Image-based Intelligent Surveillance System with the Robust Universal Middleware Bridge Service

Chia-Hsu Kuo [1], Huan-Ming Hsu [2]

[1.] Department of Software Engineering, National Kaohsiung Normal University, Kaohsiung, Taiwan
[2.] School of Electronics and Computer Science, University of Southampton, United Kingdom
kuoch@nknu.edu.tw

Abstract: Considerable emphasis has recently been placed on surveillance topics and homeland security because of the prevalence of terrorist attacks over the past decade. The development of surveillance equipment is progressing from analog CCTV cameras (which were prevalent in the early days) toward digital IP cameras (which are currently prevalent). Image/object surveillance topics have constituted the main interest for many studies in recent years. This paper proposes an image-based intelligent surveillance system (IISS) integrated with the Universal Middleware Bridge Service (UMBS) for IP cam networking. The UMBS provides mechanisms related to manual system setting, automatic configuration, and management for improving the entire setting and installation procedures. The UMBS comprises four main functional modules, namely live video, Playback Video, Intelligent Scheduler, and System Configuration modules. The structure of the robust UMBS ensures adaptability and flexibility in designing and developing intelligent image recognition surveillance systems based on IP cam networking. Therefore, in this study, an efficient implementation of the IISS integrated with the robust UMBS was achieved. Furthermore, an image/object recognition and matching service was implemented for the IISS with UMBS.
[Chia-Hsu Kuo, Huan-Ming Hsu. **Image-based Intelligent Surveillance System with the Robust Universal Middleware Bridge Service.** *Life Sci J* 2015;12(7):76-87]. (ISSN:1097-8135). http://www.lifesciencesite.com. 9

Keywords: Image Recognition; Middleware; Universal Middleware Bridge Service (UMBS); IP Camera; Platform.

1. Introduction

Intelligent surveillance system attracts people worldwide and is considered in recent years to be the key technology for global industrial and commercial development activities. Many studies have emphasized combining framework and application services with cloud computing.

The development of surveillance equipment is progressing from analog CCTV cameras (which were prevalent in the early years) to digital IP cameras (which are currently prevalent). Image surveillance topics have been the main interest for numerous studies. Two international alliances, namely the PSIA (Physical Security Interoperability Alliance) [1] and ONVIF (Open Network Video Interface Forum) [2], were proposed in 2008 for front-end/rear-end integrated protocols based on a network image surveillance system. According to a market survey, so far, numerous products for the software underlying a surveillance system include claims of being able to fully support the mentioned international alliances. However, the ONVIF core specification Version 1.0 was first proposed in 2008. The ONVIF core specification Version 2.0 was then updated in 2010. In 2012, the ONVIF core structure specification Version 2.1.1 was launched. For users and vendors of IP cam, the uncertain protocol of the ONVIF core specification causes difficulty and inconvenience in developing the software underlying a surveillance system because of setting, installation, and management problems. The problem associated with an unratified protocol and unstable surveillance system including hardware and software is still encountered in the development of such high-cost systems. On the basis of the highlighted observations, the current study proposes an image-based intelligent surveillance system (IISS) integrated with a Universal Middleware Bridge Service (UMBS) software platform for IP cam networking. From the viewpoints of end users, enterprise subscribers, and technology vendors, the complexity encountered in the web-based connection and installation of IP cam devices and equipment having a network-supported interface is higher than that encountered in the connection and installation of those without a network-supported interface. However, barriers exist in human-cyber interaction. No support mechanism has so far been elaborated for offering end users, enterprise subscribers, and technology vendors a high degree of freedom and an intuitive method for interacting with, accessing, and managing the various IP cam devices provided by diverse vendors. The UMBS provides the related mechanisms for manual system setting, automatic configuration, and management to improve the entire setting and installation procedures. The UMBS comprises four main functional modules: Live Video (LV), Playback Video (PV), Intelligent Scheduler (iScheduler), and System Configuration (SC) modules. The structure of the robust UMBS ensures adaptability and flexibility in designing and developing IP cam application service systems.

In this study, an image/object recognition and matching service was also implemented for the proposed IISS with UMBS. The rest of this paper is organized as follows. Section 2 reviews the related technical works regarding ONVIF and PSIA for the front-end/rear-end integrated protocols applied in this study. The system structure, system design, and system flow of the UMBS for IP cam networking is depicted in Section 3. Section 4 presents the system implementation of the UMBS. Section 5 presents the proposed design flow and search algorithm, which was used to conduct the image recognition and matching for the IISS with UMBS. Conclusions are presented in the final Section.

2. Related Work

The scope of this section is limited to ONVIF specifications for the front-end/rear-end integrated protocols applied in this study. In 2008, three companies, namely Axis Communication, Bosch Security System, and Sony Corporation, launched the ONVIF, called ONVIF alliance. The objective of the ONVIF alliance is to promote the integration of heterogeneous IP cam devices developed by different vendors. Meanwhile, the ONVIF alliance devotes to the interface standardization and interoperability of products supported by hardware vendors, software developers, and system integrators [2]. In 2011, Wu [4] proposed a device searching mechanism for verifying a surveillance system based on the ONVIF protocol. It focused only on the device discovery and system verification. Tsai et al. [5] presented a distributed multimedia content application in an ONVIF surveillance system. However, those studies did not demonstrate the importance of the ONVIF protocol for front-end/rear-end integrated protocols applied in surveillance systems. In [6], Kuo et al. proposed a conceptual universal middleware bridge system for IP cam networking. It can be applied to the related applications for future integration in surveillance systems.

The ONVIF core specification Version 1.0 was first proposed in 2008. The ONVIF core specification Version 2 was then updated in 2010. In 2012, the ONVIF core structure specification Version 2.1.1 was launched. In mid-2012, the latest ONVIF of core structure specification Version 2.2 was updated and the total number of members of ONVIF is currently approaching 363. As illustrated in Figure 1, the ONVIF framework is divided into three layers: network layer, transport layer, and application layer [2][3]. We organize the ONVIF framework based on media plane and control plane by ONVIF specification. TCP/IP protocol suites including TCP/UDP and IPv4/IPv6 operate on the transport layer and the network layer, separately. Transport layer protocols are applied for data exchange by using WS-Discovery, WS-Addressing, and WS-Security for web services and XML schema, SOAP, and WSDL for interoperability of web services in the application layer. The related web services and the developed ONVIF applications are finally open to system vendors and integrators in the application layer and user interface. Based on the observations, the ONVIF framework is more flexible for its higher interoperability.

Apart from ONVIF, other companies, namely Cisco System, Genetec, IBM, ObjectVideo, Panasonic, Pelco, Texas Instruments, and Verint, proposed the PSIA specification in 2008. The objective of the PSIA is to promote the interoperability and the interface standardization of heterogeneous IP cam devices developed by various vendors, software developers, and system integrators. PSIA specifies the media devices and related Application Programming Interfaces (APIs) included in system management, networking connection, audio/video IO, Pan/ Tile/ Zoom, dynamic detection, AV streaming, and event-driven processing and setting [1]. The competition between ONVIF and PISA has not settled yet. However, the ONVIF performs with an increasing market penetration.

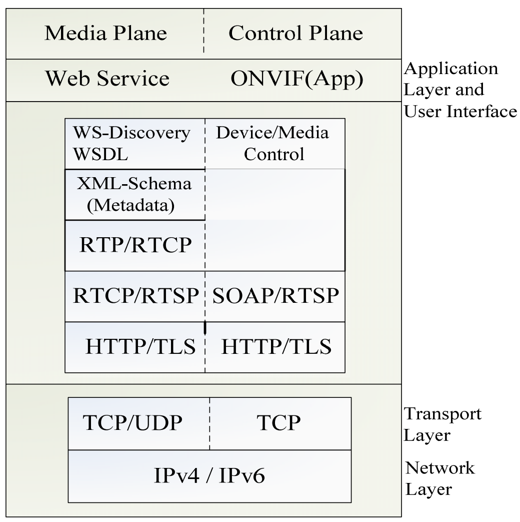


Figure 1. ONVIF framework.

WSDL (Web Services Description Language) defines the detailed specifications in the implementation description and interface description in XML format for web services. WSDL conforms to the development structure of web services approved by W3C (WWW Consortium). WSDL abstractly describes an XML-based public interface, which is concerned with how web services communicate. Specifically, the related agreements and data formats

(Types, Message, Port Type, and Binding) of the defined web services in the service directory are exchanged interactively. Then, the practical network agreement and data formats bind to the desired web services for physical implementation [7].

SOAP (Simple Object Access Protocol) is a standard communication protocol for web services. SOAP simplifies the procedures of fetching an XML-based data format from web servers for communications between web services through HTTP protocol. This XML-based communication procedure between web services is not associated with any programming language or system hardware. This procedure is a type of platform-independent system [8].

Figure 2 illustrates the web service development structure approved by W3C; in this structure, SOAP unifies a standard communication protocol for web services, and WSDL defines the detailed specifications in the implementation and interface description for web services in XML format [9]. SOAP is used for transferring XML-based data to communicate with various objects operated between service providers and clients. A service provider capitalizes on a WSDL document containing detailed specifications of web services by using SOAP. A client obtains the complied and integrated WSDL document and the driver of the discovered web services from the service provider. The expected web service is executed for the client. The detailed web-service-based development principles and specifications are provided in the latest version of ONVIF Core Specification [2].

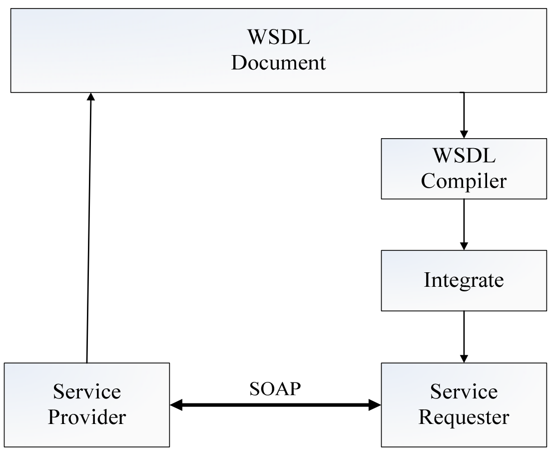


Figure 2. Web service development structure.

An XML schema defines an XML application structure combined with namespaces, database, and object-oriented design. An XML document is valid if the XML schema and XML application structure meet the specified requirement. As shown in Figure 3, the XML extended application structure is similar to a Document Type Definition. The XML schema data model comprises XML element type (<element> and <name-spaces, xmlns>), XML content model (empty/simple/mixed), and XML data type (basic/derived).

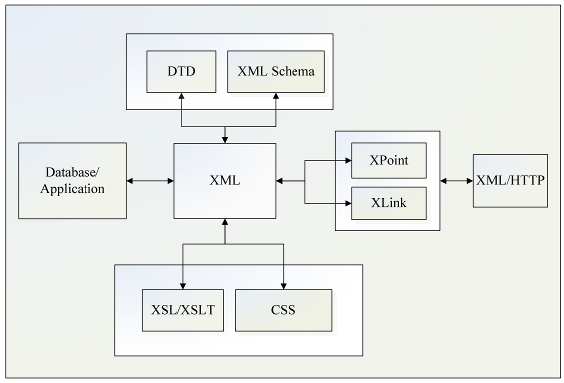


Figure 3. XML extended application structure.

3. System Structure and Design for UMBS

The system structure, system design, and design flow of the UMBS for IP cam networking is discussed in this section.

3.1 System Structure of UMBS

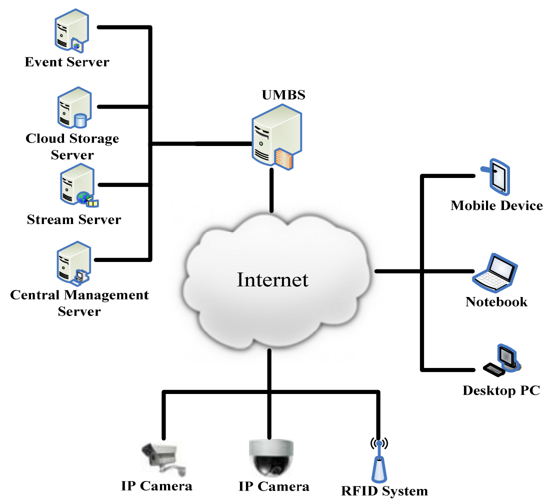


Figure 4. System structure of the UMB.

Figure 4 illustrates the system structure of the proposed UMBS service. A surveillance system equipped with IP cam networking, RFID systems, and Web services can provide a UMBS service through the Internet [11]. The UMBS connects to a server pool comprising a CMS (Central Management Server), SS (Stream Server), CSS (Cloud Storage Server), and ES (Event Server) for monitoring, control, and management processes in an intelligent surveillance system. Clients can operate the intelligent surveillance system integrated with UMBS through smart devices for monitoring and system configuration [12].

3.2 System Design for the Server Pool

The detailed functionalities supported by UMBS service include the provision of a back-end server pool comprising the CMS, SS, CSS, and ES for monitoring and control management as shown in Figure 5.

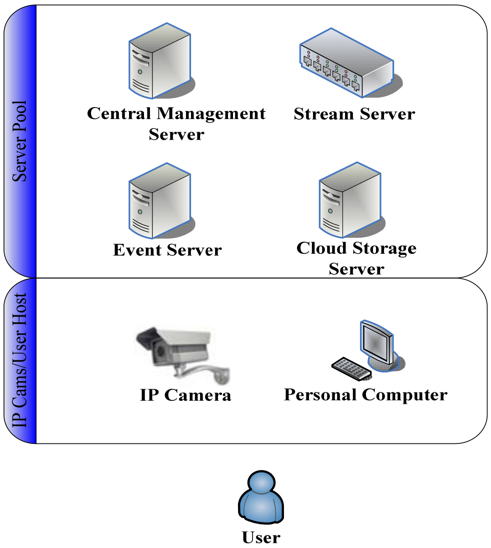


Figure 5. System design for the server pool.

- **Central Management Server**

The CMS dominates the entire servers in the back-end server pool for real-time operation, control management, and firmware update. The CMS also monitors the other servers to prevent them from corrupting the system; furthermore, the CMS provides security to the system as well as a user-friendly HMI (human–machine interface) for easy operation.

- **Stream Server**

Two application modes, namely online mode and on-demand mode, are applied in multimedia streaming. In the online mode, compressed resources are transmitted through the Internet and played in real time by clients. Typical applications are two-way video conferencing and one-way real-time monitoring and control. In the on-demand mode, multimedia resources are stored in the servers of the server pool, and such resources are streamed and played when the SS receives a request. The system structure and design of the proposed UMBS are operated in the on-demand mode.

- **Cloud Storage Server**

In the CSS, a search algorithm is applied to the file searching process for the multimedia streaming. In [10], the researchers presented a simple and fast search algorithm, called binary search tree, capable of providing on-demand multimedia streaming resources. In the CSS, duplicating the resources for backup and expanding the fault-tolerant capacity of the system can be efficiently supported.

- **Event Server**

The ES records the entire status and all event-driven lists triggered in the front-end devices (IP cam). The clients examine if the entire system operates adequately according to the system status stored in the ES. Meanwhile, an alert listener in the crisis notifies the client based on the event-driven list if a trigger occurs in the ES.

3.3 System Design Flow

Figure 6 shows the system design flow of the proposed UMBS. The UMBS serves the server pool, which comprises the mentioned servers for monitoring and control management. Table 1 shows the detailed steps of the design flow.

Table 1. Design flow for the proposed UMBS.

Steps of the design flow for the proposed UMBS
Step 1: System start.
Step 2: Discover and connect to the IP cams.
Step 3: Are IP cams attached to the intelligent UMBS?
Step 4: If Flag=True, exit; else proceed to Step 6.
Step 5: Exit.
Step 6: Set up the desired IP cams.
Step 7: Does an IP cam event trigger exits? If Flag=True, proceed to Step 8 for the event-driven trigger in the ES; If Flag= False, proceed to Step 9 for recording.
Step 8: The event-driven trigger in the ES.
Step 9: Is the recording initiated? If Flag=True, proceed to Step 10 for recording process in the CSS; If Flag= False, proceed to Step 11 for intelligent fine tuning.
Step 10: Start the recording process.
Step 11: Intelligent fine tune? If Flag=True, proceed to Step 12 for intelligent fine tune setting; If Flag= False, proceed to Step 13 for the playback process.
Step 12: Intelligent fine tune setting for IP Cam
Step 13: Streaming playback in the SS? If Flag=True, proceed to Step 14 for Streaming playback in the SS; If Flag= False, proceed to Step 15 for the main menu.
Step 14: Play back the streaming from the SS.
Step 15: Return to the main menu.

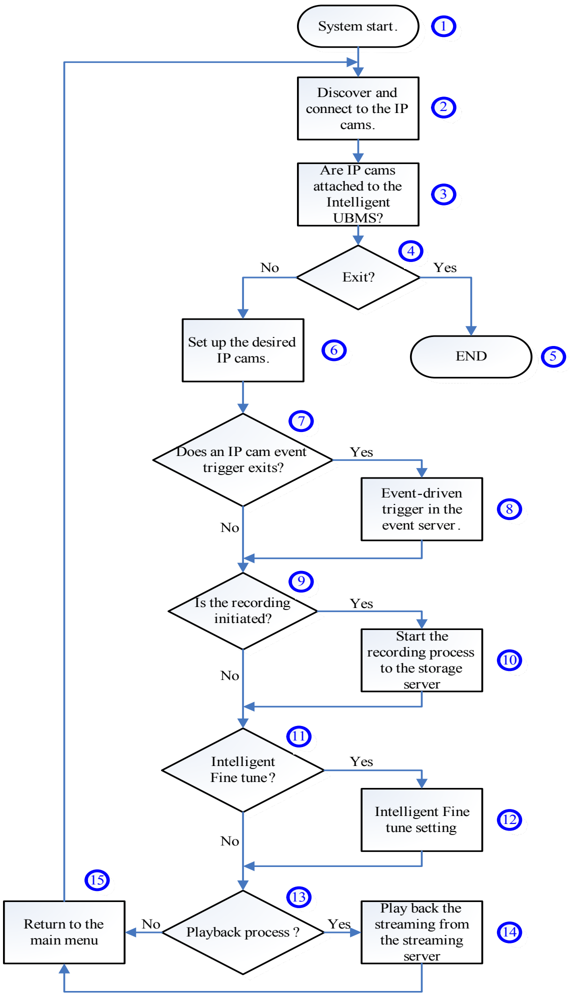


Figure 6. System design flow of the UMBS.

3.4 UMBS Middleware

Figure 7 illustrates the UMBS middleware integrated with IP cams including the ONVIF and PSIA specifications. IP cams embedded the proprietary protocol for AXIS or IPUX are available in the future. The UMBS middleware can execute all instructions and get/set system parameters through device discovery for web service, device information captured for device parameters, playing for video streaming, and intelligent metadata setting of system parameters. Those proprietary protocols can be transferred each other through the UMBS middleware.

4. UMBS Implementation

As mentioned, the UMBS comprises four main functional modules and system UIs, and Table 2 shows these modules and UIs.

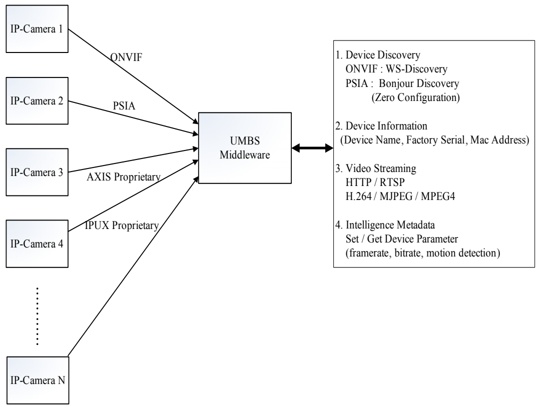


Figure 7. Structure of the UMBS middleware.

Table 2. Four functional modules and system UIs.

LV	- The web service is combined with the UMBS that supplies an LV module with an embedded player. - Remote clients connect to the web service and UMBS for system operations.
PV 	- The UMBS supplies the PV module with an embedded player. - The PV module operates according to the SS and CSS.
iScheduler 	- The UMBS supplies the iScheduler module. - The iScheduler module operates according to the ES and CSS for the intelligent scheduling of event-driven recording.
SC 	- The UMBS supplies the SC module for monitoring the servers. - The SC module provides the user-friendly HMI of IP cams for manual system setting, automatic configuration, and management.

4.1 Live Video Module

UMBS is a web service that supplies an LV module with an embedded player. The video streaming is decoded and played by the embedded player. Remote clients connect to the UMBS for system operations. The SC module provides a user-friendly HMI of the IP cams for manual system setting, automatic configuration, and management based on ONVIF and PSIA. As illustrated in Figure 8, a frame switching mechanism including one partition and four partitions is designed for ensuring fine observation. The detailed status including the IP address, streaming resources, and recording status of the targeted IP cam are also available in the LV module.

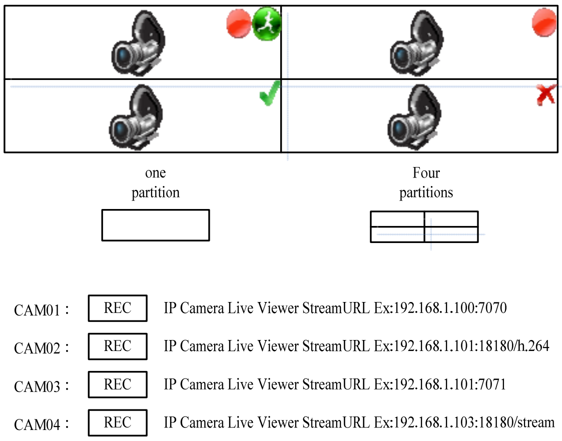


Figure 8. LV module.

4.2 Playback Video Module

As shown in Figure 9, the UMBS supplies the PV module with an embedded player for video playback. The PV module operates based on the SS and CSS. Four camera channels can be assigned and played at any playing speed (1/2X, 1X, 2X, 4X, 8X, 16X, 32X) with the one-partition and four-partition frames at a specified time.

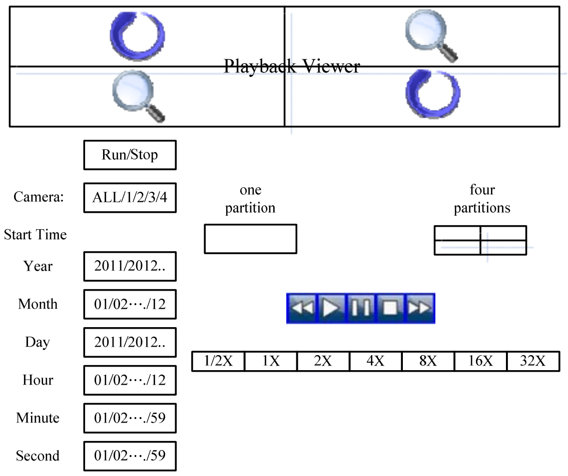


Figure 9. PV module.

4.3 Intelligent Scheduler Module

As illustrated in Figure 10, the UMBS supplies an embedded player to the iScheduler module. The iScheduler module operates according to the ES and CSS for for event-driven recording processes. This module includes three recording modes, namely manual-mode recording, auto-mode recording, and event-driven mode recording, according to the specified time. When an IP cam alert is issued or when an intruder is detected, the event-driven mode is triggered.

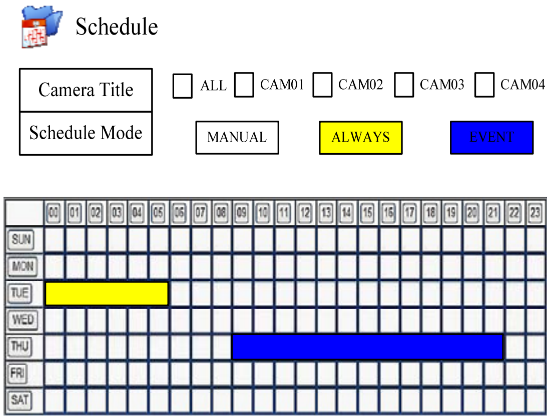


Figure 10. iScheduler module.

4.4 System Configuration Module

As shown in Figure 11, the UMBS provides the SC for monitoring all servers in the pool. The SC module provides a user-friendly HMI of the IP cams for manual system setting, automatic configuration, and management. In the IP cam setting, a group of IP cams can be searched and added using ONVIF or PSIA. The parameters of IP cam, IP address, port, SDP, hue, saturation, and brightness can be set by conducting a manual system setting and automatic configuration based on the system time.

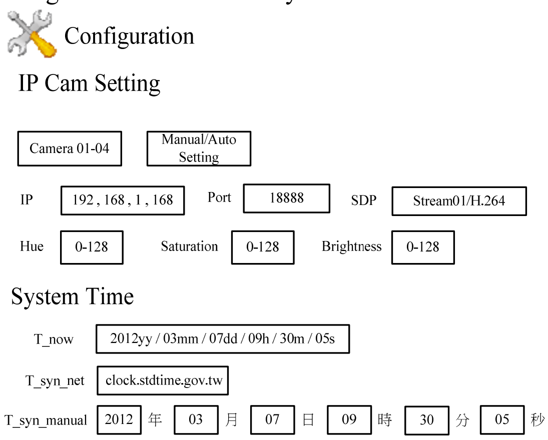


Figure 11. SC module.

5. Proposed Search Algorithm for IISS with UMBS

5.1 Design Flow

The design of the proposed IISS with UMBS algorithm is presented in this section. Figure 12 illustrates the design flow of the IISS algorithm for image/facial recognition and matching.

An object ID database for the proposed search algorithm was established according to the requirement of the IISS with UMBS. Table 3 shows the detailed steps of the IISS algorithm according to the design flow.

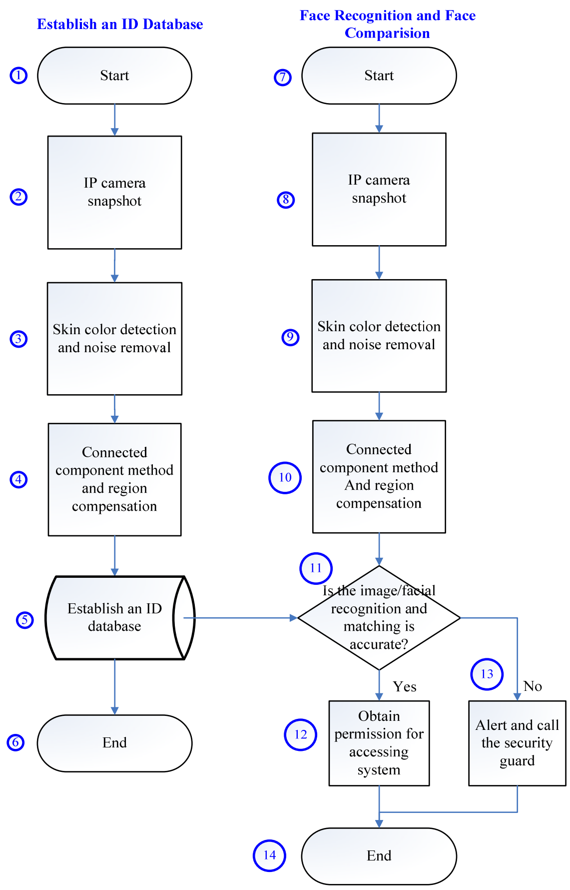


Figure 12. Design flow of the IISS algorithm.

Table 3. Detailed steps of the IISS algorithm.

Algorithm: IISS with UMBS algorithm

Step 1: Start the process of establishing an ID database.
Step 2: IP camera snapshot.
Step 3: Skin color detection and noise removal.
Step 4: Connected component method and region compensation.
Step 5: Establish an ID database.
Step 6: End the process of establishing the ID database.
Step 7: Start the facial recognition and comparison process.
Step 8: IP camera snapshot.
Step 9: Skin color detection and noise removal.
Step 10: Connected component method and region compensation.
Step 11: Is the image/facial recognition and matching is accurate?
If Flag=True, proceed to Step 12 for passing the access system;
If Flag= False, proceed to Step 13 for alerting and calling the security guard.
Step 12: Obtain permission for accessing the system.
Step 13: Alert and call the security guard. Proceed to Step 14 for streaming playback in the SS.
Step 14: End the facial recognition and comparison process.

5.2 Skin Color Detection

In the algorithm, normal color coordinates (NCCs) are adopted to reduce the influence of environmental brightness on the skin color detection [13]. The NCCs are transformed using Eq.(1), where r(x, y), g(x, y), and b(x, y) are the values of the red, green, and blue pixels, respectively, in the NCC space. Furthermore, R(x, y), G(x, y), and B(x, y) are the values of the red, green, and blue pixels, respectively, in the RGB space. The first step is to distinguish a human face from the entire snapshot captured by an IP camera in the surveillance system. Six formula rules (L1, L2, L3, L4, L5 and L6) are applied to skin color detection, as shown in Eqs.(2)–(6).

$$r(x, y) = R(x, y)/(R(x, y)+G(x, y)+B(x, y))$$
$$g(x, y) = G(x, y)/(R(x, y)+G(x, y)+B(x, y)) \quad (1)$$
$$b(x, y) = B(x, y)/(R(x, y)+G(x, y)+B(x, y))$$

As shown in Eqs. (2) and (3), $Boundary_{upper}$ and $Boundary_{lower}$ are the upper and lower boundaries of the distribution of the skin pixel values in the rg plane. When all the R(x,y), G(x,y), and B(x,y) pixel values are set to 255, the pixel values of r(x,y) and g(x,y) are both 0.33 (i.e., 0.33=255/(255+255+255)). It means that the white pixels may partially or fully belong to the desired region. To remove white pixels with values of 0.33 from the desired region, w(x,y) is defined according to Eq. (4) in the rg plane [14].

Figure 13 illustrates the skin color detection results obtained using NCCs in the RGB space according to L1–L5. Figure 13(b) shows the result obtained using NCCs and Eqs. (2) - (4). According to the results, the accuracy of using Rules L1–L3 to filter out nonskin pixels is not adequate, particularly for yellow-green, blue, and orange pixels that are distributed around the top, left side, and right side of the original image, respectively. To improve the results, we introduce two extra rules, namely Rules L4 and L5, as shown in Eqs. (5) and (6). Rule L4 is derived according to the observation that pixels associated with the human skin implicitly tend to be red and yellow. This observation implies that blue pixels always demonstrate the lowest intensity among pixels in the RGB channel for skin color detection. Hence, blue pixels can be effectively removed based on L4. Rule L5 is defined based on Eq. (6) for removing yellow-green pixels [15]. Figure 13(c) illustrates the improved result obtained using the NCCs in the RGB space according to Rules L4 and L5 shown in Eqs. (5) and (6).

L1: $B_{upper}=-1.376r^2(x, y)+1.0743r(x, y)+0.1452,$
$g(x, y)<B_{upper}, \quad (2)$

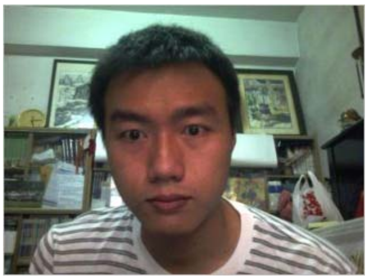
L2: $B_{lower}=-0.776r^2(x, y)+0.5601r(x, y)+0.1766,$
$g(x, y)<B_{upper}, \quad (3)$

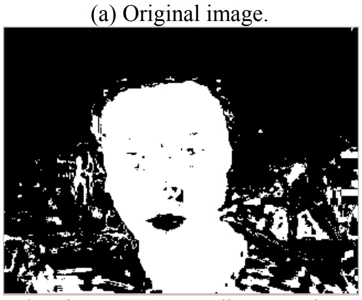
L3: $w(x, y)=[r(x, y)-.33]^2+[g(x, y)-.33]^2>0.0004 \quad (4)$

L4: $R(x, y) > G(x, y) > B(x, y) \quad (5)$

L5: $R(x, y) - G(x, y) \geq \sigma, \sigma=[0...45] \quad (6)$

L6: $S(x, y) = 1, L1 \cap L2 \cap L3 \cap L4 \cap L5$
$0, otherwise \quad (7)$


(a) Original image.


(b) Result using NCCs according to Rules L1–L3.

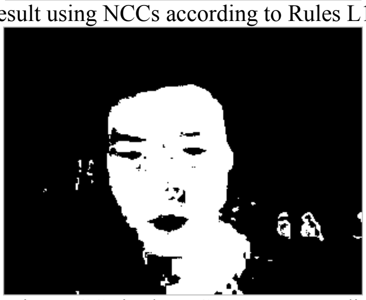
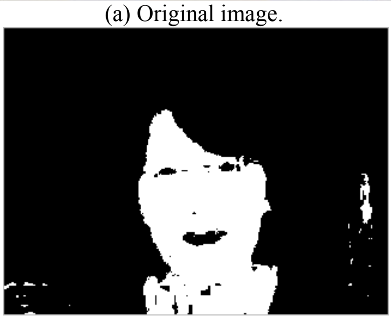

(C) Result using NCCs in the RGB space according to Rules L1–L5.

Figure 13. Skin color detection using NCCs in the RGB space according to Rules L1–L5.

Researchers in a previous study [15] suggested setting the delta coefficient σ in Eq. (6) to 15 or 45. The delta coefficient can be adjusted between 0 and 45 for optimal performance in skin color detection. This is because the delta coefficient is sensitive to the brightness variation in an environment. Rule L6 (Eq. (7)) summarizes Rules L1–L5, and it was employed for obtaining optimal skin color detection results obtained using the NCCs in the RGB space. Figure 14 shows the skin color detection results obtained using the NCCs in the RGB space according to L6. Figure 14(b) and Figure 14(c) depict the results obtained according to L6 by setting σ to 10 and 20, respectively, indicating that the results obtained by setting σ to 20 are more favorable than those obtained by setting σ to 10.

Next, experiments conducted in a YC_bC_r space by using a nonlinear transformation are presented according to the transformation of the YC_bC_r space from the RGB color space [13][14][15] that is widely used in video compression standards (e.g., MPEG and JPEG). The main objective for the transformation of the YC_bC_r space is to achieve a perceptually uniform distribution and to realize optimal separation of luminance and chrominance for skin color detection [16][17].


(a) Original image.


(b) NCC+RGB, binary image, $R(x, y) - G(x, y) \geq \sigma$, $\sigma = 10$.

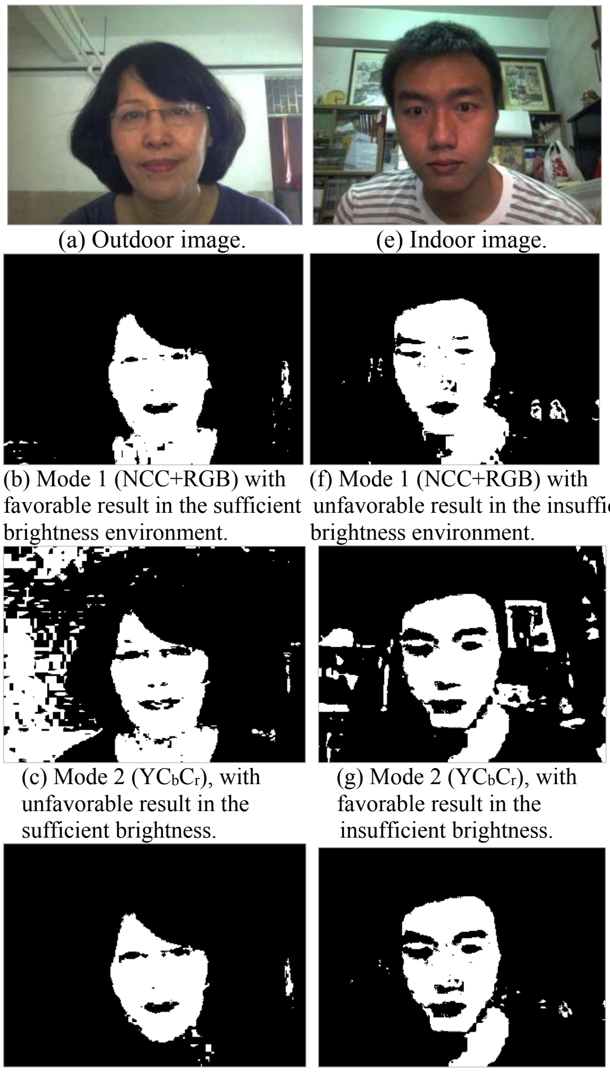

(c) NCC+RGB, binary image. $R(x, y) - G(x, y) \geq \sigma$, $\sigma = 20$.

Figure 14. Skin color detection using NCCs in the RGB space according to L6.

According to the presented results, we propose three modes, namely NCC and RGB mode (Mode 1), YC_bC_r mode (Mode 2), and NCC, RGB, and YC_bC_r mode (Mode 3), to enhance the performance of the image/facial recognition and matching algorithm. The improvement registered after using Mode 3 is obvious (Figure 15).

- **Mode 1 (NCC and RGB):** The NCC and RGB mode demonstrated a more favorable skin color detection performance than that of the YC_bC_r mode in an environment with adequate brightness. Using the YC_bC_r mode to differentiate the skin color and facial features from the background is difficult.

- **Mode 2 (YC_bC_r):** The YC_bC_r mode demonstrated favorable skin color and face feature recognition in an environment with insufficient brightness.
- **Mode 3 (NCC, RGB and YC_bC_r):** In an environment with normal and uniform brightness, the NCC, RGB, and YC_bC_r hybrid mode demonstrated more favorable skin color detection results than Mode 1 and Mode 2 did.



(a) Outdoor image. (e) Indoor image.

(b) Mode 1 (NCC+RGB) with favorable result in the sufficient brightness environment. (f) Mode 1 (NCC+RGB) with unfavorable result in the insufficient brightness environment.

(c) Mode 2 (YC_bC_r), with unfavorable result in the sufficient brightness. (g) Mode 2 (YC_bC_r), with favorable result in the insufficient brightness.

(d) Mode 3(NCC+RGB+YC_bC_r) with more favorable result in the sufficient brightness. (h) Mode 3(NCC+RGB+YC_bC_r) with more favorable result in the insufficient brightness.

Figure 15. Skin color detection results using different modes with varying brightness environments.

5.3 Morphological Operations

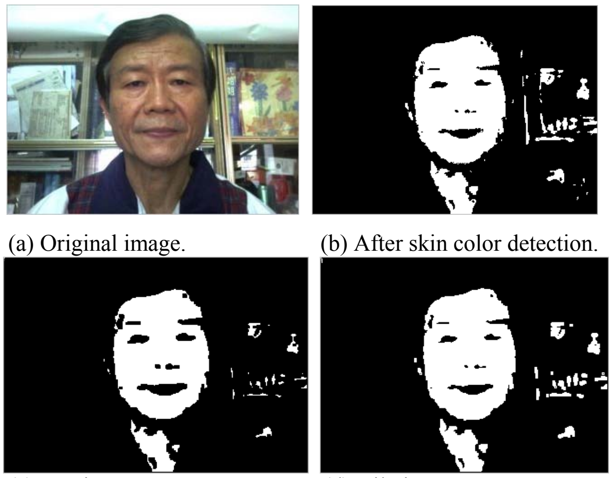
Figure 16 shows the results of morphological operations [18] conducted for removing noisy pixels. Numerous noise-connected pixels still remained after the skin color detection operation (Figure 16(b)), indicating that this operation did not distinguish the desired image/facial region from the background. Hence, we used an opening morphological operation to remove noisy background pixels. The opening operation includes a three-by-three mask is an erosion process; this operation is followed by a dilation process conducted to separate the noise-connected objects.

- **Erosion process:** This process is conducted to determine if the center point M is set based on the surrounding points $[P_1 \ldots P_8]$. If $p_i = 1$, for any i, the center point M is set to one, else it is set to zero. Figure 16(c) shows the erosion result after the skin color detection.

$$M = M \cap P_1 \cap P_2 \cap P_3 \cap P_4 \cap P_5 \cap P_6 \cap P_7 P_8 \quad (8)$$

- **Dilation process:** This process is conducted to determine if the center point M is set based on the surrounding points $[P_1 \ldots P_8]$. If $p_i=1$, for any i, the center point M is set to one, else it is set to zero. Figure 16(d) illustrates the dilation result after the skin color detection.

$$M = M \cup P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5 \cup P_6 \cup P_7 \cup P_8 \quad (9)$$



(a) Original image. (b) After skin color detection.

(c) Erosion process. (d) Dilation process.

Figure 16. Morphological operations for removing noisy pixels.

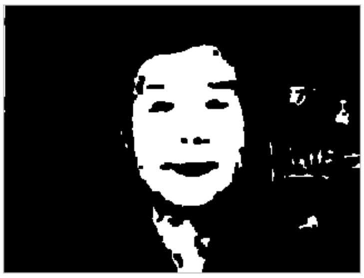
5.4 Connected-Component Labeling

Figure 17(a) indicates that several undesired objects still exist after the morphological operations. To obtain an optimized desired region, we used the connected-component Labeling (CCL) algorithm for determining the maximal skin color in the image/facial recognition and matching process. The principle of the CCL algorithm is to detect connected regions in a source binary image. This algorithm generally involves applying four-connected neighborhoods and eight-connected neighborhoods. For efficiency, we used four-connected neighborhoods [19] to improve the performance. The pseudo code of the 4-connected neighborhood CCL implementation is shown in Table 4. Figure 17(b) depicts the CCL result, indicating that the undesired objects are removed base on the four connected-component labeling.

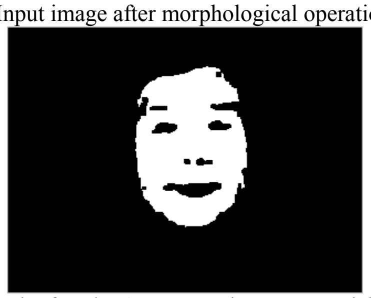
Table 4. Pseudo codes of 4-connected neighborhood CCL implementation.

Algorithm: CCL algorithm

```
Algorithm: Find_MaxObjectID(Pic)
int maxid, id;
new Map[Pic→Width][Pic→Height];
for y in Pic→Height
     for x in Pic→Width
      If Pic[x][y] is White then
              int counts = CCL(Pic, x, y);
                If counts > maxcounts then
                set maxcounts to counts;
                set maxid to id;
             id++;
delete Map;
return maxid;
recursion CCL(Pic, x, y)
   int count=0;
   count++;
   Set HandlePixel Pic[x][y] to Black;
   Set ConnectedComponentLabelMap[x][y] to id;
    If RightNeighbor [x+1][y] is White
        Then count+=CCL(Pic, x+1, y);
    If DownNeighbor [x][y+1] is White
        Then count+=CCL(Pic, x, y+1);
              If LeftNeighbor [x-1][y] is White
        Then count+=CCL(Pic, x-1, y);
    If UpNeighbor [x][y-1] is White
        Then count+=CCL(Pic, x, y-1);
return count;
```



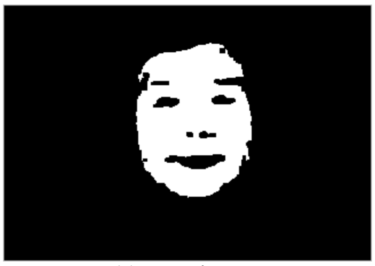
(a) Input image after morphological operations.



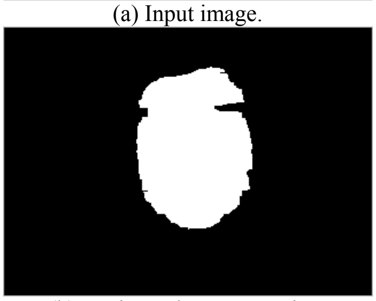
(b) Result after the 4-connected component labeling.

Figure 17. Results after the four-neighborhood CCL.

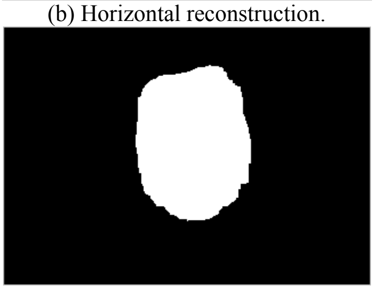
5.5 Region Reconstruction



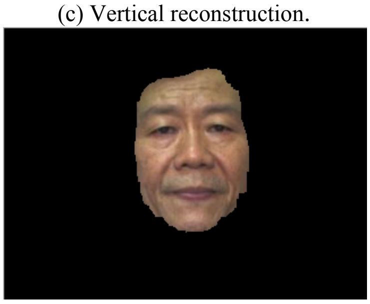
(a) Input image.



(b) Horizontal reconstruction.



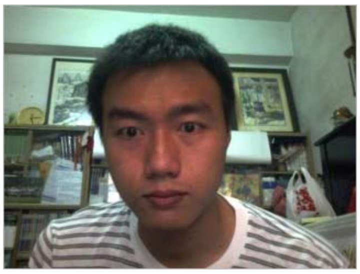
(c) Vertical reconstruction.



(d) The resultant image after reconstruction.

Figure 18. Final image after region reconstruction.

Reconstruction the desired region of the main object (Figure 18 (a)) is mandatory after skin color detection, morphological operations, and CCL. The main object includes the eyes, mouth, nostrils, and the other non-skin regions. The desired region must be reconstructed to eliminate the non-skin regions (black region). The first process involves determining the range of the minimal and maximal coordinates for each horizontal axis from the top side to the bottom side, and then reconstructing the image/facial region (Figure 18 (b)). The second process involves repeating the first procedure for each vertical axis from the left side to the right side, and then reconstructing the object region (Figure 18 (c)). The final process involves conducting the mapping from the object region of the skin color pixels to the original source image. Figure 18(d) illustrates the final image after the reconstruction process.



Input image.

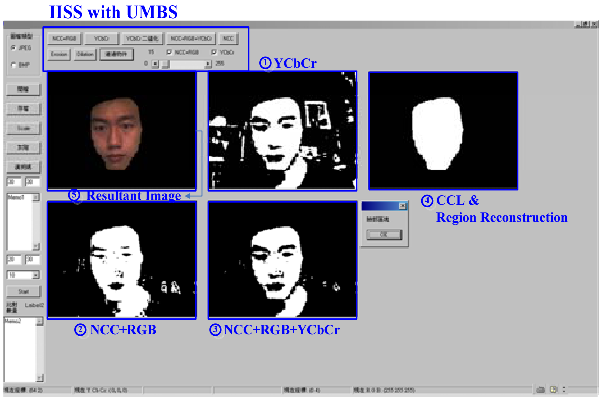


Figure 19. Implementation of image-based intelligent surveillance system with UMBS.

Figure 19 illustrates an implementation of the IISS with the detailed procedures conducted according to the design flow. The entire IISS with UMBS is realized based on the three modes in the color space, CCL algorithm, and region reconstruction.

5.6 Image/Facial Recognition and Matching Algorithm

The perceptual hash algorithm [20] is widely used to search for desired objects and images. This algorithm returns particular codes (called fingerprints) for each object. The fingerprint represents a hamming distance code and corresponds to the target object. To search for a desired object among various objects in a data set, the objects in the data set can be compared according to their respective hamming distance codes. Figure 20 shows the design flow of the face/object matching algorithm. Figure 21 illustrates the matching results of the proposed IISS with UMBS for face/object matching conducted according to the corresponding steps in the design flow.

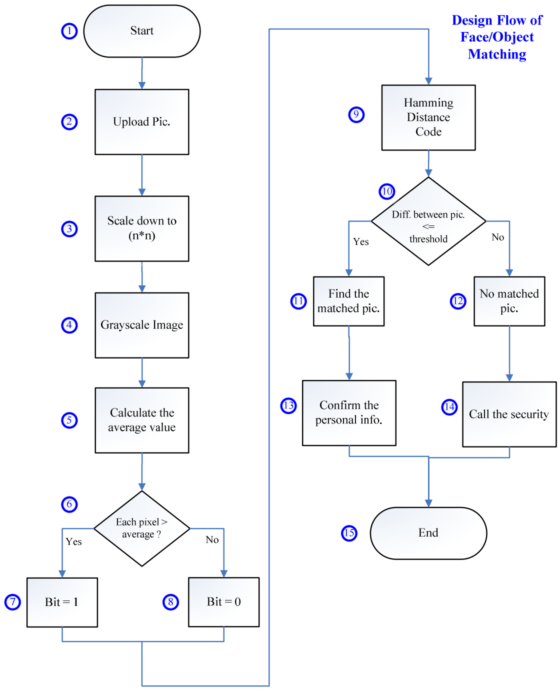


Figure 20. Design flow of the face/object matching system algorithm.

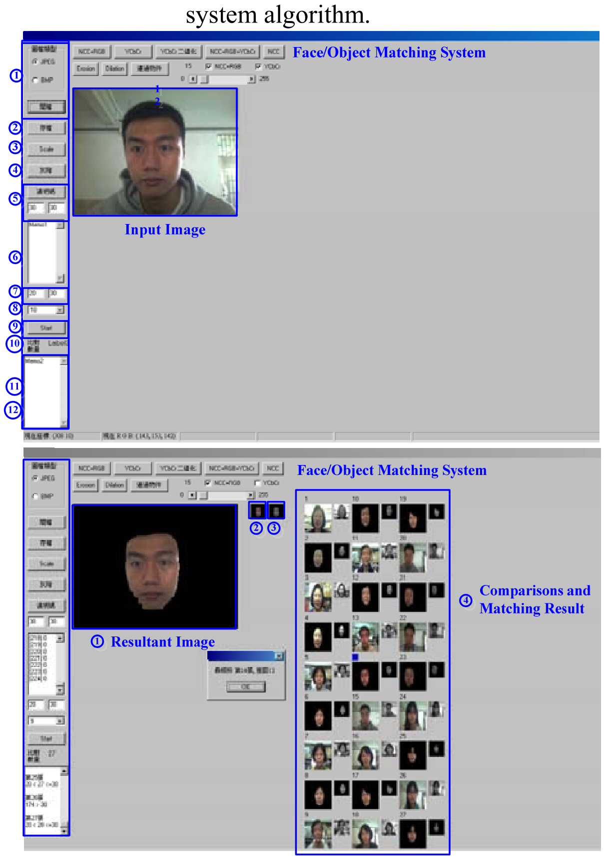


Figure 21. Matching results of the proposed IISS with UMBS for face/object matching.

6. Conclusions

In this study, an efficient implementation an IISS integrated with a robust UMBS was realized. A face/object recognition and matching service was also implemented for the IISS with UMBS. The UMBS for IP cam networking based on the ONVIF serves as the front-end/rear-end integrated protocol applied in the proposed system. The UMBS provides related mechanisms for manual system setting, automatic configuration, and management for improving the entire setting and installation procedures. The structure of the robust UMBS ensures adaptability and flexibility in designing and developing IP cam application systems. An intelligent home automation system integrated with the robust UMBS has potential applicability in the future. Our future research interests include combining framework and application services with cloud computing. The extension of the proposed UMBS for IP cam networking to mobile platforms combined with the principle of object-oriented design pattern is also expected in future research.

Corresponding Author:
Dr. Chia-Hsu Kuo,
Department of Software Engineering,
National Kaohsiung Normal University,
Kaohsiung 824, Taiwan
E-mail: kuoch@nknu.edu.tw

References

1. Physical Security Interoperability Alliance, PSIA Common Metadata/Event Management Specification Ver. 1.2 Rev. 0.4, January 2012.
2. Open Network Video Interface Forum. ONVIF Core Specification Ver. 2.2, May 2012.
3. http://www.onvif.org/
4. Y. Y. Wu, "Searching Mechanism Design and Verification on Open surveillance Image Manage System", July 2011.
5. Y. H. Tsai, J. K. Hsu, Y. E. Wu, and W. F. Huang, "Distributed Multimedia Content Processing in ONVIF Surveillance System," 2011 International Conference on Future Computer Sciences and Application, 2011:70-73.
6. Chia-Hsu Kuo, Huan-Ming Hsu, Shu-Chun Ho, Wen-Tin Lee, "Universal Middleware Bridge System for IP cam networking," 2013 IEEE International Symposium on Next-Generation Electronics (ISNE 2013), Kaohsiung, Taiwan, , Feb., 2013:291-295.
7. http://www.w3school.com.cn/wsdl/index.asp
8. http://www.w3schools.com/soap/default.asp
9. http://www.w3school.com.cn/schema/index.asp
10. Yanlong Fu, Junming Xu, Zhonghe Chen, "The Design of Embedded Digital Video Recorder Network Storage System Based on FTP," 2011 International Conference on Electrical and Control Engineering, 2011:3710-3713.
11. Kyeong-Deok Moon, Young-Hee Lee, and Chang-Eun Lee, Young-Sung Son, "Design of a Universal Middleware Bridge for Device Interoperability in Heterogeneous Home Network Middleware," IEEE Transactions on Consumer Electronics, 2005;51(1):314-318.
12. N. Boonma, A. Sangthong, S. Mitatha and C. Vongchumyen, "Image Recorder Server with IP Camera and Pocket PC," Procedia Engineering, 2011;8:182–185.
13. M. Soriano, S. Huovinen, B. Martinkauppi, M. Laaksonen, "Using the skin locus to cope with changing illumination conditions in color-based face tracking," IEEE Nordic Signal Processing, Symposium, Kolmarden, Sweden, 2000:383-386.
14. M. Soriano, B. Martinkauppi, S. Huovinen, M. Laaksonen, "Adaptive skin color modeling using the skin locus for selecting training pixels," Pattern Recognition 2003;36(3):681-690.
15. Cheng-Chin Chiang, Wen-Kai Tai, Mau-Tsuen Yang, Yi-Ting Huang, Chi-Jaung Huang, "A novel method for detecting lips, eyes and faces in real time," Real-Time Imaging, 2003;9(4):277-287.
16. Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Anil K. Jain, "Face Detection in Color Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002;24(5):696-706.
17. Stan Z. Li and Anil K. Jain, Handbook of Face Recognition, 2004.
18. C. G. Rafael and E. W. Richard, Digital Image Processing 2nd Edition, 2002.
19. B. R. Kiran, K. R. Ramakrishnan, Y. S. Kumar, and K. P. Anoop, "An Improved Connected Component Labeling by Recursive Label Propagation," 2011 National Conference on Communications, 2011.
20. Zhen-Kun Wen, Jie Ouyang, Peng-Fei Liu, Yi-Hua Du, Meng Zhang, Jin-Hua Gao, "A Robust and Discriminative Image Perceptual Hash Algorithm," 2010 4th International Conference on Genetic and Evolutionary Computing, 2010: 709-712.

08/05/2015