments

4.1 Hardware Selection

Component selection is one of the most challenging tasks for designers or researchers, in which they have to have a balance between two conflicting requirements, which are low cost and high performance.

In our experiment, due to some constraints such as small in size, high processing ability, low-power consumption, and lightweight, Raspberry Pi 2 model B (RPI) is selected as the main processing board. All system software steps are implemented on this processing board. Raspberry Pi 2 model B is a credit card sized single-board computer. With a small size, this processing board is suitable for the single user to use our product.

To get the high quality input video data, Raspberry Pi camera module is selected as the system input. Raspberry Pi camera module has a 5-Megapixel camera, which can record high-definition (HD) videos up to 720p resolution, with high sensitivity, low noise and low crosstalk features. The maximum capture rate is up to 60 frames per second (fps). Moreover, Raspberry Pi camera module has extremely small size ($25 \times 20 \times 9$ mm) and lightweight (3 grams), which is sufficient to make wearable devices as neckband devices.

Furthermore, the RPI has 1 GB RAM, which is enough for real-time processing. The 16 GB SDCard is selected as the external storage of the system.

The operation system and all software are stored in the external storage. The reading and writing speed of SD card also affect the system performance.

Results are presented on the LCD screen. In our experiment, we choose the LCD touch screen. The LCD screen is the same size as the PB. The touch LCD screen not only provides useful information, but can also work as the interface for the device.

Finally, the battery is selected for our system. We choose a small size, high capacity, and suitable shape battery for a wearable device. The proposed neckband device hardware is shown in Figure 4.

4.2 Datasets

To evaluate proposed noise reduction methods, datasets are recorded in real conditions where the illumination changes and the user keeps on walking while the system records video. The datasets are recorded in different environments including various indoor and outdoor environments. The user's walking speed is considered as one of the key elements, which affects the final results. The user's walking speed is classified into three categories: slow, normal, and fast speed. The details about datasets are provided in Table 1.

Figure 4 The neckband device. (a) Hardware structure, (b) Setup device for a user.

Table 1 Test dataset description

Video Sequence	Environments	Walking Speed	Frames
Video 1	Indoor	Fast	166
Video 2	Indoor	Normal	262
Video 3	Indoor	Slow	283
Video 4	Outdoor	Slow	286

5 Results and Discussions

The software algorithms are implemented by using the OpenCV library based on Python programming language. To evaluate the running time of our algorithm on proposed platform, we compute the average running time on each step by the following Equation:

$$\bar{t} = \frac{\sum_{i=1}^N t_i}{N} \quad (7)$$

where N is the number of frames in one video sequence, t_i is processing time for frame i , $\bar{t}$ is the average time consumption for whole video. It is computed separately for each step for each video. Finally, we compute the average consumption time for four videos by Equation (7) with N is the video number, and t_i is the processing time for video i . The average processing time for each step is presented in Table 2.

Based on Table 2, the average total running time for one frame, which has frame size of 320×240 pixels, is approximately equal to 771 milliseconds. The Pre-processing step for reducing small static noise by Gaussian Filter takes around 11 milliseconds to complete. Moving foreground object detection takes the longest time due to a number of sub-steps inside such as feature detection, feature tracking, homography matrix estimation and adaptive color threshold technique. The human face detection in region of interest (ROI) based on LBP descriptor and Adaboost classify method takes 200 milliseconds. This method is mainly applied when users walk or run, so it is possible to be real-time method for these applications.

We extend our previous work [1] by comparing feature extraction processing times on one frame with different techniques. The approximate running time per frame for each method is computed by Equation (7). Moreover, the average number of detected key points in each frame is estimated in Equation (8):

Table 2 Average processing time for each step per frame

Step	Time Consumption (second)
Pre-processing	0.0109
Moving foreground object detection	0.5634
Face classification	0.1969
Total Time	0.7712

$$\bar{k} = \frac{\sum_{i=1}^N k_i}{N} \quad (8)$$

where N is the number of frame in one video sequence, k is number of detected key points in frame i^{th} , $\bar{k}$ is the average feature detected in video sequences. The detailed results and samples are presented in Table 3 and Figure 5.

Table 3 Average runtime feature extractions per frame

Features	Number of Key Points	Time Consumption (seconds)
Good Feature to Track [28]	52	0.0147
FAST [29, 30]	479	0.0088
ORB [31]	446	0.0315
BRIEF [32]	55	0.0128

Figure 5 Comparison of features detection methods. (a) Good Features to Track, (b) FAST features, (c) ORB features, (d) BRIEF features.

We realize that various other feature detection methods such as FAST [29, 30] and BRIEF [32] consume less time to detect feature points in compare with Good Feature to Track [28]. However, after several experiments, we find out that other feature detection methods such as FAST, ORB and BRIEF produce higher wrong detection points for the images with moving objects. Since homography matrix is calculated based on detected feature points, wrong detected points lead to fail in calculation of homography matrix. In order to achieve high accuracy and acceptable processing time, we choose Good Feature to Track method for detecting key points in our experiments.

Recently, in [40, 41], authors proposed the process execution of object classification based on Haar-features with the same platform. We adopt LBP descriptors and Adaboost classification, which is distinct from previous published studies to achieve better running time. The results are shown in Table 4.

Average processing time per frame is calculated and compared with other image denoising and object detection methods as presented in Table 5.

The accuracy of our proposed methods for noise cancelling due to the user's walking movements is presented in Table 6. In this part, we test our algorithms on several video sequences with different walking speeds and various environments.

Table 6 show that Video 3, which is recorded in indoor environments with the slow walking speed, give the best accuracy result. When the user walks faster, the accuracy decreases due to calculation error of homography matrix.

Table 4 Comparison of face classification performance

Features	Time Consumption (second)
Haar-features [40, 41]	0.556
LBP descriptors	0.1969

Table 5 Average processing time per frame

Method(s)	Application	Processing Time (seconds)
M. Kim <i>et al.</i> , 2015 [24]	Image denoising	6.8
M. Mahmoudi <i>et al.</i> , 2005 [26]	Image denoising	481
D. Szolgay <i>et al.</i> , 2011 [27]	Moving object detection	39.4138
R. J. Moreno, 2014 [40]	People detection	2.5
W. F. Abaya <i>et al.</i> , 2014 [41]	Security camera at night	2.5
Our method	Noise Reduction & Moving object detection	0.7712

Table 6 The system performance

Video Sequence	Environments	Walking Speed	Accuracy
Video 1	Indoor	Fast	69.71%
Video 2	Indoor	Normal	73.04%
Video 3	Indoor	Slow	88.70%
Video 4	Outdoor	Slow	81.48%

With the same speed but different environments, the accuracy is different, due to the vibrational background. In indoor environments, we get more stable key points from the non-moving objects such as walls and doors. However, in the outdoor case, the detected key points may not only on a static object but also from some small moving objects such as trees and flags moved by wind.

For better understanding of the proposed method, we illustrate a sample on two consecutive images, called previous image and current image, in our video sequences as shown in Figure 6. First, two color input images are converted to grayscale images as shown in Figure 6(a) and 6(d). Then, Gaussian Filter is applied to remove static noises such as random noise and other small noises. Results of static noise cancellation method are presented in Figure 6(b) and 6(e). The feature points are detected from the previous image as in Figure 6(c). These feature points are tracked from the previous image to current image, respectively. In case of static camera, we can apply simple background subtraction technique to achieve foreground objects. However, it is impossible to achieve only foreground object regions if we apply the same process for an unstable camera as our cases, as shown in Figure 6(g). For that reason, the homography matrix H is calculated based on detected feature points on the previous image and tracked feature points in the current image. After applying Equation (4) and Equation (5) respectively, we get the error image as illustrated in Figure 6(h). Due to the lack of information about foreground objects, we continuously correct the error image by Equation (6), and the results are shown in Figure 6(i). The foreground objects are determined from this modified error image, which is estimated by white region in Figure 6(j). Figure 6(k) illustrates the region of interest (foreground moving objects), which is shown in the green rectangular box. Finally, face classification based on LBP descriptor and Adaboost classifier in the region of interest is shown in Figure 6(l). After a number of steps, the user can know whatever there are moving objects or not in blind areas regardless of human or non-human objects. The results show that our algorithm is able to work well enough to reduce various kinds of noises and correctly classify human or non-human objects.

Figure 6 The sample result. (a) (d) Two grayscale consecutive images, (b) (e) Two consecutive images after removing static noises, (c) Detected features in the previous image, (f) KLT tracker features, (g) Normal subtraction, (h) Error image, (i) Modified error image, (j) Region of Interest, (k) Moving foreground objects, (l) Face detection.

6 Conclusions and Future Works

A design and implementation of a wearable neckband device for rear-view monitoring is presented in this study. A monitoring system has to face various types of noise. Some of them are from the hardware electronic devices while

others are from the outside environment. Effective noise reduction methods are proposed in this paper to deal with different kinds of noises. We propose combining a Gaussian filter, an adaptive color filter and a homography matrix to reduce the external noise effects. The Gaussian filter is adopted to reduce common noises such as Salt and Pepper noise, which is caused by camera quality. We choose Good Feature to Track method to extract key points and the homography matrix to estimate moving foreground objects. We propose an adaptive color threshold method to get moving foreground objects. The remaining noise is removed by morphology technique. Finally, we apply the LBP descriptor and Adaboost to classify as human or non-human in the video frame.

The proposed noise reduction process for wearable neckband device is able to execute high accuracy when the user walks with a normal speed. The static and motion noises are removed by combining homography decomposition technique and adaptive color method. The proposed method is investigated using a small microprocessor device. It is possible to increase the accuracy of the system in the future. In the next stage, we will extend this research to other platforms in case of camera moving with a higher speed. The above results of noise reduction method can be applied to other real devices for monitoring blind spots around cars, search and rescue based drones or even for mobile navigation.

Acknowledgments

This study was financially supported by Chonnam National University, 2016.

References

- [1] Nguyen, H. T., Choi, Y. S., Sun, G. S., Na, S. Y., and Kim, J. Y. (2016). Effective noise reduction methods for rear-view monitoring devices based on microprocessors. *Mobile Wirel. Technol.* 391, 51–58.
- [2] Jafari, O., Mitzel, D., and Leibe, B. (2014). “Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras,” in *ICRA International Conference on Robotics and Automation*. RWTH Aachen University, Germany.
- [3] Mitzel, D., and Leibe, B. (2012). “Close-range human detection for head-mounted camera,” in *BMCV British Machine Vision Conference*. RWTH Aachen University, Germany.
- [4] Rashmi, R., and Latha, B. (2013). “Video surveillance system and facility to access Pc from remote areas using smart phone,” in *ICICES*

- International Conference on Information Communication and Embedded System* (Rome: IEEE), 491–495. doi: 10.1109/ICICES.2013.6508393
- [5] Lefang, Z., Jian-xin, W., and Kai, Z. (2013). “Design of embedded video monitoring system based on S3C2440,” in *ICDMA International Conference on Digital Manufacturing and Automation* (Rome: IEEE), 461–465. doi: 10.1109/ICDMA.2013.108
- [6] Jiangsheng, X. (2011). “Video monitoring system for large maintenance machinery,” in *ICEMI International Conference on Electronic Measurement and Instrument* (Rome: IEEE) 3, 60–63. doi: 10.1109/ICEMI.2011.6037855
- [7] Morse, B. S., and Engh, C. H. (2010). “UAV video coverage quality maps and prioritized indexing for wilderness search and rescue,” in *Proceedings of the 5th ACM International Conference on Human-Robot Interaction* (Rome: IEEE), 227–234.
- [8] Tseng, B. L., Lin, C. Y., and Smith, J. R. (2002). “Real-time video surveillance for traffic monitoring using virtual line analysis,” in *International Conference on Multimedia and Expo* (Rome: IEEE), 2, 541–544.
- [9] McCall, J. C., and Trivedi, M. M. (2006). Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Trans. Intell. Transport. Syst.* 7, 20–37.
- [10] Bhaskaranand, M., and Gibson, J. D. (2013). Low Complexity Video Encoding and High Complexity Decoding for UAV Reconnaissance and Surveillance. *Int. Symp. on Multimedia*, 163–170.
- [11] Li, Z., Yang, C., C-Y Su, Deng, J., and Zhang, W. (2016). Vision-Based Model Predictive Control for Steering of a Nonholonomic Mobile Robot. *IEEE Trans. Control Syst. Technol.* 24, 553–564.
- [12] Arroyo, R., Yebes, J. J., Bergasa, L. M., Daza, I. G., and Almazán, J. (2015). Expert video-surveillance for real-time detection of suspicious behaviours in shopping malls. *Int. J. Expert Syst. Appl.* 42, 7991–8005.
- [13] Guler, P., Emekksiz, D., Temizel, A., Mustafa Teke, and Temizel, T. T. (2016). Real-time multi-camera video analytics system on GPU. *J. of Real-time Image Processing* 11, 457–472.
- [14] Yadav, P. (2015). “Color image noise removal by modified adaptive threshold median filter for RVIN.,” in *EDCAV International Conference on Electronic Design, Computer Networks and Automated Verification* (Rome: IEEE), 175–180.
- [15] Foi, A., Trimeche, M., Katkovnik, V., and Egiazanrian, K. (2008). Practical Poissonian–Gaussian Noise Modeling and Fitting for Single-Image Raw-Data. *IEEE Trans. Image Process.* 17, 1737–1754.

- [16] Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., G. Smyth, Kapur, N., and Wood, K. (2006). "SenseCam: a retrospective memory aid," in *International Conference of Ubiquitous Computing* (Berlin: Springer), 177–193.
- [17] Lv, Z., Feng, S., Feng, L., and Li, H. (2015). Extending touch-less interaction on vision based wearable device. In *Virtual Reality (VR)*, 231–232.
- [18] Woodberry, E., Browne, G., Hodges, S., Watson, P., Kapur, N., and Woodberry, K. (2015). The use of a wearable camera improves autobiographical memory in patients with Alzheimer's disease. *J. Memory*, 23, 340–349.
- [19] Pantelopoulos, A., and Bourbakis, N. G. (2010). A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis. *IEEE Trans. Syst. Man Cybern C Appl. Rev.* 40, 1–12.
- [20] Hung, K., Yang, Y. T., and Tai, B. (2004). Wearable medical devices for tele-home healthcare. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 7, 5384–5387.
- [21] Hernandez, J., Li, Y., Rehg, J. M., and Picard, R. W. (2014). "BioGlass: Physiological parameter estimation using a head-mounted wearable device," in *Proceedings of the 2014 EAI 4th International Conference on Wireless Mobile Communication and Healthcare (MOBIHEALTH 2014)*, 55–58, Athens.
- [22] Setz, C., Arnrich, B., Schumm, J., Marca, R. L., Tröster, G., and Ehlert, U. (2010). Discriminating stress from cognitive load using a wearable EDA device. *IEEE Trans. Inf. Technol. Biomed.* 14, 410–417.
- [23] Dakopoulos, D., and Bourbakis, N. G. (2010). Wearable obstacle avoidance electronic travel aids for blind: a survey. *IEEE Trans. Syst. Man Cybernetics C* 40, 25–35.
- [24] Kim, M., Park, D., Han, D. K., and Ko, H. (2015). A novel approach for denoising and enhancement of extremely low-light video. *IEEE Trans. Consum. Electron.* 61, 72–80.
- [25] Ponomaryov, V. I., Montenegro-Monroy, H., Gallegos-Funes, F., Pogrebnyak, O., and Sadovnychiy, S. (2015). Fuzzy color video filtering technique for sequences corrupted by additive Gaussian noise. *J. Neurocomputing* 155, 225–246.
- [26] Mahmoudi, M., and Sapiro, G. (2005). Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Process. Lett.* 12, 839–842.

- [27] Szolgay, D., Benois-Pineau, J., Megret, R., Gaestel, Y., and Dartigues, J. F. (2011). Detection of moving foreground objects in videos with strong camera motion. *J. Pattern Anal. Appl.* 14, 311–328.
- [28] Shi, J., and Tomasi, C. (1994). “Good features to track,” in *Proceedings of the CVPR International Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 593–600.
- [29] Rosten, E., and Drummond, T. (2006). “Machine learning for high speed corner detection,” in *Proceedings of the ECCV the 9th European Conference on Computer Vision*, Graz, 430–443.
- [30] Rosten, E., Porter, R., and Drummond, T. (2010). Faster and better: a machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 105–119.
- [31] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. R. (2011). “ORB: an efficient alternative to SIFT or SURF,” in *Proceedings of the ICCV International Conference on Computer Vision* (Rome: IEEE), 2564–2571. doi: 10.1109/ICCV.2011.6126544.
- [32] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). “BRIEF: binary robust independent elementary features,” in *Proceedings of the ECCV the 11th European Conference on Computer Vision*, Heraklion, 778–792.
- [33] Tomasi, C., and Kanade, T. (1991). *Detection and Tracking of Point Features*. Technical Report CMU-CS-91-132. Pittsburgh, PA: Carnegie Mellon University.
- [34] Fischler, M., and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting applications to image analysis and automated cartography. *Commun. ACM* 24, 381–395.
- [35] Dalal, N., and Triggs, B. (2005). “Histograms of oriented gradients for human detection,” in *Proceedings of the CVPR Conf. on Computer Vision and Pattern Recognition* (Washington, DC: IEEE Computer Society), 886–893.
- [36] Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 1627–1645.
- [37] Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recogn.* 29, 51–59.
- [38] Ahonen, T., Hadid, A., and Pietikinen, M. (2006). Face description with local binary patterns: application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, 2037–2041.

- [39] Freund, Y., and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. *Comput. Learn. Theory* 55, 23–37.
- [40] Moreno, R. J. (2014). “Robotic explorer to search people through face detection,” in *Proceedings of the CIIMA International Congress of Engineering Mechatronics and Automation* (Rome: IEEE), 1–4.
- [41] Abaya, W. F., Basa, J., Sy, and Abad, A. C. (2014). “Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV,” in *Proceedings of the HNICEM International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control Environment and Management* (Rome: IEEE), 1–6.

Biographies

H. T. Nguyen is a Ph.D. candidate in School of Electronics and Computer Engineering, Chonnam National University, Gwangju, Republic of Korea. He received his Bachelor's degree in Electronic & Communication Engineering from Thai Nguyen University of Technology, Vietnam in 2012. His research interests include Computer Vision, Wearable Devices, Robot Vision and Embedded Systems.

S. Y. Na received B.S. degree from Seoul National University in 1977, M.S and Ph.D degrees in 1984 and 1986, respectively, from the University of Iowa, USA. Since 1987, he has been a professor at the department of Electronics and Computer Engineering, Chonnam National University. His current research topics are controller design, soft computation methods, sensor-based control, robotics and pattern recognition.

J. Y. Kim received B.S. degree, M.S degree and Ph.D degree in 1986, 1988 and 1994, respectively, from Seoul National University. From 1993 to 1994 he was engaged as a research engineer at Korea Telecom. Since 1995, he has been a professor at Chonnam National University. His current research topics are audio-visual speech process, image processing and cognitive radio.

G. S. Sun received Ph.D degree in 2015 from Chonnam National University. Since 1996, he has been a professor of Mechatronics Engineering at Korea Polytechnics College, after working at various departments of Kia Motors Company. He published nine books in the field of control applications. Now he is deputy director of Smart Factory work promotion center in Gwangju Metropolitan Office. His current research topics are smart factory, system integration for IoT, and sensor-based factory automation.