Journal of Engineering and Applied Sciences 13 (Special Issue 2): 2963-2970, 2018
ISSN: 1816-949X

Study for an Individual Recording System Using Video-Based Tracking Target in Mobile Phone

[1]Juwon Jin and [2]Byeongtae Ahn
[1]Computer Science and Engineering, 31253 Koreatech, South Korea
[2]Liveral and Arts College, Anyang University, 14028 Anyang, South Korea

Abstract: As modern society be increasingly personalized, the trend of media market. We propose a system consisting of a commercially available mobile camera (on an 'Android' mobile OS), a servomotor for rotating the smartphone camera to the right or left, a micro-controller ('Arduino') for controlling the motor and a wireless Bluetooth ear set for audio input. We propose a system consisting of a commercially available mobile camera (on an 'Android' mobile OS), a servomotor for rotating the smartphone camera to the right or left, a micro-controller ('Arduino') for controlling the motor and a wireless Bluetooth ear set for audio input. This study proposes an unmanned recording system in mobile phone using image processing technologies in order to detect and track an object without a person that monitors the object and controls the camera. The automatic tracking object system is designed by the process of detecting and tracking the face of an object. As for automatic tracking object, 'Arduino', a kind of micro-controller is used. The system this study proposes has much better performances and efficiencies than the existing unmanned recording system using infrared signals by hardware-intensive-technologies.

Key words: Face detection, CAM-shift, FFMPEG, unmanned recording system, image processing, Bluetooth

INTRODUCTION

It turns out that a camera for making video by mobile phone and digital camera is being popularized. In particular, fields of camera for surveillance and lecture as camera application have been much widely used. This study designs the unmanned recording system through video-based tracking object that applying to camera for surveillance and lecture. Although, the system of video-based tracking object is very complicated owing to the software-intensive-technologies that include face detection, CAM-shift and FFMPEG in a field of image processing, it has much better performances and efficiencies than the existing unmanned recording system using infrared signals by hardware-intensive-technologies. In addition as individual internet media such as "Youtube" and "Social Network Service (SNS)" grows up, the necessity of self-video recording is also, on the increase. While individual self-image technologies such as "selfie stick" is being popularized because of many researches individual self-video recording technologies still has not been developed.

Therefore, this study proposes an unmanned recording system in mobile phone using image processing technologies in order to detect and track an object without a person that monitors the object and controls the camera. The automatic tracking object system is designed by the process of detecting and tracking the face of an object. As for automatic tracking object, 'Arduino', a kind of micro-controller is used. Moreover, it can not only help track object in camera frame but also, on the point of getting out of the camera frame it can be used for real-time tracking object with a rotating equipment in motor (Yasukawa *et al*., 2016).

Related works of object tracking

Unmanned recording system: 'Swivl', an unmanned recording system for lectures that uses infrared signals is shown in Fig. 1 (He *et al*., 2015). Swivl is a piece of

Fig. 1: Unmanned recording system for lecture, 'Swivl'

equipment that is used to easily capture and share video using a tablet or smartphone. It is one of the primary examples of unmanned recording systems for lectures. The system has the benefit of being extremely stable for detecting and tracking an object. However, the system is expensive owing to the high costs of producing the equipment. These high costs arise from the fact that designing and producing equipment that uses infrared signals requires hardware-intensive technologies.

In addition, there is the ‘PTZ camera’, an unmanned recording system for surveillance which includes an automatic object-tracking function that is used to maintain security in a company, avoid latent dangers in public facilities or track the boundary of an arm (Raimondo *et al*., 2010; Zhang *et al*., 2016). The PTZ camera, one of the primary examples of an unmanned recording system for surveillance is shown in Fig. 2. The PTZ camera consists of a fixed camera that is attached to a ceiling or wall. Although, such placement can bring stability to the object-tracking function, it can also, cause blind spots. Moreover, the camera operates using infrared signals, its costs for detecting and tracking an object are increased and the price of the product may be extremely expensive.

Object tracking research: There are four major approaches to tracking a moving object: 3D Model-based tracking area-based tracking, active contour-based tracking and feature-based tracking.

Figure 3 shows a schematic diagram of 3D Model-based image-processing strategies. The 3D

Fig. 2: Unmanned recording system for surveillance, ‘PTZ camera’ (Model: B00DPSBV1G)

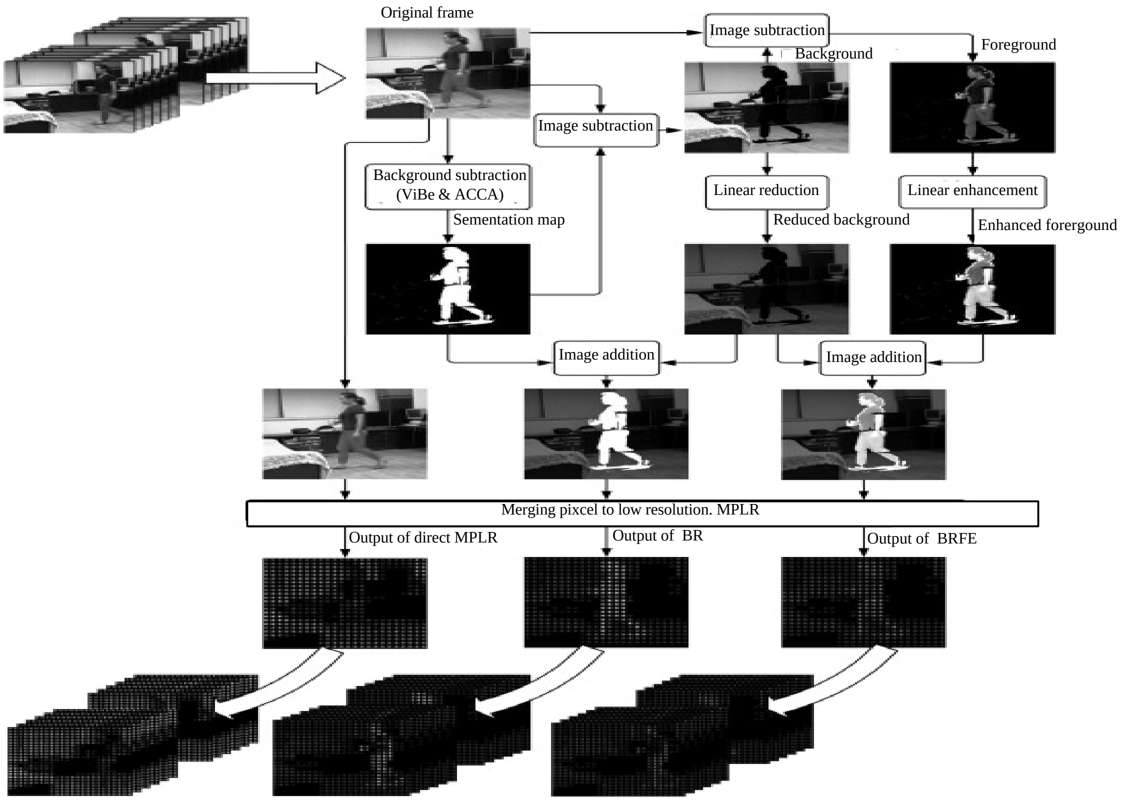

Fig. 3: Schematic diagram of the image-processing strategies

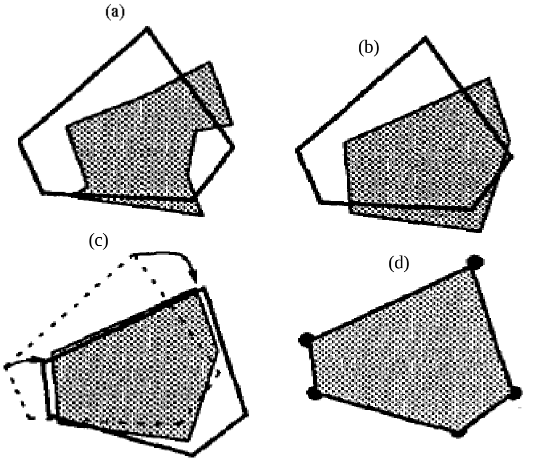

Fig. 4: The measurement algorithm: a) Observation obtained by segmentation (grey region) and prediction (solid line); b) Convex hull of the observation; c) Matching of polygons and d) Effective measurement: vertices of the grey region

are expected to provide more insight into a situation than 2D Models which helps analysts understand the content of captured images (Mastruko, 2009). In addition, a 3D reconstruction could be useful for the visualization of events as can be seen in the forensic casex s described by Neis *et al.* (2000), Subke *et al.* (2002) and Thali *et al.* (2000). A negative thing of this approach is that the procedure described below is still extremely labor intensive and time consuming.

Figure 4 describes the measurement algorithm of region-based tracking (Wang *et al.*, 2015). The primary example is the background subtraction method by Doyle *et al.* (2014). A negative thing of this method is its vulnerability to rapidly changing background images.

Thirdly, in (active contour-based tracking), a part of process of contour-based tracking in active contour-based tracking, a part of the contour-based tracking process is shown in Fig. 5 (Yin *et al.*, 2012). This picture shows contour points using contour-based tracking process independently, hence, contour deformation forms a high-dimensional deformation space. As a result, the application of particle filtering is expensive.

Finally, the target search and feature updating process in feature-based tracking is described in Fig. 6. However, there are still several improvements to be made: an online discriminative algorithm that formulates visual tracking as a classification problem can be added to effectively separate the target from the background, similarly to the concept described by Kalal *et al.* (2012); the proposed algorithm can be extended to track multiple objects in video sequences and in the case of static scenarios, a background subtraction algorithm such as the mixture Gaussian background model (Kalal *et al.*, 2012) can be added to improve the tracking capabilities.

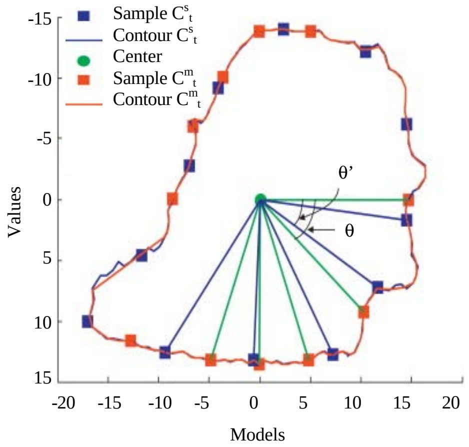

Fig. 5: Convert contour model CTS to CTM

Because each method has positive things and negative things, the negative things must be appropriately offset. Thus, this study proposes a system that harnesses the positive things of each of the three major methods without suffering the associated negative things these methods are area-based tracking, active contour-based tracking and feature-based tracking. Face-detection methods have been applied to detect an object and the CAM-shift method has then been applied to track the object. This approach can maximize the positive things and minimize the negative things of these methods.

The related works of face detection: There are four general approaches to face detection. The first is a method for face detection from a controlled background image that uses a plain solid-color image or fixed background image that was defined in advance. The second method performs face detection using face color. This method is a dominant method in real-time and limited environments because it uses the classic colors of faces to find a face area. The third method is face detection by motion. This method simply calculates a moving face area using real-time video. However, a problem occurs when another object is moving in the background. The last method performs face detection from an unlimited scene. This method employs a neural-network approach using a statistical clustering data and an accurate algorithm for face detection in gray scale.

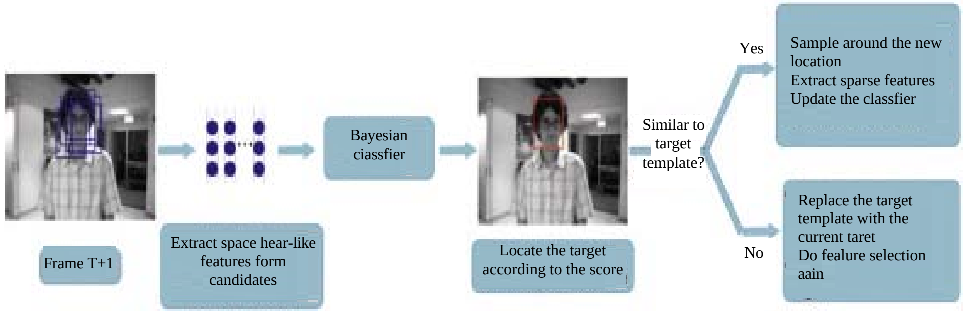
Frame T+1
Extract space hear-like features form candidates
Bayesian ciassfier
Locate the target according to the score
Similar to target template?
Yes
Sample around the new location
Extract sparse features
Update the classfier
No
Replace the target template with the current taret
Do fealure selection aain

Fig. 6 : Target search and feature updating

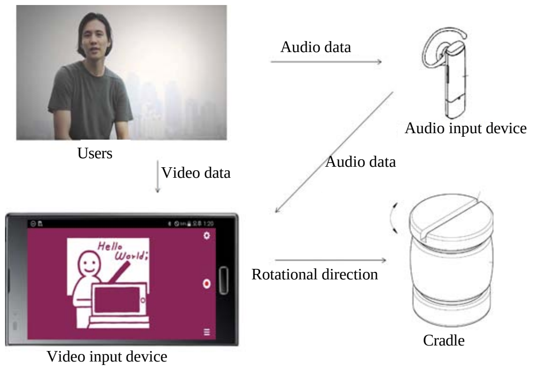

Fig. 7: System structure

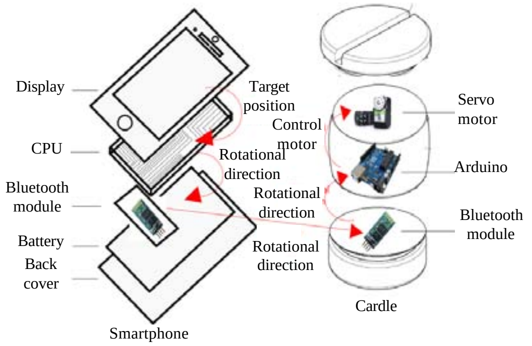

Fig. 8: Process for tracking an object based on video

MATERIALS AND METHODS

Design

System structure: This system consists of a camera device, a wireless ear set and a cradle. The cradle consists of a micro-controller (Arduino) and rotating equipment (servomotor). Figure 7 illustrates the system structure. The components of this structure are a camera device (smartphone) that contains the camera and is able to connect with other equipment by Bluetooth, a wireless Bluetooth ear set with a microphone and a rotating cradle that serves as a brace for the camera device. The camera device consists of a display that shows the user the contents of the screen, a camera that records an object and a processor to execute the program commands. An android smartphone has two cameras: the front camera is placed on the side of the display and the back camera is placed on the opposite side. This system is compatible with both cameras. The cradle has a brace that is connected to a servomotor, enabling the servomotor to rotate the main body. In the brace, there is a furrow that enables the placement of the smartphone on the head of the cradle. The recorded object is equipped with a wireless ear set which enables the object's input audio to be sent to the camera device by Bluetooth. The Bluetooth network is used to transmit audio from the ear set to the android smartphone and for synchronization of audio and video that are transmitted through the audio channel instead of the data channel.

Figure 8 describes a schematization of the process for video-based object tracking. First, the smartphone sends the location coordinates of the object in the display to its CPU. The CPU compares the received coordinates with the coordinates of the screen center and determines whether to rotate and if so where to rotate based on the calculated coordinates. If the absolute value of the difference in (X, Y) is not large, each coordinate is less than or equal to the defined value (threshold) and it does not rotate in any direction, otherwise, it does rotate. Second, the Bluetooth module chip in the smartphone sends the determined rotational direction to another Bluetooth module chip that is built into the cradle, to which it is connected via. Arduino. Third, the chip in Arduino sends the rotational direction to the Arduino controller. Finally, the Arduino tells the servomotor to rotate based on the received rotational direction. Consequently as the servomotor rotates the head of the cradle, the smartphone on the head of the cradle rotates according to the controlled direction and can assist in the real-time tracking of the object.

As Fig. 9 shows, this study proposes an unmanned recording system by video-based tracking object consisting of such seven steps. First step is the step of

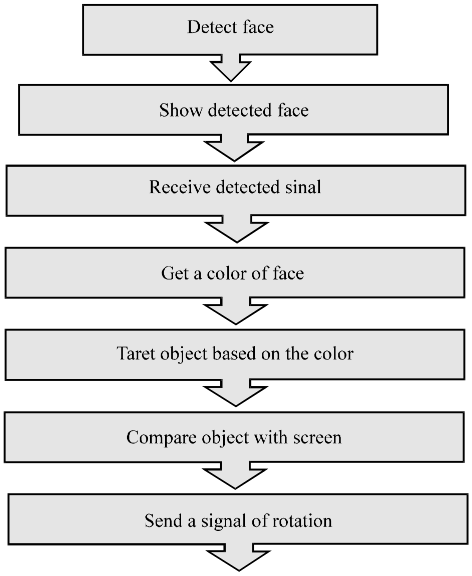
Detect face
Show detected face
Receive detected sinal
Get a color of face
Taret object based on the color
Compare object with screen
Send a signal of rotation

Fig. 9: Seven steps in video-based trackin object

detecting a face through camera connected with device. Second step is the step of showing the detected face based on the face area. Third step is the step of receiving the signal of detection for checking whether marked area of face is object. Fourth step is the step of getting data of color in the selected object. Fifth step is the step of setting coordinates of target according to their object's location. Sixth step is the step of comparing the object's coordinates with particular coordinates of screen from camera. Seventh step is the step of sending a rotating equipment a signal having the rotational direction in order to decrease the compared differences.

Besides, the process of face detection and tracking object, recording process consists of three steps. First step is the step of recording video from camera. Second step is the step of receiving audio from an auditory input instrument equipped with object. Third step is the step of storing consequent video synchronized with audio and video.

RESULTS AND DISCUSSION

Face detection and object tracking: In the process of face detection, we can extract the area of face that can be guessed according to a video image obtained from smartphone camera. In that process, several features are used for detecting face and the type of image is not RGBA but gray. However, RGBA type is required to be seen on the screen. Thus, both gray image and RGBA image should be obtained from camera. In this study, we use face-detection in openCV library to process gray image value and RGBA image value.

Figure 10 describes the process of face detection. In general, a man's face appears that in their eyes both light and shade are dark and in nose both light and shade are bright. In face-detection library, the pattern of face is obtained using these both light and shade (Shen *et al.*, 2016). It is seen from this figure that after a rectangular image in gray type is overlapped on detection area to get determined whether this area is face or not. The mean of pixels value in bright area is compared with that of pixels value in dark area. Then, if the compared difference is more than threshold, it can be said that this detection area has a feature of face. This is because there is little difference both light and shade according to particular area. Although, faces of people are various, the pattern of faces is similar to each other.

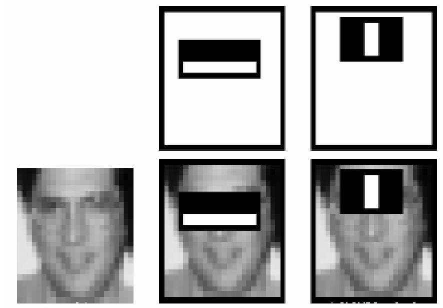

Fig. 10: Process of face-detection

Algorithm 1; Face-detectuin:

```
if (mCascade ! = null) {
  int heiht = m(Gray.rows()
  int faceSize = Math.round(height*FdActivity.minFaceSize)
  List<Rect>face = new LinkedList<Rect>()
   mCascade.detectMultiScale(mGray, faces, 1.1, 2, 2, new Size(faceSize,
faceSize))

  for(Rect r:faces)
  Core. Rectanle(mRgba, r. t1(), r. Br(), Scalar(0, 255, 0, 255), 3)
}
```

Algorithm 1 describes the algorithm of face detection. The height of the whole image is calculated by rows in gray matrix and the detection area of face in the whole image is determined. For storing the detected areas, lists are defined and find a face. The perimeter(top left, bottom right) of rectangular in the found face is set and the detected area is marked by a green line of the thickness in level 3. Obtaining a color data in the detected area, we can resize the search window through contracting and expanding this size based on the color data.

Figure 11 describes CAM-shift algorithm in object tracking algorithm. This algorithm is improved from mean shift algorithm. The mean shift algorithm is tracking of ROI (Region Of Interest) object algorithm in high speed founded on density distribution(features, corners, colors) in data set. This algorithm is that from initial search window when it selects size and location it can extract ROI

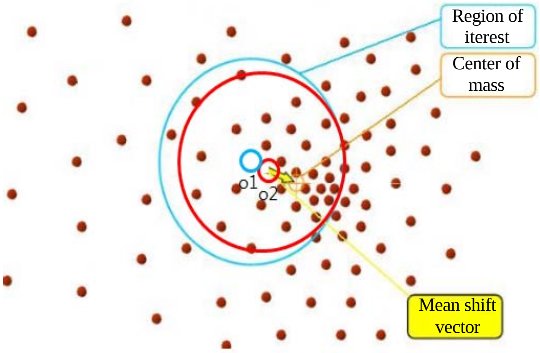

Fig. 11: Process of CAM-shift

object and determine the boundary based on the area including the defined color (Zheng *et al*., 2015). The CAM-shift algorithm is complemented from mean shift in color segmentation method in order to utilize in streaming environment. In particular, it is complemented by resizing the size of search window for itself. Initial window is blue one. This coordinates of search window's center are marked as 'o1'(blue one) in blue window. However, if you find a real center of mass in blue window, it is 'o_2'(yellow one). Certainly, there is a difference between o1 based on the blue window and o_2 based on the red window. Accordingly, move the blue window so, that 'o1' is to be the center of mass. The red window is center of mass. In most cases, next step may be not correct to the center of mass. Subsequently, move again the center of window based on the real distribution. This process iterates until the center coordinates of window is correct to the center of mass. Then, the moving window can consider red points at most and update their center coordinates of window to correct center of mass. Finally, we can get center of mass:

Alghorithm 2; Determinin rotations:

```
$diff_x$ = $ScreenCenter_x$-$Target_x$
$diff_y$ = $ScreenCenter_y$-$Target_y$

Turn left  if $diff_x$> $threshold_x$
Stop       if -$threshod_y$< $diff_y$
Turn right if $diff_x$< $threshold_x$

Tilt up    if $diff_y$>$threshold_y$
Stop       if -$threshod_y$< $diff_y$
Tilt down if $diff_y$< $threshold_y$
```

Algorithm 2 and Fig. 12 describe the rotation algorithm and an example of determining a rotational direction, respectively. The center coordinates of the search window are compared with the center coordinates of the camera frame. If the difference is greater than the threshold, the cradle holding the smartphone camera rotates to the left or right. Figure 13 illustrates that the center coordinates of the camera frame (X1, Y1) are (400, 320) and the center coordinates of the search window (X2, Y2) are (300, 160). Because the difference in each (X, Y) is (X1-X2, Y1-Y2) = (400-300, 320-160) = (100, 160), each difference is positive. In this case, the cradle holding the smartphone camera is required to turn left and up for a particular object to be located in the center position in the camera frame. However, if we instruct the cradle to stop rotating only when the corrected difference is precisely 0, it will rotate microscopically and infinitely. Thus, the quality of the recorded video will become lower. Therefore, we should define a threshold in advance and command the cradle to stop rotating if each difference of (X, Y) is not more than the threshold. For example, consider that we define the threshold as ±50, the center coordinates of the camera frame (X1, Y1) as (380, 200) and the center coordinates of the search window (X2, Y2) as (400, 320). Because the differences in each coordinate (X1-X2, Y1-Y2) are (20, -120), the cradle did not rotate on the X-axis but did rotate on the Y-axis.

Synchronization with audio and video: Audio input by the built-in microphone of a wireless Bluetooth (Gentili *et al*., 2016) ear set and video recorded by the smartphone are synchronized by FFMPEG. Although, Android OS has a library of functions for video recording, storing and playing, we use FFMPEG instead of this library because this system must process each frame using image-processing libraries such as face detection and CAM-shift. Therefore, after the recorded frames are processed by these image-processing libraries, all these frames are gathered by FFMPEG and finally, the video is made.

FFMPEG is an open-source multimedia framework supporting a cross platform. When we use FFMPEG, we can utilize nearly all multimedia functions such as 'playing video', 'encoding/decoding', 'transcoding', 'muxing/demuxing' and 'stream'. The description about FFMPEG shows several detailed library:

- Libavcodec: encoder and decoder in audio/video
- Libavformat: muxer/demuxer in container format of audio/video
- Libavutil: a variety of utility required when FFMPEG is developed
- Libpostproc: video post-processing
- Libswscale: image scaling in video, color-space transformation of pixel-format
- Libavfilter: edit and examination of audio/video between encoder and decoder
- Libswresample: audio resampling

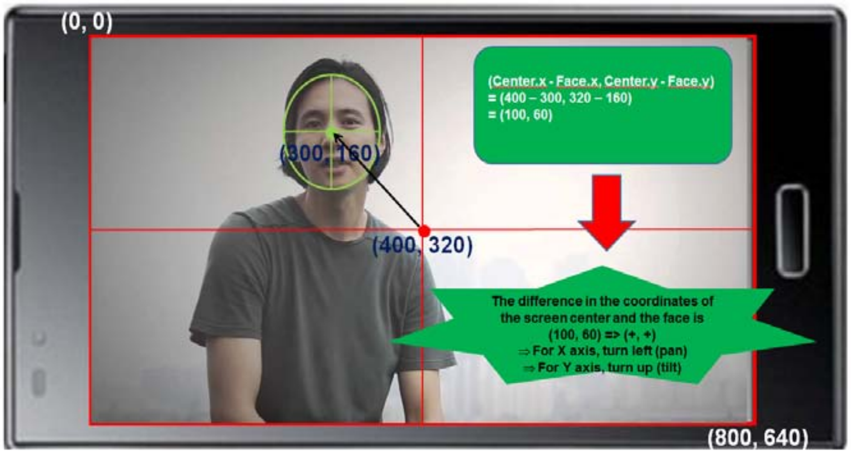
(0, 0)
(300, 160)
(400, 320)
(800, 640)
(Center.x - Face.x, Center.y - Face.y)
= (400 – 300, 320 – 160)
= (100, 60)
The difference in the coordinates of the screen center and the face is
(100, 60) => (+, +)
⇒ For X axis, turn left (pan)
⇒ For Y axis, turn up (tilt)

Fig. 12: Example of determinig rotation

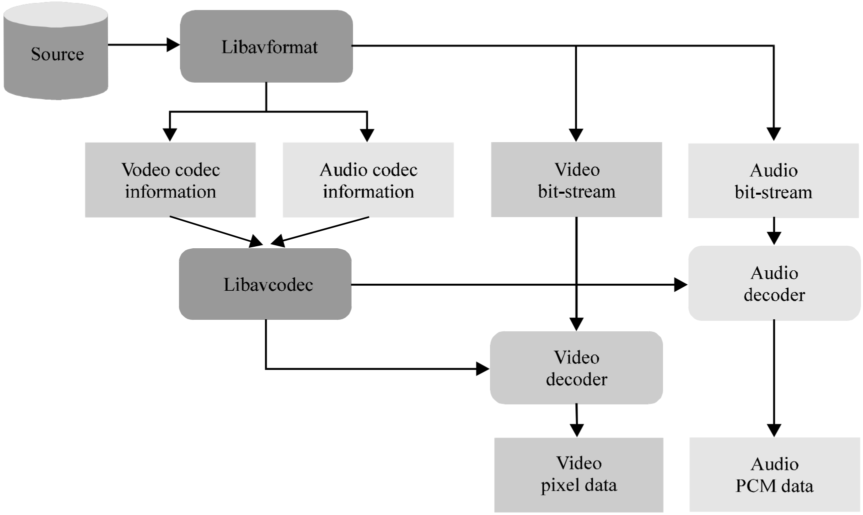

Fig. 13: Process of FFMPEG

Figure 13 illustrates that the process of extracting an original data through reading the video file. Therefore, each of the recorded video frames are processed by face detection and CAM-shift. Then, they can be stored by FFMPEG in order to make a video.

CONCLUSION

This study proposes instead of the existing and expensive unmanned recording system through video-based tracking object for surveillance or lecture, an accessible and inexpensive one. The feature of this system is moving the search window, contracting or expanding based on the area of moving object continuously utilizing the detected face in video, the color data in the area and the center of moving object. Subsequently when the motion of object is more than the defined threshold, the object is tracked by servo motor rotating to the direction in the motion of object. The unmanned recording system this study proposes is designed by the technology of image processing unrelated to the technology of hardware-based such as infrared signals that the existing unmanned recording system utilize. This aspect can say that it is the technology-compact system and which enables to make the product in mass production in very low price. In a next research, we leave the point that this system is susceptible to light as our future works. This is because that in a heavily dark or bright place, the accuracy of object tracking by CAM-shift algorithm applying this system is decreased comparing with in a place having a normal light. Thus, it needed to research about newly detection of feature in order to improve effectively.

REFERENCES

Doyle, D.D., A.L. Jennings and J.T. Black, 2014. Optical flow background estimation for real-time pan/tilt camera object tracking. Meas., 48: 195-207.

Gentili, M., R. Sannino and M. Petracca, 2016. Bluevoice: Voice communications over bluetooth low energy in the internet of things scenario. Comput. Commun., 89: 51-59.

He, Y.J., M. Li, J. Zhang and J.P. Yao, 2015. Infrared target tracking via weighted correlation filter. Infrared Phys. Technol., 73: 103-114.

Kalal, Z., K. Mikolajczyk and J. Matas, 2012. Tracking-learning-detection. Trans. Pattern Anal. Mach. Intell., 34: 1409-1422.

Mastruko, V., 2009. Reconstruction: Three Dimensional. In: Wiley Encyclopedia of Forensic Science, Jamieson, A. and A. Moenssens (Eds.). John Wiley and Sons Ltd., Chichester, UK., ISBN:978-0-470-01826-2, pp: 2257-2267.

Neis, P., T. Fink, M. Dilger and C. Rittner, 2000. Use of the software Poser4 in reconstruction of accident and crime scenes. Forensic Sci. Intl., 113: 277-280.

Raimondo, D.M., S. Gasparella, D. Sturzenegger, J. Lygeros and M. Morari, 2010. A tracking algorithm for PTZ cameras. IFAC. Proc. Volumes, 43: 61-66.

Shen, X., X. Sui, K. Pan and Y. Tao, 2016. Adaptive pedestrian tracking via. patch-based features and spatial-temporal similarity measurement. Pattern Recognit., 53: 163-173.

Subke, J., S. Haase, H.D. Wehner and F. Wehner, 2002. Computer aided shot reconstructions by means of individualized animated three-dimensional victim models. Forensic Sci. Intl., 125: 245-249.

Thali, M.J., M. Braun, W. Bruschweiler and R. Dirnhofer, 2000. Matching tire tracks on the head using forensic photogrammetry. Forensic Sci. Intl., 113: 281-287.

Wang, Z., J. Wang, S. Zhang and Y. Gong, 2015. Visual tracking based on online sparse feature learning. Image Vision Comput., 38: 24-32.

Yasukawa, S., H. Okuno, K. Ishii and T. Yagi, 2016. Real-time object tracking based on scale-invariant features employing bio-inspired hardware. Neural Netw., 81: 29-38.

Yin, J., C. Fu and J. Hu, 2012. Using incremental subspace and contour template for object tracking. J. Netw. Comput. Appl., 35: 1740-1748.

Zhang, P., T. Zhuo, L. Xie and Y. Zhang, 2016. Deformable object tracking with spatiotemporal segmentation in big vision surveillance. Neurocomputing, 204: 87-96.

Zheng, H., X. Mao, L. Chen and X. Liang, 2015. Adaptive edge-based mean shift for drastic change gray target tracking. Opt. Intl. J. Light Electron. Opt., 126: 3859-3867.