

Object Tracking and Identification using a camera in Digital Signage System

Hyun-suk Roh*, Nam-kyung Lee*, Sang-ho Lee**

*Electronics and Telecommunication Research Institute (ETRI), Daejeon, Korea

**Computer Science Department, Chungbuk National University, Cheongju, Korea

ventus@etri.re.kr, nkleee@etri.re.kr, shlee@cbucc.chungbuk.ac.kr

Abstract— This paper will outline the software specification and design of the Multi-camera Object Tracking and Identification System (MOTIS) for Digital Signage System. This document will provide detailed views of the system's design in order to facilitate communication and understanding of the system. It intends to capture and convey the detailed architectural and design decisions that have been made for the MOTIS.

In this paper, we propose several methods to increase accuracy of device's position.

Keywords— identification, tracking, face, camera

I. INTRODUCTION

Face recognition technology using the biometric characteristics is one of those Biometric technology that recognizes the identity of each person. Face recognition technology is known as, by its nature, non-aggressive and non-intrusive method. Due to its User-friendly biometric sense. When the biometric information is used for the purpose of identity authentication, it has the advantage that a user can be recognized as a natural way.

Research on face recognition in the last decades of Image Processing is based on various research fields, such as signal processing, computer vision, pattern recognition, machine learning, Databases, statistics and linear algebra techniques, optimization and remarkable outputs are reported steadily.

Applications for the initial face recognition technology such as access control system and bank card identification is used in a simple application that verifies the user's identity. Research and systems development Analyzing surveillance video via a CCTV camera have been developed steadily since 911 terror in U.S. for the safety of the immigration security management and public places.

In example, using face recognition and searching technology, system detects a terrorist from the image obtained from a CCTV. The system automatically identifies the terror and crime suspects. In the CCTV footage detecting a face, the face detection is predefined.

Another example is identify the location of a person. After a face is detected, a person is recognized as the same

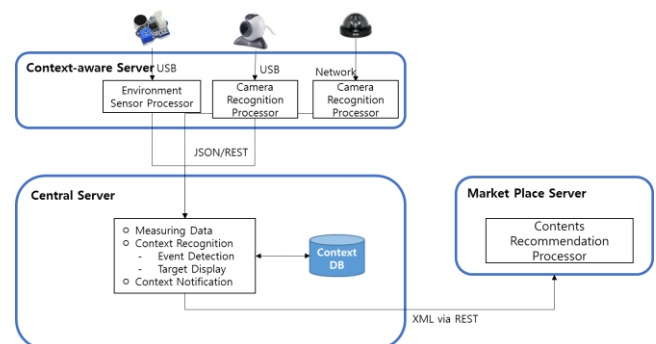
person when the user moves to another location. We can provide a customized advertising service using the surrounding information relating person's location.

On the other hand, the development of smart phones and latest communication technology trigger another development of technology such as classified electronic album, video calls, Entertainment, human robot interaction.

The attempt to apply face recognition to the searching engine and internet services is also in progress.

The remainder of this paper is organized as follows. We briefly introduce object identification and tracking system in Section 2. Section 3 describes hardware modules and Section 4 does software modules for this system. Section 5 describes detailed design and finally, we draw our conclusions in Section 6.

II. OBJECT IDENTIFICATION AND TRACKING SYSTEM



This system consists of hardware and software modules. The hardware modules include Face Camera, Dome Camera, and Environment Sensor. And, the software consists of three modules, namely: Face Recognizer, Object Tracker, and Sensor Reader. Each module will handle a specific task to achieve the system goals. Figure 2 shows the logical overview of this technology.

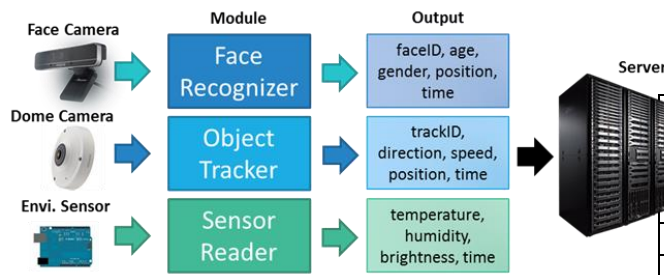


Figure 2. System Overall

III. HARDWARE MODULES

A. Face Camera

This system uses 3D depth (Intel RealSense F200) camera to capture facial appearance of objects. Table 1 shows minimum system requirements that must be satisfied to make this device working properly:

Table 1. Face Camera

Minimum Requirements	Sample Figure
OS: Microsoft Windows 8.1 Update (64 bit) 4 GB RAM 4 GB free hard disk space USB 3.0 port <u>Firmware:</u> - Depth Camera Manger 1.3.20.55679 - SDK 5.0.3.18777	

B. Dome Camera

This system will use 360 degree fisheye IP-based (Samsung SNF-7010) dome camera to capture object from above (bird-eye) view point. Table 2 shows recommended PC specifications to work with this camera:

Table 2. Dome Camera

Minimum Requirements	Sample Figure
CPU: Intel Core 2 Duo 2.4 GHz or higher RAM: 2GB or higher OS: WinXP, VISTA, 7, 8 Video Memory: 256 MB or higher	

C. Environmental Sensor

This system uses Arduino Uno as the main board of the sensor. It also consists of smaller components to receive specific environmental information shown in Table 3. This device is connected to the PC via USB-port. It sends the data through the serial (COM) port. Table 3 shows the schema of environmental sensor hardware configuration.

Table 3. Environmental Sensors

No.	Component	Function
1	Arduino Uno	Main board
2	DHT 11	Humidity & temperature sensor
3	Photocell	Light intensity Sensor
4	Passive IR	Motion sensor
5	LED	Activity indicator
6	Potentiometer	Sensor Read Cycle Controller

IV. SOFTWARE MODULES

A. Face Recognizer Module

This module is mainly used for object identification. It uses images captured by face camera as input, and generates several information based on facial features, such as: object ID, age-group, and gender. Figure 3 illustrates the block diagram of this module:

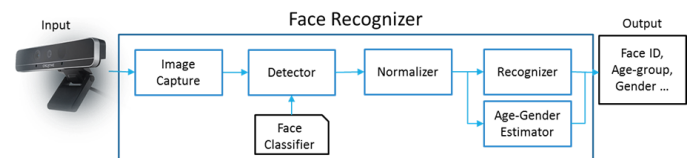


Figure 3. Face Recognizer Components

There are several components which are required to build this module:

1. Image capture for acquiring image from camera.
 - Input: Face camera stream
 - Output: Image
2. Detector for locating face and eyes region.
 - Input: Image
 - Output: Face image (face and eyes region)
3. Face Classifier for assisting Detector component in finding face and eyes region.
4. Normalizer for standardizing face image obtained from face camera so that it can be used for recognition and estimation process.
 - Input: Face image
 - Output: Normalized face image (aligned and cropped)
5. Recognizer for extracting facial features and finding matching face in database.
 - Input: Normalized Face image
 - Output: Face ID
6. Age-gender Estimator for calculating approximate gender and age group of the face.
 - Input: Normalized Face Image
 - Output: Age and gender

B. Object Tracker Module

This module is mainly used for tracking object movement. It can track several objects which move on a certain room or location. It uses images captured by dome camera as input, and generates track ID, and object movement trajectories. Figure 4 shows the block diagram of this module.

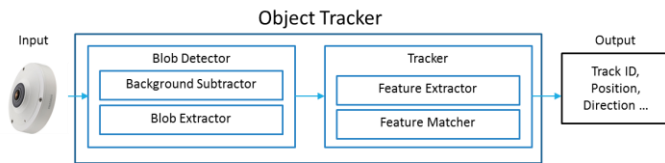


Figure 4. Object Tracker Components

This modules consists of two major components:

1. **Blob Detector** for extracting moving objects information.
 - a. **Background Subtractor**: it performs background-foreground segmentation.
 - Input: Image
 - Output: Foreground Mask
 - b. **Blob Extractor**: it extracts object's bounding box
 - Input: Foreground Mask
 - Output: object bounding box
2. **Tracker** for extracting and analyzing object appearance and location.
 - a. **Feature Extractor**: it extracts object's feature
 - Input: Image, object bounding box
 - Output: object feature
 - b. **Feature Matcher**: it finds matching feature to track object movement.
 - Input: image, object feature and bounding box
 - Output: updated object feature and bounding box

Object bounding box is a 2-dimensional rectangle representing object's current location. And, object feature is a set of important points (x-y coordinate) and pixel representing object's general appearance.

C. Environment Sensor Reader

This module is mainly used for obtaining environmental information such as: temperature, humidity, light intensity, etc. It uses Arduino and various sensors in the hardware layer. To read the environmental information, the device is equipped with a pre-installed firmware. As client software, Sensor Reader will read all of the raw data from sensors and perform further data processing. Figure 5 shows the block diagram of this module.

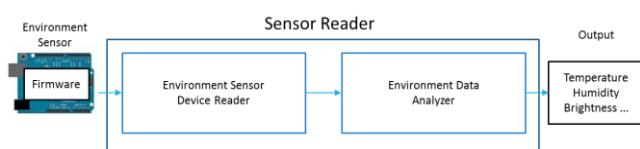


Figure 5. Sensor Reader Components

On the client software layer, there are two major components:

1. **Device Reader** for receiving raw data from sensor devices.
 - Input: Raw Sensor Data
 - Output: Formatted Sensor Data
2. **Data Analyzer** for analyzing data and detecting significant changes regarding environmental condition.
 - Input: Formatted Sensor Data
 - Output: Significant Change in Sensor Data (Temperature, Humidity, etc.)

V. DETAILED DESIGN

A. Face Recognizer Module

Initially, this module will acquire an image from face camera. After that, it will try to detect the existence of face(s) inside the captured image. If there is one or more faces within the image, then it will normalize the face(s). This normalization process is required since the face may appear arbitrarily which will make the recognition process becomes very hard. Thus, standardizing the face image by aligning it in the eye level, cropping non-relevant area outside the face region, and resizing the face into a predefined size will help this module to recognize the face more accurately. Next, the system will try to recognize the normalized face. If it is recognized, then the system will get the face ID. Otherwise, the system will store the new face and generate a new face ID. Finally, the system will try to approximate the age and gender of the normalized figure. 5 illustrates the processing steps conducted by this module to identify an object by using its facial appearance.

Table 4. Face Recognition Process Description

No.	Process	Input	Output	Description
1	Acquire Image	Camera Stream	Image	
2	Detect Face	Image	Face Image	Raw face image
3	Normalize Face	Face Image	Normalized Face	Standardized face image
4	Recognize Face	Normalized Face	Recognition status	Yes/ No status
5	Get Face ID	Normalized Face	Face ID	Unsigned integer ID
6	Store New Face	Normalized Face	New Face ID	Newly generated face ID
7	Estimate Age/G.	Normalized Face	Age and gender	Approximate age and gender.

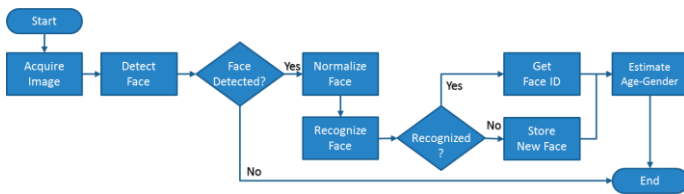


Figure 6. Face Recognition Processing Flow

In this section, several entities which will be useful to represent data during face recognition process is described. There are three types of entities:

1. **CAPTURE** represents capture device data.
2. **FACE** represents face information.
3. **REPORT** represents message which will be sent to the server.

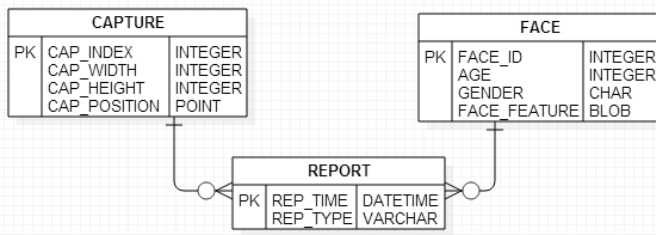


Figure 7. Face Recognition Data Structure

B. Object Tracker Module

Initially, this module capture an image from dome camera. And then, the captured image will be analyzed to obtain the foreground mask which represents the moving object image mask. Next, the blob extractor will encapsulate object location by creating bounding box (2D rectangle) around it. If there is a detected blob, then the feature extractor will analyze the object appearance and find the important point and pixel which can be used for tracking purpose. After that, feature matcher will find any matching feature on the database to check whether it is a new or previously tracked object. If it is a new object, then a new object tracker will be created. Otherwise, if it is an already tracked object, then the system will get the track ID and update object feature and bounding box. The object movement direction can be calculated by analyzing object's previous and current location. Figure 8 shows the processing flow of Object Tracker module.

Table 5. Object Tracking Process Description

No	Process	Input	Output
1	Acquire Image	Camera Stream	Image
2	Subtract Background	Image	Foreground Mask
3	Extract Blob	Foreground Mask	Bounding Box
4	Extract Feature	Image, Bounding Box	Object Feature
5	Match Feature	Image, Object Feature & Bounding Box	Feature matching status
6	Get Track ID	Image, Object Feature & Bounding Box	Track ID
7	Create New Track	Image, Object Feature & Bounding Box	New Track ID

8	Update Tracker	Track ID, Image, Object Feature & Bounding Box	Track ID, New Object Feature & Bounding Box
---	----------------	--	---

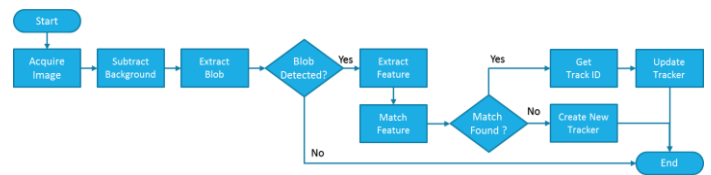


Figure 8. Object Tracking Process Flow

To support the operation of Object Tracker, there are several data types which must be stored by the system:

1. **DOME_CAPTURE** represents dome camera configuration data.
2. **TRACKED_OBJECT** represents object which is tracked by the system.
3. **REPORT** represents message which will be sent to the server.

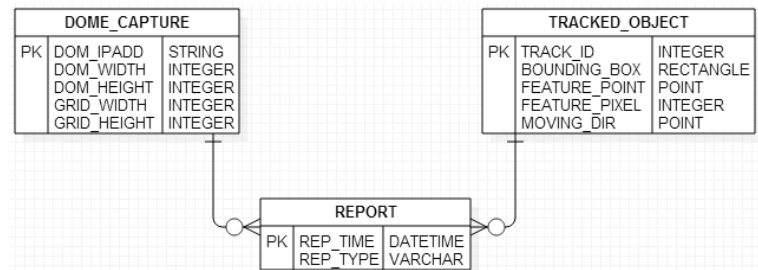


Figure 9. Object Tracking Data Structure

C. System Operation View

Figure 10 show actual system operation. Face recognition module allocate a ID and tracker module trace it.

Our system can trace multiple object at same time and we improve some errors like identifying non-human object.

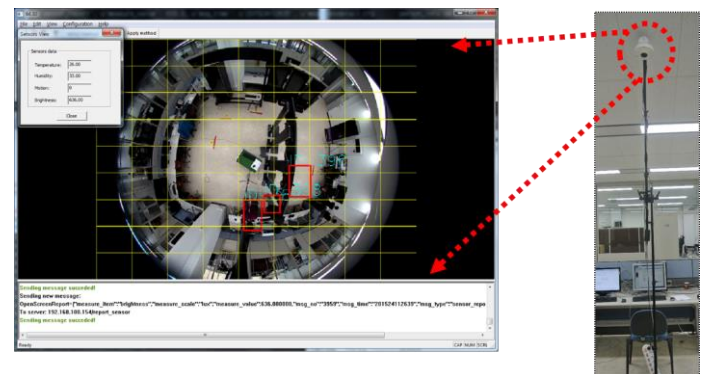


Figure 10. Operation View

VI. CONCLUSIONS

In this paper, we proposed the software specification and design of the Multi-camera Object Tracking and Identification System. We explained detailed solutions for object tracking using the depth-camera and dome-camera. Our solution is about algorithm for acquiring user location using camera images.

Using proposed method we can provide various user-targeting services in advertising services. In the future, we will expand our schemes to various environments and increase accuracy of device's position.

Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea. [B0184-15-1002, Development of Smart Signage System linked with IoT Standard Platform]

REFERENCES

- [1] W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld, Face recognition: A literature survey, *Acm Computing Surveys (CSUR)* 35 (4) (2003). pp. 399–458.
- [2] Z. Kalal, K. Mikolajczyk, J. Matas, Face-tld: Tracking-learning-detection applied to faces, in: *Image Processing (ICIP)*, 2010 17th IEEE International Conference on, IEEE, 2010, pp. 3789–3792.
- [3] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction," in *ICPR*, 2004.
- [4] S. Murgia and L. Nardelli, "Opencvblobslib," *OpenCVBlobsLib*, 2015. [Online]. Available: <http://opencvblobslib.github.io/opencvblobslib/>.
- [5] Z. Kalal, K. Mikolajczyk and J. Matas, "Tracking-Learning-Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409-1422, 2012.
- [6] Z. Kalal, K. Mikolajczyk and J. Matas, "Forward-Backward Error: Automatic Detection of Tracking Failures," in *International Conference on Pattern Recognition*, Istanbul, 2010.
- [7] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Descriptor of the algorithm," *Intel Microprocessor Research Labs*, 1999.