International Journal of Innovative Computing, Information and Control
Volume **9**, Number **3**, March **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 1293–1320

THERMAL IMAGING BASED OFF-TIME SWIMMING POOL SURVEILLANCE SYSTEM

Wai Kit Wong[1], Joe How Hui[1], Chu Kiong Loo[2] and Way Soong Lim[1]

[1]Faculty of Engineering and Technology
Multimedia University
Jln Ayer Keroh Lama, Melaka 75450, Malaysia
{ wkwong; wslim }@mmu.edu.my; huijoehow@gmail.com

[2]Faculty of Computer Science and Information Technology
University of Malaya
Lembah Pantai, Kuala Lumpur 50603, Malaysia
ckloo.um@um.edu.my

Received January 2012; revised June 2012

Abstract. *Swimming pool surveillance system plays an essential role in safeguarding the premises. In order to deploy the monitoring system, human detection is required coupled with several image processing techniques. In this paper, an off-time swimming pool surveillance using thermal imaging system is proposed to enhance the security of the surrounding area. Two sub-algorithms are developed to detect any human intrusion into the region outside and inside the swimming pool. Animals and humans outside the swimming pool are able to be discovered and differentiated by the surveillance system. Meanwhile, the algorithm is capable of detecting not only human movements in different directions, but also water activities in the swimming pool, such as swimming, water-splashing, and others. Human height and width are the main criteria in monitoring the system outside the pool, whereas the important characteristic used to observe for any intruder inside the pool is the dimension of human head shape. The water activity detection utilises the difference of black colour pixels between successive thermal images. Based on the experimental findings, the intruder detection algorithm has an accuracy of 95.58% for region outside the pool, and 92.44% inside the pool.*

Keywords: Image processing, Human-machine learning, Off-time surveillance system, Knowledge-based systems, Swimming pool surveillance system, Thermal imaging

1. Introduction.

This section contains problem statements, motivation, objectives and project outlines.

1.1. Problem statement.

Child safety has been a major concern issue in a modern society. The inquisitive nature of children has driven them to explore new and foreign areas, more specifically, their surroundings. To make matters worse, children's agile movements increase the risk of drowning accidents when a swimming pool is available in houses, apartments, or hotels [1]. Recent statistical results reveal that drowning is the second-major cause of unintentional injury for children [2]. Furthermore, a study conducted by the Life Saving Society Malaysia (LSSM) discloses that approximately 50% of the drowning cases victims are children aged 18 years and below [3]. Obviously, this shows the necessity of installing a monitoring device on swimming pools for surveillance purposes.

1.2. **Motivation.** Usually, manpower is needed to observe the surveillance monitor. However, watching the surveillance equipment like closed circuit television (CCTV) is very tedious, due to human nature, which is, decreasing attention caused by fatigue [4]. Such conventional system is inclined to errors because human exhaustion is inevitable. Therefore, the need for an off-time swimming pool surveillance system arises.

There are several types of surveillance systems in the market. Yet, one major problem regarding surveillance systems is the change in ambient light. This occurs more often in an outdoor environment, where the lighting condition varies naturally [5]. The environment occasionally can be completely dark, making surveillance systems to be more challenging. As a result, a conventional digital colour image analysis is very difficult in a smart surveillance. Hence, it is not suitable for detection of children or intruders in the dark at swimming pools.

Following this, this paper seeks to explore the use of thermal camera as an off-time swimming pool surveillance system. The decision to opt for thermal camera is highly attributed to the ability of the camera in providing good photo images at all times, especially during the night, and when the lighting condition is poor [6]. Several image processing techniques are applied into the algorithms development to produce a reliable, robust surveillance system, which is capable to detect intruders.

1.3. **Objectives.** Both human detection method and thermal imaging system are the two main themes in this project, in which the objectives are:

1) To design a surveillance system to detect intruders in the swimming pool,
2) To implement the algorithm using thermal imaging system.

The outcome of this project is an off-time smart swimming pool surveillance system using image processing technique.

1.4. **Project outlines.** This thesis encompasses nine chapters. Section 1 introduces the problem statements as well as the project motivation and its objectives. Meanwhile, a critical literature review is presented in Section 2. Reviews on several previous works done are shown in Section 3. Subsequently, Section 4 explains the system model and Section 5 elaborates the proposed algorithm in-depth. These chapters are followed by the experimental results in Sections 6 and 7 together with optimisation of the setup parameters. The overall accuracy and performance comparison are highlighted in Section 8. Lastly, the paper ends with a conclusion and future work discussion in Section 9.

2. **Critical Literature Review.**

Currently, there are various methods to categorise human surveillance monitoring system. Active motion sensors and passive motion sensors are part of the categorising approach. The former sensors operate by transmitting optics or sound waves and measuring the feedback signal for motion detection. One application which employs active motion sensor is microwave motion detector. On the contrary, the passive motion detectors do not emit signals. Instead, it measures the changes in temperature of the field of view in the infrared spectrum. A good example is the passive infrared motion detectors [7]. Additionally, another approach to classify the swimming pool surveillance system is contact and non-contact sensors. Contact sensors alarm involves physical contact with the water surface. An example for this category is the ultrasonic pool alarm and surface wave sensors. Meanwhile, the non-contact sensor camera does not involve physical contact with the water surface. Several types of sensors under this classification include passive infrared sensor (PIR), thermal camera, and microwave motion sensors.

In this paper, it is more appropriate to explain each of the surveillance systems in terms of contact and non-contact sensors. The few main types of contact sensors discussed later are ultrasonic motion sensors, surface wave sensors, and underwater camera. Firstly, ultrasonic motion sensors are generally active type sensors. A transducer in the system produces an ultrasonic sound wave, which is approximately 20000Hz [8] through the conversion of electrical energy to sound. This sound wave is emitted from the sensors. It strikes objects within the detection range and returns to the sensors as a feedback echo. As long as there is no movement or object, the wave is undisturbed. Nevertheless, when the wave encounters an object, it is reflected back to the sensor, which interprets the echo. From the echo signal analysis, the sensor determines the existence of an intruder and calculates the distance to the object by calculating the time interval between the transmitted sound wave and the received echo. Then, it relays the signal to the alarm component, which triggers the alarm. Although ultrasonic sensors are proven to be among the most sensitive surveillance system and have the capability to work in high-glare environments, it has several notable drawbacks. The high sensitivity of the ultrasonic sensor can cause false alarm. Innocuous objects, such as insects and birds nearby the swimming pool, may trigger the unnecessary alarm. Furthermore, winds or vibrations from nearby moving objects like vehicles, might also lead to unwanted alarm triggering [9]. This could cause undesired hassle for the users to check the swimming pool and reset the alarm once the alarm is detected as faulty. Another point to take note is the ultrasonic sensor has its own dead zone; where the sensors cannot effectively take the measurement. This issue is highly attributed to the ringing phenomenon occurred in the transducer. Therefore, strategic location in the swimming pool is essential to place the ultrasonic sensor in order to avoid the dead zone [10]. A better version of ultrasonic sensor is available to overcome such weakness stated above. Produced by the Kongsberg Maritime, the SM2000 is an advanced underwater surveillance system based on ultrasonic [11]. Unfortunately, a lot of paperwork and legal procedures are needed to be used by civilian as such the device is more applicable for military purposes.

Since the ultrasonic sensors give various disadvantages and inconveniences, underwater camera is suggested to handle the surveillance system in the swimming pool. It outweighs the ultrasonic sensors in several aspects. There is no dead zone in the underwater camera system. Poseidon safety system, made by Vision IQ, a French company, is equipped with overhead cameras and underwater cameras to monitor any possible intruders in the swimming pool [12]. The overhead cameras monitor the shallow water no deeper than 9 feet, while underwater cameras monitoring system works for water depth of more than 9 feet. Both cameras complement each other to ensure no blind spot in the underwater camera system. A central processor, LED panels also come together in the underwater system as auxiliary tools. False alarm is unlikely to occur in this underwater system as the database in the central processor compares the texture of underwater object, water movement, and the wave trajectory with the captured result before it confirms the foreign object in the swimming pool is a human and rings the alarm. This feature has further enhanced the underwater camera system, which makes it a more outstanding surveillance system than the ultrasonic system. Nonetheless, despite of its advantages over the ultrasonic sensor, the underwater camera system has its disadvantages as well. More space is required to place the central processor, LED panel, and workstation near the swimming pool. This may pose some difficulties to the owner if the swimming pool is located in the condominium. Poor lighting condition also serves as another major problem to the underwater camera system. The performance of the underwater camera may be affected in the evening or night, where there is no sunlight. Some may install several spotlights near the swimming pool to resolve this issue. However, these spotlights require additional space to be installed. The swimming pool area could possibly be congested with surveillance tools and the spotlights especially if they are located in small areas like some apartments in Singapore.

The high sensitivity in the ultrasonic system which may trigger false alarm, as well as the unsatisfactory surveillance performance in poor lighting condition in the underwater system, exemplify that the contact-sensor system is not suitable for the swimming pool surveillance system. Hence, the non-contact system is proposed to counter the weak points in the contact sensor system. One of them is the night vision camera for swimming pool surveillance. As the name implies, this camera can perform well in dark environments or when the sunlight is absent in the evening or night. Image intensifier is the central component to drive the night vision camera operation. It amplifies the available light with the help of an objective lens which focuses light (photons) on the photocathode. Huge amount of electrons are released from the cathode due to the light energy. The internal specially-coated walls increase the amount of electrons when they bounce. Thus, an intensified version of original image is produced by the high density of electrons [13]. This characteristic allows swimming pool surveillance activities possible at night, unlike the underwater camera system. However, such vision camera cannot work optimally during the day [13,14]. This is because the image is intensified unnecessarily; where the image captured without the night vision camera itself is sufficiently bright, rendering the choice of night vision camera as incompatible for the swimming pool surveillance system.

Passive Infrared (PIR) Sensor is another widely used surveillance tool nowadays. It uses the heat emitted by the living objects for motion detection. Unlike the ultrasonic sensors, no wave signals are emitted. Once the heat emission is captured by a mirror or lens, it is transferred to a solid-state chip to record temperature changes. The "hot spot" on the chip moves when the subject does so [15]. Any changes in the reading will trigger the alarm. False alarm due to wind and vibration does not occur as both causes do not transmit any heat energy. In addition, PIR sensor is able to work under any day and night condition. This further counters the underwater camera system and the night vision camera. Moreover, the small PIR sensor chip makes it easier to install, and cheaper when it is mass produced. Unfortunately, this PIR sensor comes with several setbacks. Strategic location at the swimming pool is needed to place this sensor to prevent false alarm due to intruding pets. Also, location which has excessive amount of heat or cold should be avoided to install the sensor. Direct sunlight direction also needs to be averted as it may affect the PIR sensors operation. In addition, the sensors should not be positioned near the heated water pump machine at the swimming pool as it could trigger unwanted false alarm. Intelligent children can bypass the PIR sensor detection range by studying the field-detection pattern of the PIR sensors [15].

As there are problems raised in using PIR sensor, microwave motion detector can be an alternative for the surveillance system. Its sensor emits microwave energy produced by a Gunn diode to its detection area [16]. Any movement can cause disruption in the detection field to activate the alarm system. The microwave motion detector is divided into monostatic and bistatic. The former has the emitter and the receiver placed in one unit and the detection range is up to 400 linear feet, whereas the latter detection range is extended to 1500 linear feet. The users cannot precisely define the detection area; hence it has high tendencies for false alarm. The microwave beams can propagate through majority of surface materials, except metal [16]. Here comes the main disadvantage. For instance, children can place metal objects in the microwave motion detection area to create dead zone – where no motion can be detected. The high sensitivity of the microwave motion detector is also prone to false alarm due to winds, unlike the PIR sensors. However, some experts recommend pairing the PIR sensors with the microwave sensors. This is because the PIR sensors are more sensitive to lateral motion, while the microwave motion detectors are good at detecting forward motion. Again, the probability of the alarm being triggered is reduced significantly when users activate both sensors together via programming [16]. This becomes the main drawback of employing the microwave motion sensors and the PIR sensors.

In contrast, thermal imaging system provides a great solution for off-time swimming pool surveillance. Prior to the development of the thermal imaging system, one major concern encountered by most surveillance systems is the change in ambient light, be it during the day or at night. This problem is further worsened when surveillance takes place at outdoor settings, where the lightning condition varies naturally [17]. Since most swimming pools are located outdoor, a conventional digital colour image analysis may not be able to perform surveillance task efficiently. Consequently, thermal imaging system is introduced to alleviate such issue. The thermal imaging is independent of ambient light. Hence, this provides an advantage for thermal camera to function well when there is absence of light, as compared to the night vision camera which requires minimal light to amplify an image. Any object which radiates infrared energy as a function of their temperature is detected by the thermal camera. Apart from that, its light-independent characteristic enables the thermal imaging system to operate well in weather-hazard environment such as smoke, fog, and haze [18]. For this reason, countries like Russia can employ such surveillance system to guard the swimming pool.

Recently, thermal imaging technology is made viable to many commercial applications; one of them is moving human detection [19]. This paper suggests using thermal camera as off-time swimming pool surveillance. A simple, effective, and fast intruder detection algorithm will be developed and embedded into a cost effective thermal imaging system for detecting intruders at swimming pools during off-time in poor lighting condition. The aforementioned algorithm is coupled with image processing tools to detect intruders effectively, providing highest reliability to the thermal imaging system.

3. **Reviews on Previous Works Done.**

Prior to the development of the human detection strategy for an off-time swimming surveillance system, reviews have been conducted on several previous works. Three methods related to thermal imaging system have been chosen and are discussed in the following sections.

Image segmentation on the partition region of interest (ROI) is introduced in Section 3.1. This technique has an inherent disadvantage, which is the near far problem. Due to that, head detection algorithm is preferred instead to provide a robust performance. The next section describes an effective trespasser detection system based on the human head geometry. Lastly, Section 3.3 introduces human height and width statistical search for a home alone faint detection surveillance system.

3.1. **An effective surveillance system using thermal camera.**

The method used in [20] to detect human is image segmentation. A thermal image is acquired and divided into $(m \times n)$ regions, in which each has the same number of pixels. A matrix, M with size of $(m \times n)$ is created. A current image is compared with a previous image. The sum of the R, G, and B values of every pixel is calculated for each respective image. Later, the sum difference between both images are found, and checked with a threshold value, Q. If Q value is exceeded, the h counter is incremented. Then, if h value is more than a certain predefined parameter H, the corresponding element in M is marked as "1", else it is marked "0". Lastly, a vector, whose size is same with the number of column (n) of the M matrix, is created. Number of continuous "1" in a column is counted and recorded in

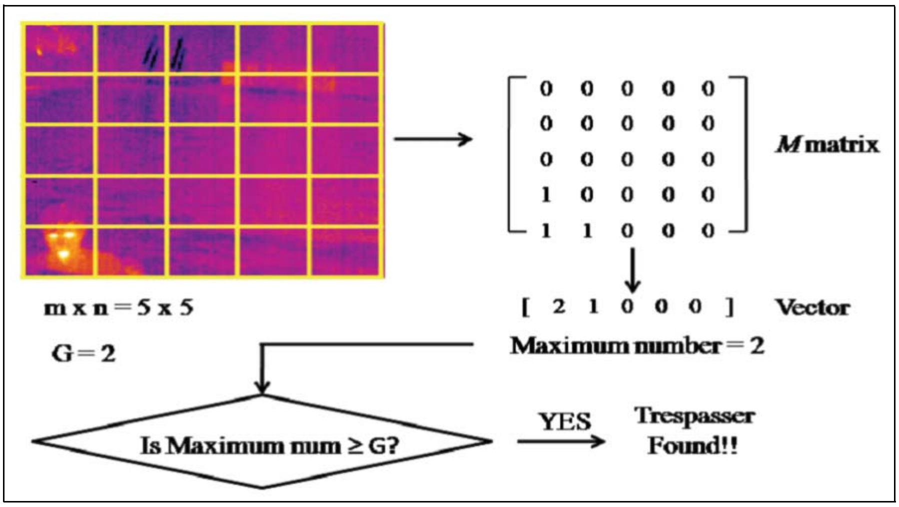


FIGURE 1. Overall concept used by [20] in the trespasser detection system

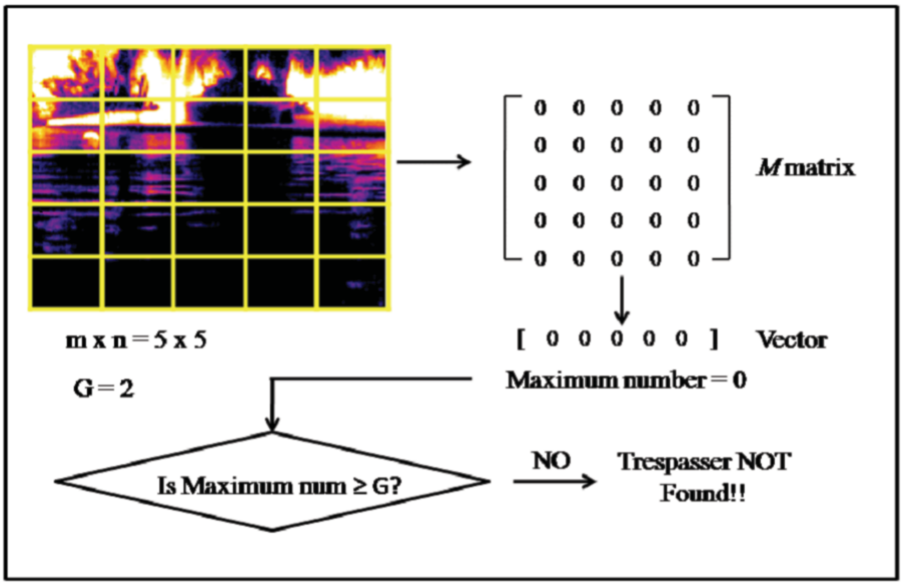


FIGURE 2. Overall concept used by [20] in the trespasser detection system

that vector. If the maximum value in that vector surpasses another threshold value G, the alarm is triggered. Refer to the following diagram for a better illustration.

Nevertheless, the methodology used above has its inherent weak point. Range could pose a problem to the system when a thermal object is located far from the thermography camera. When the distance between the thermal object and the camera increases, the object size reduces from the view point of the camera. Hence, false detection can occur when the system cannot identify that thermal object from a far distance.

Figure 2 shows the possible false detection, in which a thermal object cannot be detected by the surveillance system due to the near far problem. As a result, such an approach is not preferred in this project. Instead, human detection based on the head shape geometry is favoured as it is more robust. A detailed performance comparison between the proposed method and the method used in [20] is discussed in Section 8.

3.2. **An effective surveillance system using thermal camera.** Human shape geometry is the main idea of detecting a trespasser in a surveillance system [21]. Prior to the checking of the head curve, the image is filtered to remove the background noise. Each object boundary is extracted from the image to perform boundary line splitting. Consequently, five main points are identified on each extracted object boundary. These points are the head top point (P_{HT}), first leftmost point (Pl_p), rightmost point right (Pl_d), first rightmost point (Pr_p) and leftmost point left (Pr_p). Then, neck-body position test takes place to evaluate the symmetry of the object. Last but not least, three different curve tests for the top, left, and right section of the head are conducted to verify the existence of a trespasser.

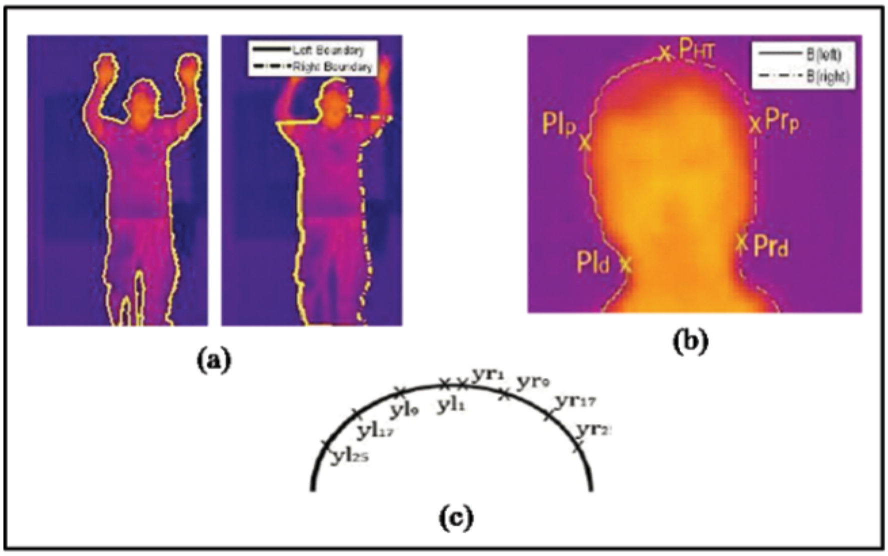


FIGURE 3. (a) Boundary line splitting, (b) five main points detected on a head boundary and (c) top curve test [21]

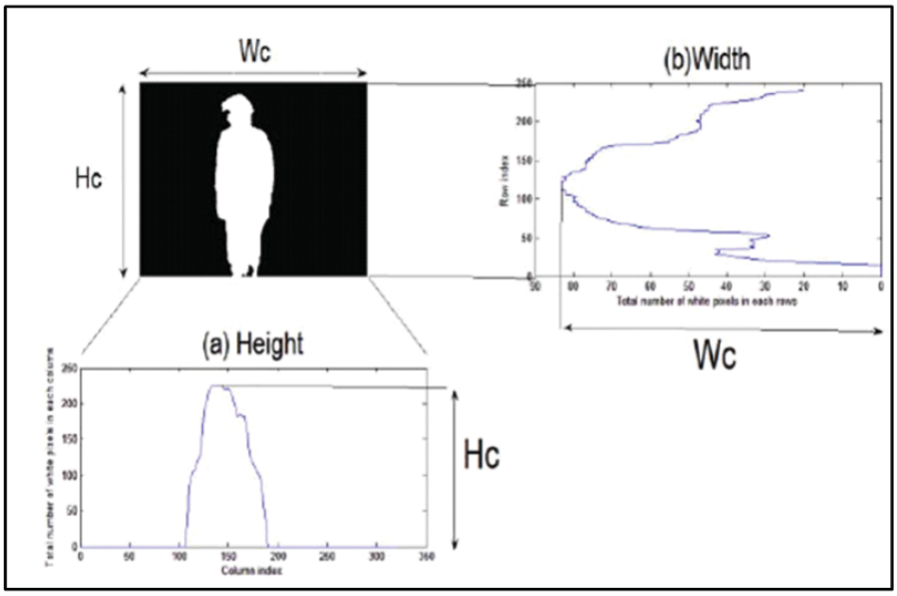


FIGURE 4. Human's height and width statistical searches [22]

Several modifications are done on the aforementioned human detection method before it is applied in this project. This algorithm depends on the human shape geometry to check for human's existence in the swimming pool region. Section 5 provides detailed information on how the algorithm operates.

3.3. **Home alone faint detection surveillance system using thermal camera.** This paper introduces the comparison between a human's height and width method to determine whether that person has fainted [22]. As usual, an image is obtained and filters are applied to remove the background noise. Morphological process is implemented for image enhancement purpose. Statistical searches are then employed to find the human's dimensions, which are height and width. Both measurements are compared for faint detection.

The methodology in [22] is useful for detecting any human and animal outside the swimming pool. Statistical search for each thermal object's height is performed to distinguish humans from animals. The algorithm in [22] is altered and adapted to suit the requirement in this thesis for surveillance system outside the swimming pool.

4. **Off-Time Swimming Pool Surveillance System Model.** An overall system model for the off-time swimming pool surveillance is presented in this chapter. It comprises a thermal camera, coupled with a computer or laptop installed with MATLAB programming (version R2009b or later) and an alarm. Figure 5 describes the system.

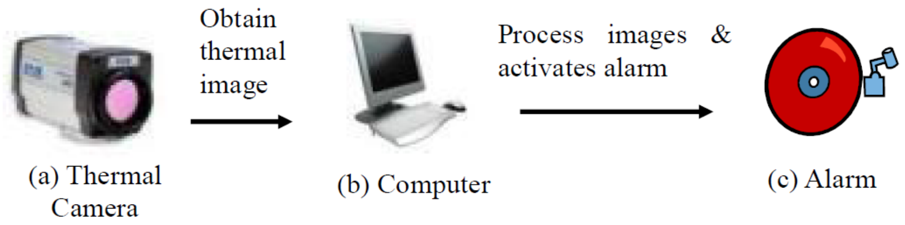


FIGURE 5. System model for off time surveillance system using thermal camera [23]

4.1. **Thermal camera.** ThermoVision$^{\text{TM}}$ A20M is the thermal camera used in this project. Produced by the FLIR manufacturer, the thermal camera operates based on the un-cooled infrared detector mechanism and it does not require extra vacuum case to cool down, unlike the cooled infrared detector, which is bulky. Moreover, the A20M camera is cheaper due to its compact and small size. Thus, it is very ideal for a surveillance system, as it can be concealed at a strategic location during installation.

Another factor, which leads to that choice of thermal camera in this project, is the A20M's superb resolution. Having a resolution of 320×240 pixels, it can capture human objects as far as 18 meters. In addition to that, this camera has a refresh rate of 50/60 Hz and the detectable temperature range is from -20°C to 250°C. In terms of connectivity, the IEEE-1394 (Firewire) and Ethernet options are available. Live streaming of thermal images can be implemented with the IEEE-1394, meanwhile the Ethernet connectivity is ideal for linking multiple cameras in a network. Such connection system is suitable for observing various swimming pools in a region. In this project, the IEEE-1394 connection is used to connect the thermal camera to a laptop.

4.2. **Laptop and alarm.** A laptop or a personal computer is used to perform image processing on the captured images. In this project, the laptop deployed in this project is produced by the Dell manufacturer. Operated by a 2.20GHz Intel Core Duo 2 processor T6600, it has a RAM of 4GB.

For image processing platform, MATLAB programming software version R2009b is chosen. Various built-in image processing functions are available in its Image Processing Toolbox, making the MATLAB software more user-friendly.

In order to raise the attention of the operator, the computer speaker is used as the alarm system. In the event of intruder detection, the system is triggered and a siren sounded via the speaker to inform the operator.

5. **Off-Time Swimming Pool Surveillance Algorithm.** This section describes the proposed surveillance system algorithm for region outside and inside the swimming pool. The region inside the swimming pool is labelled as Region 1, whereas, the region outside the swimming pool is labelled as Region 2. Region 1 algorithm operates based on two systems – The human head detection, which is adapted and modified from Chew's work [21], Lim's work [22], and the water activity detection. On the other hand, Region 2 algorithm consists of background patching and the identification of human and animal based on the width and the height of a human.

5.1. **Algorithm for surveillance system.**

Step 1: Acquire the thermal image from the thermal camera. Refer to Figure 6(b).

Step 2: Divide the thermal image as Region 1 (inside the pool) and Region 2 (outside the pool).

Step 3: Perform head detection algorithm for Region 1. If human is detected, set *reH* as true. *reH* is a variable to signify the presence of intruder.

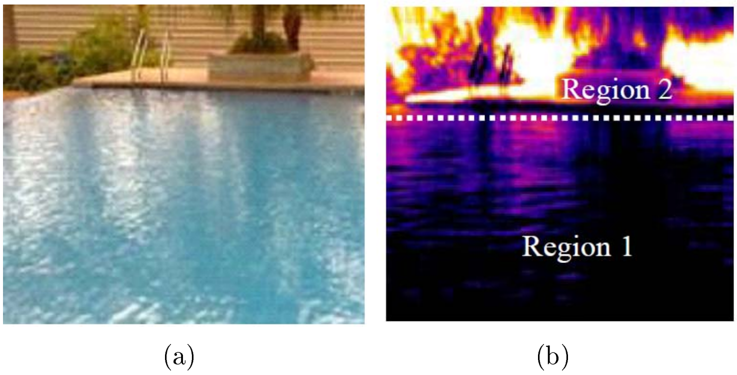
Step 4: Conduct the water activity detection algorithm for Region 1. If water activity is discovered, *reW* is set true. *reW* is a variable to indicate the existence of water activity.

Step 5: (i) If *reH* is true and *reW* is true/false, human is detected, raise the alarm, and increase human counter.

(ii) If *reH* is false and *reW* is false, no human is detected.

(iii) If *reH* is false and *reW* is true, check the previous human counter. If human counter ≥ 1, human is detected, raise the alarm, increase human counter, else no human is detected.

Step 6: Execute the background patching and human detection in Region 2. If human is detected, raise the alarm.



(a) (b)

FIGURE 6. (a) Conventional digital image and (b) equivalent thermal image with Region 1 and Region 2

5.2. **Head detection algorithm.**

Step 1: Obtain the first image, $Img(i)$ and next image, $Img(i+1)$.

Step 2: Extract the red component from both images, which are *bgRed* from $Img(i)$ and *imRed* from $Img(i+1)$.

Step 3: Repeat step 2 for the green component. The outcome will be *bgGreen* from $Img(i)$ and *imGreen* from $Img(i+1)$.

Step 4: Perform subtraction for the red and green component respectively.

$$reRed = bgRed - imRed \tag{1}$$

$$reGreen = bgGreen - imGreen \tag{2}$$

Step 5: Threshold the *reRed* to black and white image, *bwRed* using the equation below. The similar process is done for the *reGreen*, with *bwGreen* as the output. The row and column matrix coordinates are represented by m, n.

$$bwRed(m,n) = \begin{cases} 1, & reRed(m,n) > thRed \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$bwGreen(m,n) = \begin{cases} 1, & reGreen(m,n) > thGreen \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Step 6: Conduct AND logical operation on *bwRed* with *bwGreen* to produce *netRe*.

$$netRe = bwRed\ AND\ bwGreen \tag{5}$$

Step 7: Obtain *imResult* through the subtraction of the $Img(i)$ and the $Img(i+1)$. Then, convert *imResult* to black and white image through thresholding process. *RGpixel* is the addition of the red and green component from the *imResult*.

$$imResult = Img(i) - Img(i+1) \tag{6}$$

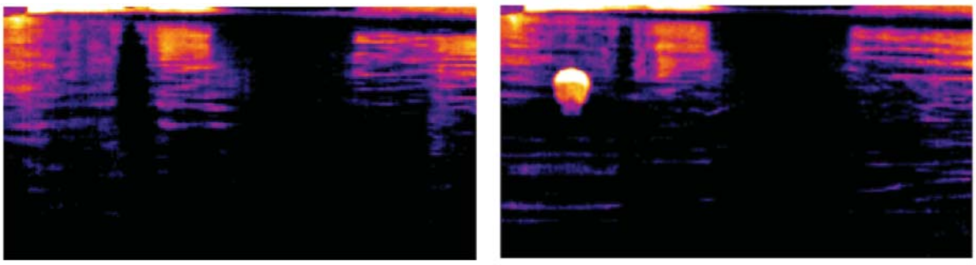
$$RGpixel = imResult(Red) + imResult(Green) \tag{7}$$

$$RedGreen = \begin{cases} 1, & tLow \le RGpixel \le tHigh \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Step 8: Perform *AND* logical operation on *netRe* with *RedGreen*. The resultant image is *reAnd*.

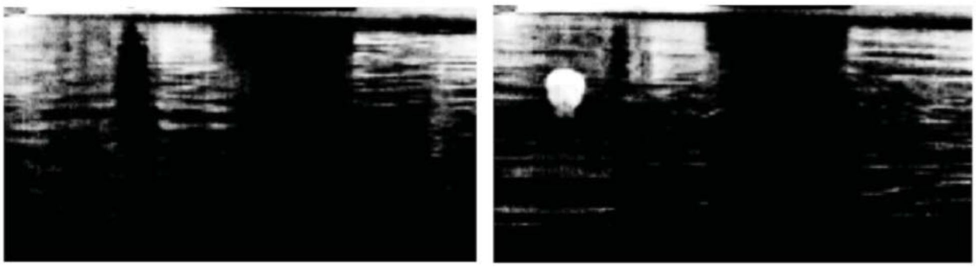
$$reAnd = netRe\ AND\ RedGreen \tag{9}$$

Step 9: If there is no blob in *reAnd* image, it can be concluded that no object is found in the image. Step 1 till Step 8 are repeated until an object is detected in the image. However, if an object is found, the $Img(i)$ will be the reference image (background) and the $Img(i+1)$ will be the current pointer of the image.



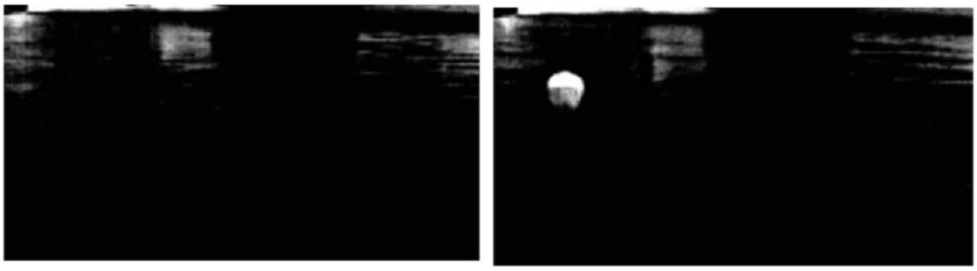
(a) (b)

FIGURE 7. (a) $Img(i)$ and (b) $Img(i+1)$



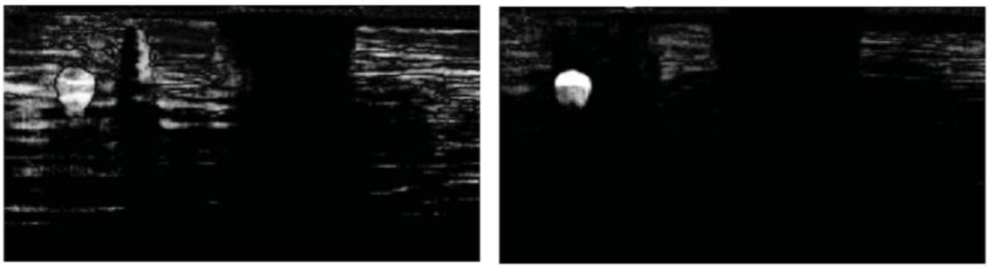
(a) (b)

Figure 8. (a) *bgRed* and (b) *imRed*



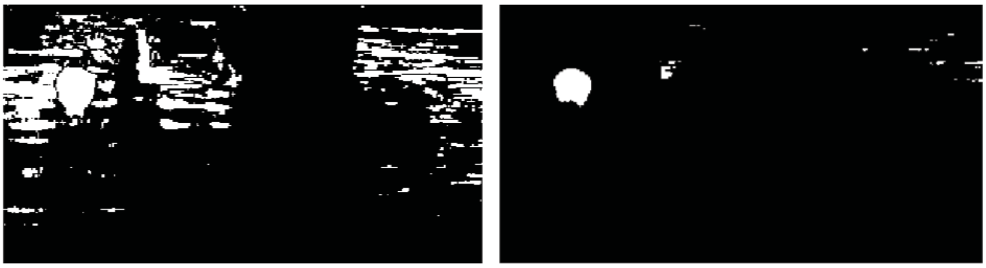
(a) (b)

Figure 9. (a) *bgGreen* and (b) *imGreen*



(a) (b)

Figure 10. (a) *reRed* and (b) *reGreen*



(a) (b)

Figure 11. (a) *bwRed* and (b) *bwGreen*

Step 10: Locate the position of the centroid (Centroid A) of the biggest object area in *reAnd* image. From Figure 10, the human's head has the largest area compared with those of other objects. Later, extract the centroid of each object in the *RedGreen* image

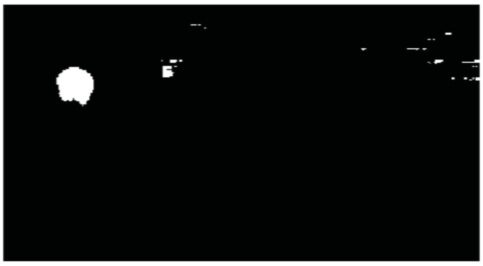
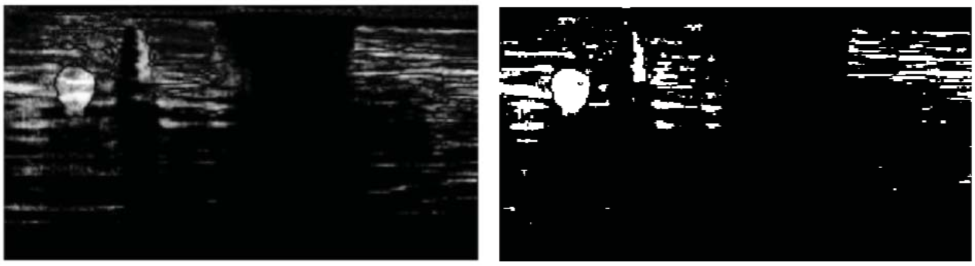


FIGURE 12. *netRe* image after AND logical operation



(a) (b)

FIGURE 13. (a) *imResult* and (b) *RedGreen*

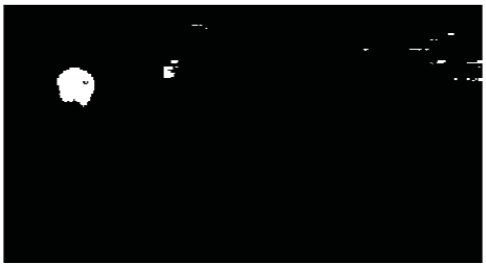
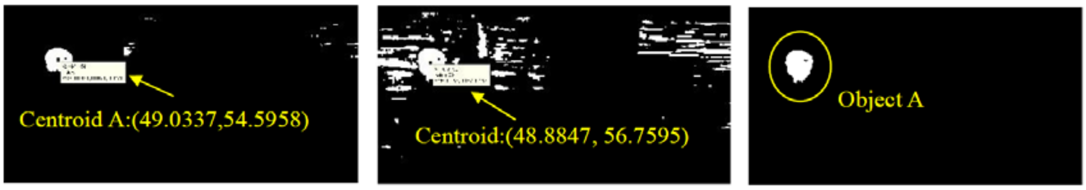


FIGURE 14. *reAnd* image after AND logical operation



(a) (b) (c)

FIGURE 15. (a) Centroid A in *reAnd* image, (b) the closest match of centroid of an object in *RedGreen* image and (c) object A in the *netArea* image

and calculate the distance of the respective centroid with Centroid A. The object (Object A), whose centroid is the closest to the Centroid A, is taken from *RedGreen* image. The outcome of this step is *netArea* image, which contains Object A.

Step 11: Although the object A in *netArea* image is a human head, sometimes noise may appear in *netArea* together with the human head. Therefore, it is necessary to perform extra tests on *netArea* image to detect and verify human head existence. Consider another *netArea* image, which has human head and noise as shown in Figure 16(a). Filter

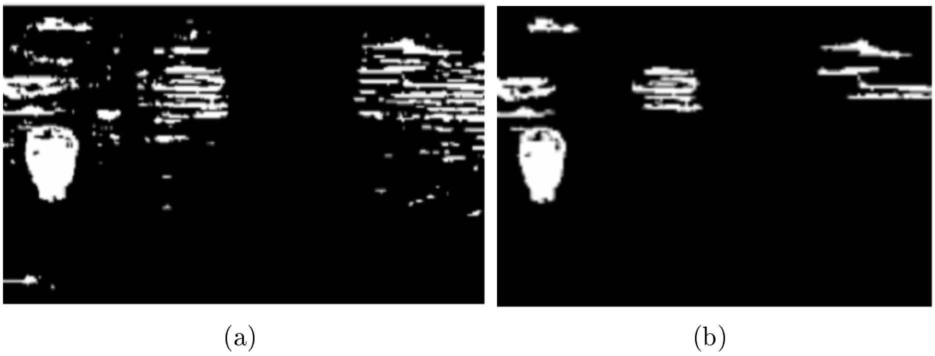


Figure 16. (a) *netArea* image and (b) *bwOpe* image

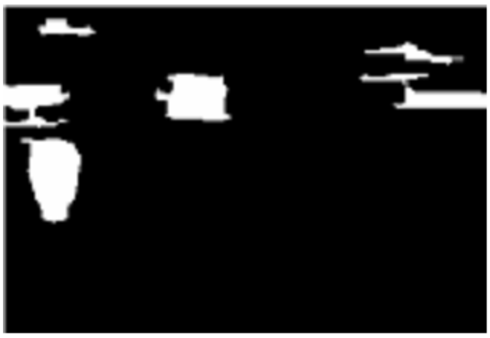


Figure 17. *bwClo* image

the noise in *netArea* to produce *bwOpe*. Any object less than *Avar* pixel is removed. *Avar* is the minimum number of pixel of a detectable human in *netArea* image. Refer to Figure 16(b).

Step 12: Produce a disk element, seA with radius, *dRad*. Conduct morphological closing on *bwOpe*, and get *bwClo* as output. *bwOpe* image is dilated and eroded using the same seA for both operations. Refer to Figure 17.

Step 13: Extract each object's boundary and save it as coordinate arrays.

Step 14: Get the mean x coordinate in the object boundary. The starting point should be the highest point (smallest y coordinate) between intersection points of the boundary with the vertical x mean line. Compare $dVar$ proceeding pixels from the starting point in clockwise and counter clockwise direction to get 2 peaks – Pc and Pcw. The highest point between 2 peaks is the head top point (HeadPpoint). Refer to Figure 18(g).

Step 15: Obtain the mean y coordinate in the object boundary. The intersection of the boundary with the horizontal y mean line gives 2 points – LeftMean and RightMean.

Step 16: Find the number of next pixel (L) using Equation (10) below. Check for L pixel from the LeftMean to look for the Most Left Point in clockwise (LeftMostOutCW point) and counter clockwise (LeftMostOutCCW point) direction. The Most Left Point shall be the point with the smaller x coordinate (most left) between those 2 points. Refer to Figure 18(e).

$$L(pixel) = \frac{x}{100} \times diffP \qquad (10)$$

where $x = LeftRightOut\%$, $diffP =$ the difference between the index of LeftMean and HeadPpoint.

Step 17: Check for L pixel from the RightMean to look for the Most Right Point in clockwise (RightMostOutCW point) and counter clockwise (RightMostOutCCW point) direction. The Most Right Point shall be the point with the higher x coordinate (most right) between those 2 points. Refer to Figure 18(b).

Step 18: Find the number of next pixel (K) using Equation (11) below. Check for K pixel from the Most Left Point to look for the Left South Point in counter clockwise direction. That point should be located at the most right (highest x coordinate) of the Most Left Point. Refer to Figure 18(d).

$$K(pixel) = \frac{z}{100} \times pixelN \tag{11}$$

where $z = NeckPercent\%$, $pixelN =$ number of pixel to build the object boundary.

Step 19: Check for K pixel from the Most Right Point to look for the Right South Point in clockwise direction. That point should be located at the most left (smallest x coordinate) of the Most Right Point. Refer to Figure 18(c).

Step 20: Determine the point (Left Break Top) to divide the boundary for the top left curve test, by finding the number of next pixel (G) using Equation (12). From the HeadPpoint, search for G pixel in counter clockwise direction to have Left Break Top coordinate. Refer to Figure 18(f).

$$G(pixel) = \frac{v}{100} \times pixelR \tag{12}$$

where $v = HeadPercent\%$; $pixelR =$ number of pixel to build the object boundary.

Step 21: Determine the point (Right Break Top) to divide the boundary for the top right curve test, by finding the number of next pixel (G) using Equation (12). From the HeadPpoint, search for G pixel in clockwise direction to have Right Break Top coordinate.

Step 22: Perform Top Curve Test. It consists of Top Curve Left Test & Top Curve Right Test.

22.1 Top Curve Left Test: Extract the coordinate array (LeftHeadCCW) arranged from HeadPpoint to Left Break Top in counter clockwise direction. Use yFilter function to filter any repeated y value coordinate to produce LeftHeadNet. Please refer to Figure 19.

Consider the LeftHeadNet coordinate array has p element and the y value comparison is done by $(p - 1)$ times.

$$countLeft = \begin{cases} 1, & y_{n+1} > y_n \| y_n < y_{n+1} < yTol \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $n = n^{\text{th}}$ element in LeftHeadNet array; $yTol =$ number of pixels can be tolerated for error; $\|$ = or condition.

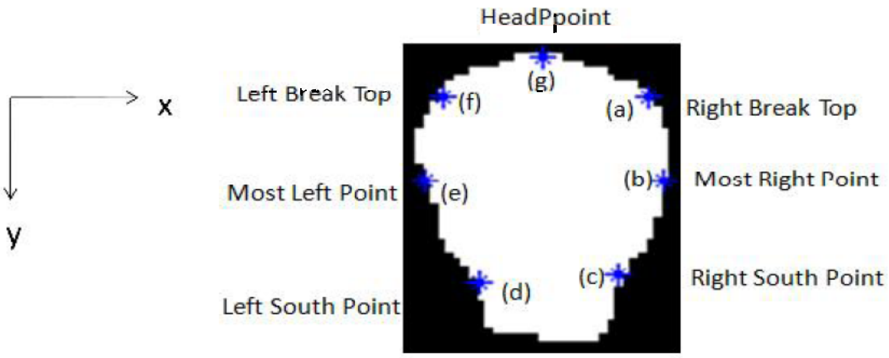


FIGURE 18. (a) Right break top, (b) most right point, (c) right south point, (d) left south point, (e) most left point, (f) left break top and (g) HeadP-point

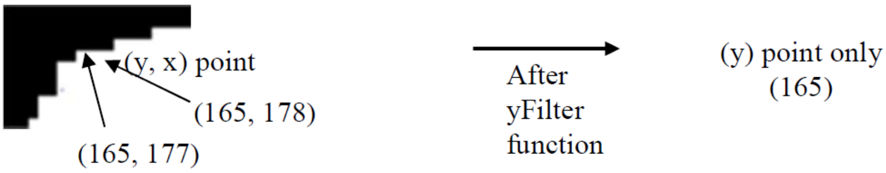


FIGURE 19. yFilter function filters repeated y coordinate

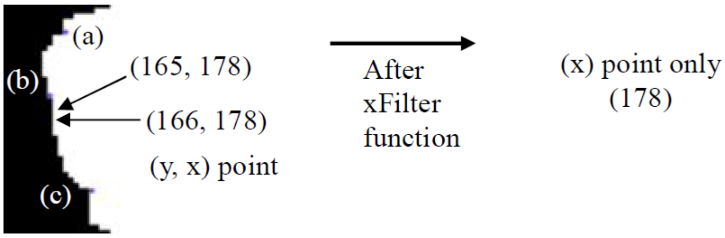


FIGURE 20. xFilter function filters repeated x coordinate. (a) Left Break Top, (b) Most Left Point and (c) Left South Point.

If number of countLeft with 1 is same with $p-1$, the line connecting the HeadPpoint and the Left Break Top is a curve.

22.2 Top Curve Right Test: The test is similar to the Top Curve Left Test by replacing the LeftHeadCCW with RightHeadCW coordinate array, which is arranged from HeadPpoint to Right Break Top in clockwise direction. Later yFilter is applied to produce RightHeadNet, followed by the y value comparison in each RightHeadNet element.

22.3 Combine Top Curve Left and Top Curve Right Test Result: Top Curve Test only confirms the line connecting from Left Break Top, to HeadPpoint till Right Break Top is a curve if the line passes Top Curve Left and Top Curve Right test.

Step 23: Perform Left Curve Test. It consists of Break To Out Test & Neck To Out Test

23.1 Break To Out Test: Extract the coordinate array (BreakLeftCCW) arranged from Left Break Top to Most Left Point in counter clockwise direction. Use xFilter function to filter any repeated x value coordinate to produce NorthLeftNet. Refer to Figure 20.

Consider the NorthLeftNet coordinate array has p element and the x value comparison is done by $(p-1)$ times.

$$coSideLeft = \begin{cases} 1, & x_{n+1} < x_n \| x_{n+1} - x_n < xTol \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where $n = n^{\text{th}}$ element in NorthLeftNet array; $xTol$ = number of pixels can be tolerated for error; $\|$ = or condition

If number of coSideLeft with 1 is same with $p-1$, the line connecting the Left Break Top and the Most Left Point is a curve.

23.2 Neck To Out Test: The test is similar to the Break To Out Test by replacing the BreakLeftCCW with SouthLeftCW coordinate array, which is arranged from Left South Point to Most Left Point in clockwise direction. Later xFilter is applied to produce SouthLeftNet, followed by the x value comparison in each SouthLeftNet element.

23.3 Combine Break To Out and Neck To Out Test Result: Left Curve Test only confirms the line connecting from Left Break Top, to Most Left Point till Left South Point is a curve if the line passes Break To Out and Neck To Out Test.

Step 24: Perform Right Curve Test. It consists of Break To Out Test & Neck To Out Test

24.1 Break To Out Test: Extract the coordinate array (BreakRightCW) arranged from Right Break Top to Most Right Point in clockwise direction. Use xFilter function to filter any repeated x value coordinate to produce NorthRightNet. Consider the NortRightNet coordinate array has p element and the x value comparison is done by $(p-1)$ times.

$$coSideRight = \begin{cases} 1, & x_{n+1} > x_n \| x_n - x_{n+1} < xTol \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where $n = n^{\text{th}}$ element in NorthRightNet array; $xTol$ = number of pixels can be tolerated for error; $||$ = or condition

If number of coSideRight with 1 is same with $p-1$, the line connecting the Right Break Top and the Most Right Point is a curve.

24.2 Neck To Out Test: The test is similar to the Break To Out Test by replacing the BreakRightCW with SouthRightCCW coordinate array, which is arranged from Right South Point to Most Right Point in counter clockwise direction. Later xFilter is applied to produce SouthRightNet, followed by the x value comparison in each SouthRightNet element.

24.3 Combine Break To Out and Neck To Out Test Result: Right Curve Test only confirms the line connecting from Right Break Top, to Most Right Point till Right South Point is a curve if the line passes Break To Out and Neck To Out Test.

Step 25: Perform Head Symmetric Test. Obtain wL and wR, in which wL = width difference between Most Left Point and HeadPpoint and wR = width difference between Most Right Point and HeadPpoint. Subsequently, determine the hL and hR, in which hL = height difference between HeadPpoint and Left South Point and hR = height difference between HeadPpoint and Right South Point. If the wL/wR< 2 or wR/wL< 2 or hL/hR< 2 or hR/hL< 2, the object passes the Head Symmetric Test.

Step 26: Perform Head Curve Test by combining the result from Top Curve Test, Left Curve Test and Right Curve Test. If the object passes any 2 of 3 tests, it passes the Head Curve Test.

Step 27: If the object passes either one of the Head Curve Test or the Head Symmetric Test or both, it is considered as human head, *reH* is set to 1 and human counter increases. The background image is updated if there is no human detected.

5.3. **Water activity detection algorithm.**

Step 1: Obtain the first image (background) and next image. Refer to Figure 21.

Step 2: Extract the black colour pixel from first image, *blackB4* and from the next image, *blackAf*.

Step 3: Calculate the percentage of the black pixel intensity for first image and next image using equation below.

$$BlackPixel\% = \frac{t}{w \times d} \times 100\% \tag{16}$$

where $t = blackB4$ or $blackAf$; w = image width; d = image length.

Step 4: Find the difference (reBlackDiff) between the black pixel % between *blackB4* and *blackAf* image.

Step 5: If reBlackDiff > minimum changes in black pixel % (*thWater*) due to human activity, the Water Activity Detection algorithm produces *reW* as true to indicate possible human intrusion due to the existence of water activity in swimming pool.

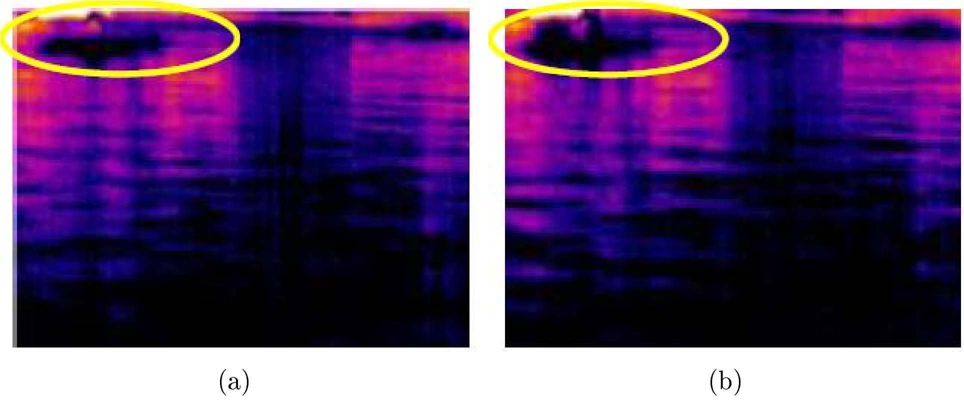


FIGURE 21. (a) *blackB4* image and (b) *blackAf* image

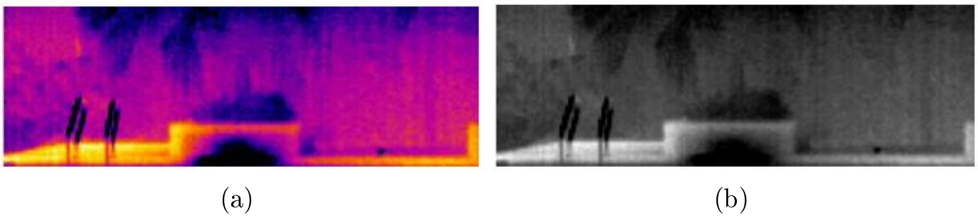


FIGURE 22. (a) $(i-1)^{\text{th}}$ image in RGB and (b) *BG* image after rgb2gray function

5.4. Surveillance for region outside swimming pool (Region 2).

5.4.1. *When no human/animal is outside the swimming pool.*

Step 1: Before the commencement of the detection, several variables are set accordingly as demonstrated below:

$$huC = anC = 0; \tag{17}$$

$$noObj = preNoHu = T; \tag{18}$$

$$i = 2; \tag{19}$$

where *huC* is the human counter, *anC* is the human counter and *i* indicates the image position. Meanwhile, *noObj* represents no object in the current image and it is updated after the condition test. Lastly, *preNoHu* shows no human existence in $(i-1)^{\text{th}}$ image.

Step 2: Perform the condition test on the *noObj* variable. Since the *noObj* is set to true at the setup, it will pass the condition test, else the background patching algorithm is initiated.

$$noObj = \begin{cases} 1, & \text{Go to Step 3} \\ 0, & \text{Go to background patching} \end{cases} \tag{20}$$

Step 3: Another condition test takes place to verify the *preNoHu* variable. Similar to *noObj*, it has been set as true at the setup, and will pass the condition test.

$$preNoHu = \begin{cases} 1, & BG = rgb2gray(im\ i-1) \\ 0, & \text{Go to Step 4} \end{cases} \tag{21}$$

Since $i = 2$ at the setup, $i - 1 = 1$. Background image (BG) is a grayscale image converted from a RGB (red-green-blue) image of i^{th} using rgb2gray function. Refer to Figure 22 below.

Step 4: i^{th} image, which is 2 for this example, is converted from RGB to grayscale image, which is saved as image *current*. Later, the *current* image is subtracted from *BG*

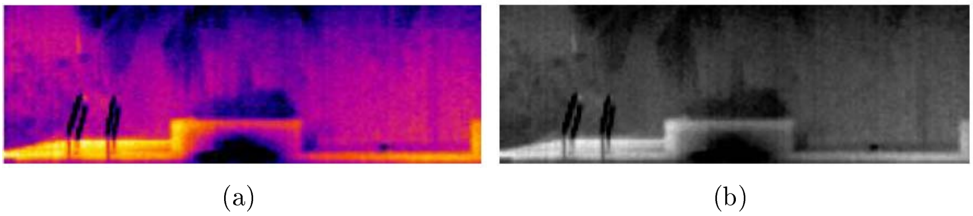


FIGURE 23. (a) i^{th} image in RGB and (b) *current* image after rgb2gray function

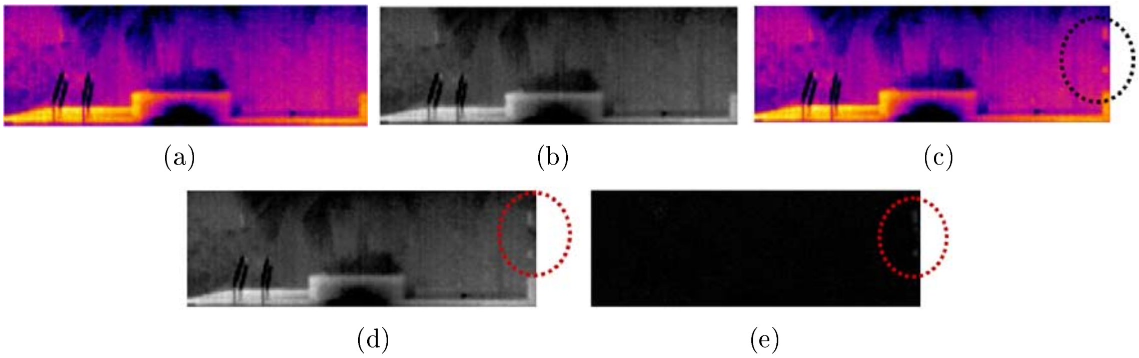


FIGURE 24. (a) $(i-1)^{\text{th}}$ image in RGB, (b) *BG* image after *rgb2gray* function, (c) i^{th} image in RGB, (d) *current* image after *rgb2gray* function and (e) *imRe1* image

image using *imabsdiff* MATLAB built-in function. The output of the subtraction is stored as *imRe1* image.

$$current = rgb2gray(i^{\text{th}}\ image) \tag{22}$$

$$imRe1 = current - BG \tag{23}$$

Take note that $imRe1$ is a blank black image as *current* image is same with *BG* image.

Step 5: Check the *noObj* variable. If it is true, the program will proceed to Step 6, else it will enter the Image Enhancement stage, which will be discussed later.

$$noObj = \begin{cases} 1, & \text{Go to Step 6} \\ 0, & \text{Go to Image Enhancement Stage} \end{cases} \tag{24}$$

Step 6: $imRe1$ image is converted from grayscale to black and white image through threshold process. Subsequently, the converted image has its noise filtered and all the blobs located nearby are combined into one element for the subsequent condition tests (to be explained in Step 5 at Section 5.4.2)

Since *current* and *BG* image are empty, the resultant image remains the same as the $imRe1$ after the threshold, noise-filtering, and blobs combination processes. A better illustration can be found in Section 5.4.2, Step 2-Step 4, where a human is detected.

Step 7: Measure the blobs height to determine whether a human or a dog is detected. Since no blob is found, so the *preNoHu* and *noObj* remain as true and the i variable is incremented to take the next image.

5.4.2. *When* $(i-1)^{\text{th}}$ *image has no human/animal, and* $(i)^{\text{th}}$ *image has human/animal.*

Step 1: Steps 1-5 from Section 5.4.1 are performed, except that the *current* image has a human detected. In the example below, assume $i = 19$.

FIGURE 25. *bwRe*2 image after thresholding process

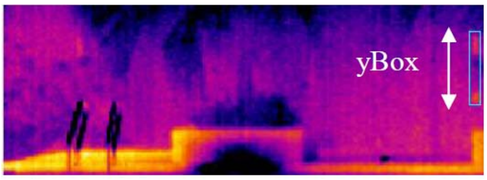


FIGURE 26. *yBox* and the printed box in i^{th} image

Take note of the human presence (dotted circle) in Figure 24(e) after Equation (23) is conducted.

Step 2: Threshold process takes place, in which *imRe*1 image is converted into black and white image based on the equation below. The resultant image is *bwRe*2 image.

$$bwRe2(m,n) = \begin{cases} a, & imRe1(m,n) > thIm \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

where *thIm* is the minimum grayscale value needed to represent the human/animal object in the *imRe*1 image. (m,n) is the pixel coordinate in the image.

Step 3: The noise is filtered to remove any object lesser than *Avar*2 pixel. The resultant image is *bwF*2.

Step 4: The blobs found in the *bwF*2 image is combined by finding the horizontal (*xDist*) and vertical (*yDist*) separation distance between each blob. If the $xDist < xGap$ and $yDist < yGap$, the blob is combined with its adjacent blob. *xGap* is the maximum possible distance between two separated blobs due to the human/animal movement. For instance, the maximum separation distance between two legs defines the *xGap* when the human is stretching his leg while walking. Whereas, the *yGap* is the maximum possible distance between two separated blobs due to the human/animal height.

Step 5: The identification of a human or an animal is performed by checking the height of the blob, *yBox*. If *yBox*> the minimum possible human height, *yHuman*, the *huC* is incremented by 1, *noObj* is set to false, and the box is printed. If $yBox < yHuman$, *yBox* is compared with *yAni*, which is the minimum possible animal height (dog/cat). If $yBox > yAni$, *anC* is incremented, *noObj* is set to false, and another colour box is printed. If $yBox < yAni$, the program will execute Step 7 in Section 5.4.1.

Step 6: i is incremented to take the next image, which is the 20th image, and program enters into Section 5.4.3 phase.

5.4.3. *When* $(i-1)^{\text{th}}$ *and* $(i)^{\text{th}}$ *image have human/animal.* From Section 5.4.2, the affected parameters are listed as below:

$$huC = 1; \quad anC = 0; \tag{26}$$

$$noObj = F; \quad preNoHu = T; \tag{27}$$

$$i = 20; \tag{28}$$

Step 1: Check the *noObj* value. Based on Equation (20), background patching happens in this section. The $(i-1)^{\text{th}}$ image becomes *oldIm* image. In this example, Figure 24(c), which is the 19th image becomes *oldIm* image.

$$oldIm\ image = (i-1)^{\text{th}}\ image \tag{29}$$

Step 2: *preNoHu* is checked to determine the valid old background image, *oldBG*. If the *preNoHu* = true, the $(i-2)^{\text{th}}$ image is set as *oldBG* image and the *preNoHu* is set to false, else *BG* image will be used as the *oldBG* (to be discussed further in Step 11). In this scenario, 18th image, Figure 24(a) is used as *oldBG* image.

$$oldBG\ image = (i-2)^{\text{th}}\ image \tag{30.1}$$

$$preNoHu = F \tag{30.2}$$

Step 3: Background patching occurs here. The pixels, which are covered by the printed box from Step 5 in Section 5.4.2 are stored. Then the RGB values are copied from *oldBG* image to *oldIm* image, overwriting the original RGB values in *oldIm* image, based on that stored pixels previously. The equation below illustrates the background patching idea.

$$oldIm(j,i) = oldBG(j,i) \tag{31}$$

where the j is the column pixel, while the i is the row pixel from the printed box coverage.

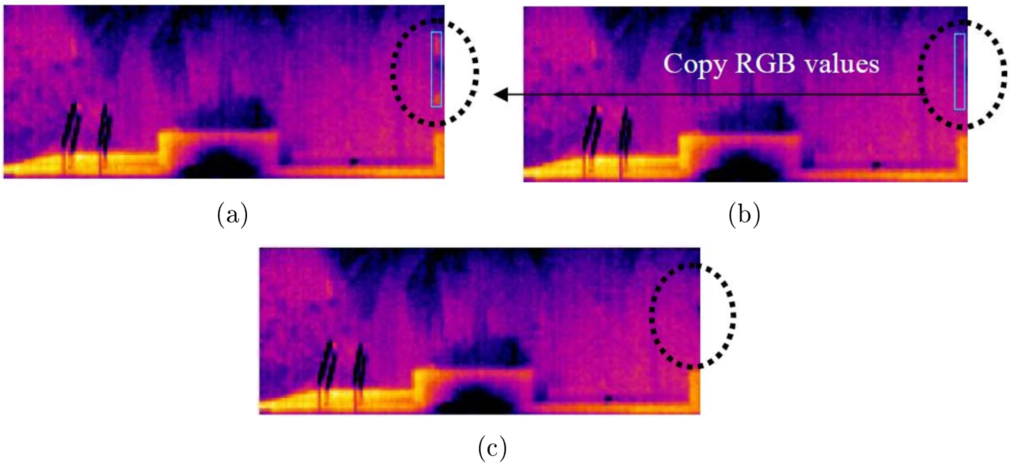
Step 4: The new *oldIm* image is converted to a grayscale image using *rgb2gray* MATLAB built-in function. The output of this conversion is a *BG* image. Subsequently, *current* and *imRe*1 image are obtained from Step 4, which is originated from Section 5.4.1, using $i = 20$.

Step 5: The similar process from Step 5, coming from Section 5.4.1, is conducted. As the *noObj* is false, the algorithm initiates the Image Enhancement procedure. Firstly, $(i-2)^{\text{th}}$ image is converted as grayscale image, using *rgb*2*gray* function. The output is *gryOldBG* image, as shown in Figure 24(b).

$$gryOldBG = rgb2gray(i-2)^{\text{th}}\ image \tag{32}$$

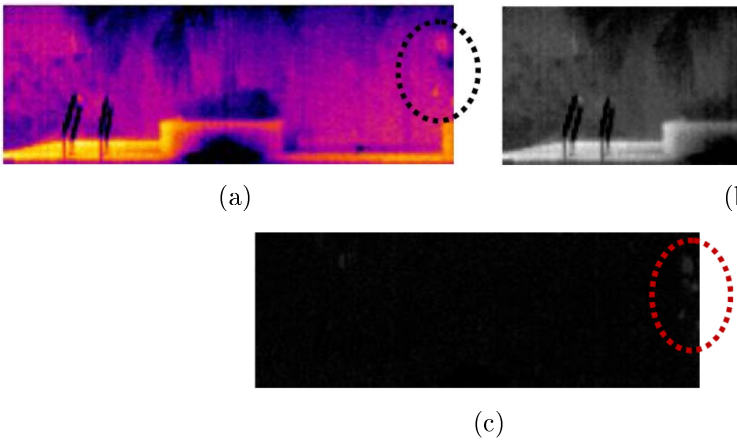
Step 6: The *gryOldBG* image is subtracted with *current* image, utilising *imabsdiff* function, found in MATLAB. The output of the subtraction is *imRe2* image.

$$imRe2 = current - gryOldBG \tag{33}$$



(a)
(b)
(c)

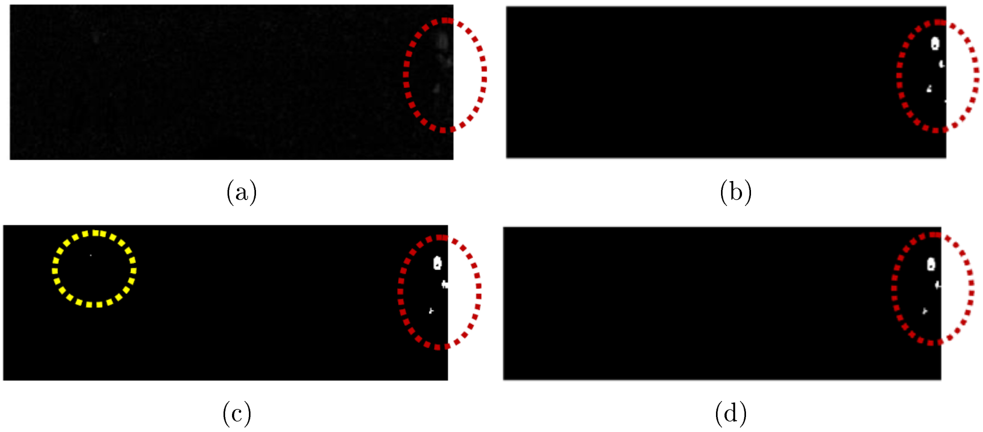
FIGURE 27. (a) *oldIm* image, (b) *oldBG* image and (c) new *oldIm* image after background patching



(a) (b)

(c)

FIGURE 28. (a) i^{th} image, (b) *current* image and (c) *imRe*2 image



(a) (b)

(c) (d)

FIGURE 29. (a) *imRe*1 image, (b) *bwRe*1 image, (c) *bwRe*2 image, (d) *bwRe*3 image

Step 7: Threshold *imRe*2 and *imRe*1 images, using the similar procedure in Step 2 from Section 5.4.2.

$$bwRe1(m,n) = \begin{cases} 1, & imRe1(m,n) > thIm \\ 0, & \text{otherwise} \end{cases} \tag{34.1}$$

$$bwRe2(m,n) = \begin{cases} 1, & imRe2(m,n) > thIm \\ 0, & \text{otherwise} \end{cases} \tag{34.2}$$

Step 8: The *bwRe*1 and *bwRe*2 images are used to perform AND logical operation. The output is *bwRe*3.

$$bwRe3 = bwRe1\ AND\ bwRe2 \tag{35}$$

The red circle shows the human existence, while the yellow circle depicts the noise in the image.

Step 9: The *bwRe*3 will have its noise filtered and its blobs combined, similar to Steps 3 and 4 respectively in Section 5.4.2. Then, the algorithm runs Step 5 in Section 5.4.2 to check the height of the combined blobs in order to verify the existence of a human or an animal.

Step 10: i is incremented to take the next image, which is number 21st.

Step 11: Step 1 in Section 5.4.3 is repeated; however, in Step 2, the *preNoHu* has been set to false previously. So the *BG* image takes the *oldBG* image, which is 18th image, because 19th image contains human object, causing it to be unsuitable for background patching.

5.4.4. *When* $(i-1)^{\text{th}}$ *has human and* $(i)^{\text{th}}$ *image does not have human.* When this situation happens, the *yBox* is less than *yHuman* and *yAni*. As a result, the *preNoHu* and *noObj* are set as true. After that, Step 7 in Section 5.4.1 is executed and next image is taken for processing.

6. Experimental Results for Region 1 Algorithm.

Several experiments are conducted to determine the optimum parameters stated in the algorithm. The setting of the experiment is at Ixora Condominium Park, which has a swimming pool. The following diagram illustrates the position of the thermal camera and the swimming pool.

Firstly, 4 sets of 10000 thermal images (Sets 1, 2, 3, 4) with test subjects (either human or animal) walking in random in the vicinity of the swimming pool and in the pool under the camera surveillance angle. The thermal images from Sets 1 and 2 are used to determine optimal parameters for *thWater*. Meanwhile, Set 3 thermal images are tested with the algorithm to optimise parameter of *tHigh*, *tLow*, *dRad*, *dVar*, *yTol*, *xTol*, and *Avar*. Lastly, the optimal value for *LeftRightOut%*, *NeckPercent%*, and *HeadPercent%* are calculated by experimenting with thermal images from Set 4.

thWater parameter is used as a threshold value, which represents the minimum changes in black pixel % from 2 consecutive images caused by human activity. From the observation of 10000 thermal images in Set 1, the black pixel changes tremendously due to human activity in water like splash and others. However, external environmental factors such as water ripples caused by outdoor winds at the pool, may affect the changes in black pixels in the images. Therefore, anemometer is employed to measure the wind speed changes between 2 consecutives images. 10 values $(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ knots for wind speed difference are selected, each with 1000 images respectively. From there, black pixel changes in consecutive images can be obtained from no wind environment followed by normal till the strongest wind at the swimming pool.

Based on Figure 31, it is clearly visible that the *thWater* optimal value is chosen as 10 because such wind strength difference is impossible in a normal condition. Hence, any water activity caused by human will exceed the *thWater* and set *reW* to 1. Apart from that, wind speed difference in two consecutive images during rain is considered in optimising the *thWater* value. Anemometer is used to measure wind speed and rain gauge is used to obtain rain intensity readings during rain condition in different rain intensities ranging from no rain, followed by mild and heavy rain. Similar to previous approach,

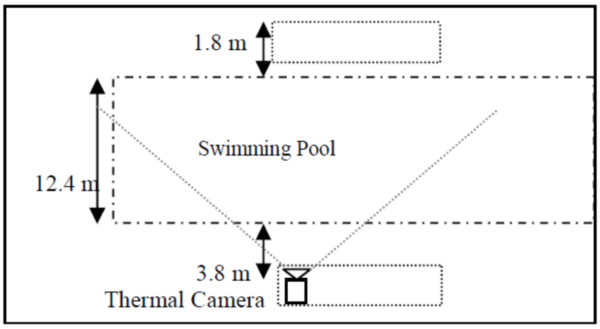


FIGURE 30. Location of the swimming pool and thermal camera

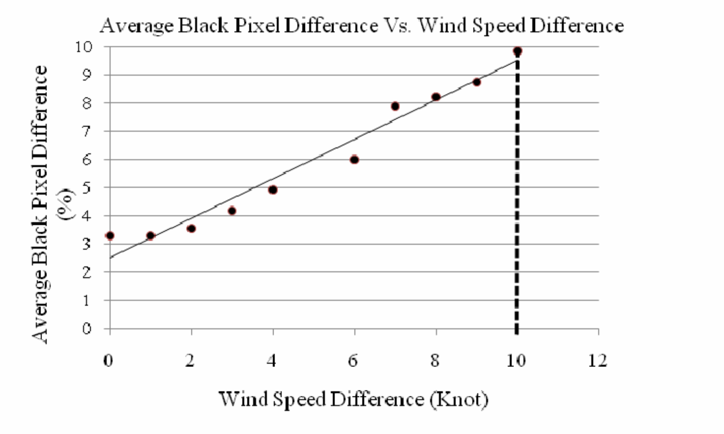


FIGURE 31. Average black pixel difference in 1000 thermal images (%) vs wind speed difference between 2 consecutive images (knot)

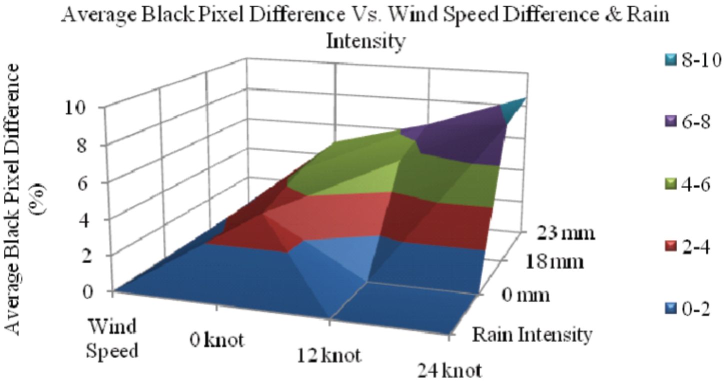


FIGURE 32. Average black pixel difference in 1000 thermal images (%) vs wind speed difference (knot) and rain intensity (mm)

average black pixel changes in 2 consecutive images are measured with respect to various combinations of wind speed difference and rain intensity. By setting the $thWater = 10\%$, false alarm in detecting water activity due to rain activity can be reduced. Refer to Figure 32.

To determine the optimal value for the parameters of *tHigh*, *tLow*, *dRad*, *dVar*, *yTol*, *xTol*, and *Avar*, thermal images from Set 3 are tested with the algorithm. In order to find a suitable *Avar* value, thermal images are recorded with a human positioned at the furthest distance from the camera (18m), located in Region 1. The pixel area of the human object is recorded and the chosen *Avar* value is 100 pixels. Meanwhile, *dRad* is needed to plot disk element, seA for morphological process. Again, 10000 images are tested and processed till Step 12 at Section 5.2 in order to determine which *dRad* value produces the most suitable dilated and eroded images. From Figure 33, it is obvious that $dRad = 3$ is a suitable value.

Conversely, *yTol* and *xTol* parameters optimisation are conducted by testing the algorithm on 10000 images with different *yTol* and *xTol* values. The algorithm accuracies are compared to determine the most suitable corresponding parameters. Here, the most suitable *yTol* and *xTol* are 3 for each.

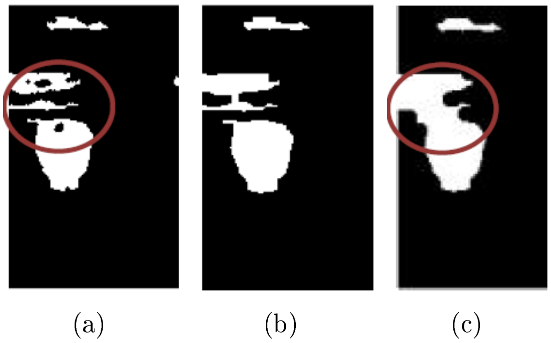


Figure 33. (a) dRad = 1, (b) dRad = 3 and (c) dRad = 5

Table 1. Overall accuracy for different parameters of yTol & xTol

	xTol				
yTol	1	2	3	4	5
1	70.54%	81.32%	83.49%	80.78%	82.32%
2	74.88%	85.23%	85.67%	78.32%	74.88%
3	79.55%	88.93%	89.45%	81.58%	80.21%
4	84.66%	87.78%	86.45%	85.90%	83.45%
5	85.89%	84.55%	80.34%	78.23%	75.67%

Table 2. Overall accuracy for different parameters of dVar

	dVar			
	2	3	4	5
Accuracy (%)	84.11%	86.56%	92.44%	89.44%

Table 3. Overall accuracy for different parameter of tHigh & tLow

	tHigh		
tLow	310	410	510
100	92.3134%	92.44031%	92.44306%
110	91.0971%	91.1259%	91.20083%
120	92.1043%	92.1289%	92.13251%

Additionally, the steps of determining optimum values for the parameters of *dVar*, *tHigh* and *tLow* are similar to Chew's work in [21], in which accuracies are compared with different parameters values. Table 2 presents the comparison result.

From the tables above, it is understood that parameters with *tHigh* = 510, *tLow* = 100, and *dVar* = 4 will produce the most optimum result.

Lastly, in order to find out the optimal value for *LeftRightOut%*, *NeckPercent%,* and *HeadPercent%*, observation on the human head is conducted by experimenting with thermal images from Set 4. Later, all the 7 points from Figure 18 are marked at a desired position on the human head from different angle. This process is repeated for all 10000 thermal images in the set to get the average *LeftRightOut%*, *NeckPercent%,* and *HeadPercent%* with respect to the number of pixels, which surround the human head boundaries. Please refer to Figure 34 for a better illustration.

From Figure 34, the arrow which connects the (a) and (d) points represents *HeadPercent%,* while the arrow linking points (b) and (c) is *NeckPercent%.* Here, the selected values for those 3 parameters are *HeadPercent%* = 14*%, NeckPercent%* = 15%, and *LeftRightOut%* = 3%.

7. **Experimental Results for Region 2 Algorithm.** Similar to Section 6, 5 sets of 10000 thermal images (Sets 5, 6, 7, 8, 9) with different movements of humans and animals are prepared. Set 5 thermal images are used to optimise the *thIm* variable, which is used in the threshold process. *thIm* is crucial in determining the effectiveness of the image conversion from grayscale to black and white format. Besides, the *xGap* value is calculated by experimenting with thermal images from Set 6. The maximum distance between two legs spread by the human defines the *xGap.* Meanwhile, the value for *yGap* is specified by the maximum height of a human and the thermal images from Set 7 are utilised for that purpose. Lastly, the optimum values for the *yHuman* and *yAni* parameters are discovered through the image processing of the thermal images in sets 8 and 9 respectively. *yHuman* is a minimum height of a human, when he is in the squat position. Whereas, the minimum height of a dog, which is sleeping, determines the *yAni* parameter.

thIm plays a vital role as a threshold value. It expresses the minimum grayscale value in the resultant image after the image subtraction is conducted with *imabsdiff* function. Random movements of humans and animals are recorded with 10000 thermal images. From the observation, several proposed *thIm* values are attempted. The effectiveness of the conversion process is measured by finding the accuracy of each *thIm* value. Table 4 describes the comparison result.

Based on Table 4, the most suitable *thIm* variable value is 30.

In order to attain the optimum values for the *xGap*, 10000 images of human spreading his leg as wide as possible are recorded. The images are processed to achieve the mean values of the separation distance between two legs. This distance is chosen as the *xGap* to combine the isolated blobs of a human after the images are threshold and filtered. The animal (dog/cat) is unable to stretch its leg as wide as human. Thus, the maximum *xGap* value is taken from the maximum width of two legs being spread by human. Figure 35 shows the *xGap* distance in the images.

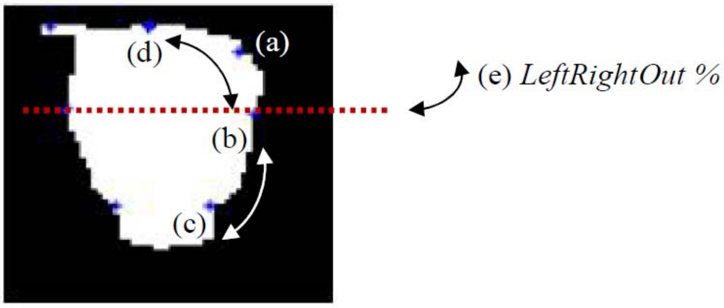


FIGURE 34. (a) Right break top, (b) most right point, (c) right south point, (d) HeadPpoint and (e) *LeftRightOut%*

TABLE 4. Overall accuracy for different parameter of thIm

	thIm			
	20	30	40	50
Accuracy (%)	93.23%	95.67%	94.41%	93.29%

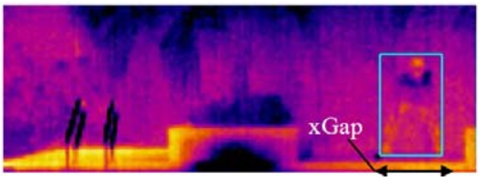


FIGURE 35. *xGap* is measured from the width of 2 legs spread by a human. The mean value is 40.

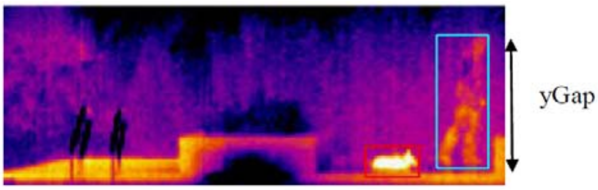
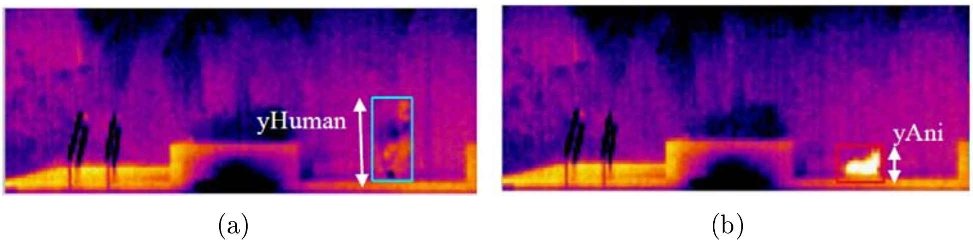


FIGURE 36. *yGap* is measured from the height of a standing human. The mean value is 60.



(a) (b)

FIGURE 37. (a) *yHuman* is measured from the height of a squatting human. The mean value is 45. (b) *yAni* is taken from the height of a sleeping dog. The mean value is 15.

Moreover, the *yGap* is obtained through the measurement of the maximum height of a human in a standing position. Similar to the approach executed in *xGap*, 10000 thermal images record the standing human, and the mean value for the *yGap* is calculated accordingly. The animal is definitely shorter than the human. For this reason, the human height is taken as the *yGap* value. When the blobs are combined, *yGap* can include the blobs, which are separated by human height and also the animal height.

Finally, the *yHuman* and *yAni* parameters optimisation are conducted by experimenting thermal images from Sets 8 and 9. Both values are used to verify the existence of a human or animal. Comparable to *xGap* and *yGap*, the mean value of 10000 thermal images is calculated for *yHuman* and *yAni*. *yHuman* is taken from the height of a human when he is squatting. Such position is chosen for a human to produce the minimum possible height. Any object higher than the *yHuman* is considered as human. On the contrary, if the object is shorter than *yHuman*, it is then compared with *yAni*; which is the minimum possible height of an animal. In this scenario, the *yAni* is extracted from the height of a sleeping dog. Any object, which has a height in between *yAni* and *yHuman* is deemed as animal. If the height of the object is shorter than *yAni*, it is classified as noise.

8. **Overall Accuracy and Performance Comparison.** All in all, the performance of the proposed algorithm is good as it achieves a high accuracy for both Region 1 and Region 2. There are three different results. The system can attain 88.63% accuracy for detecting intruders inside the swimming pool (Region 1), using head detection only. Meanwhile,

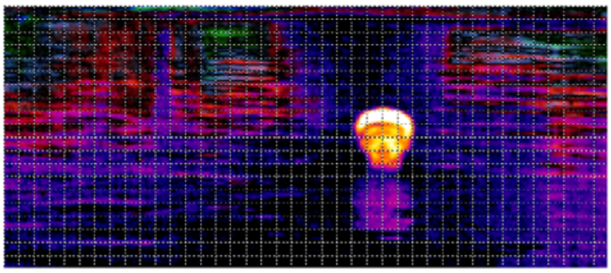


Figure 38. Partition region of interest (ROI) technique prepared by [20]

the accuracy is higher for surveillance system outside the swimming pool (Region 2), which is 95.58%. The third result is obtained by combining the head and water activity detection for trespassers detection inside the swimming pool. As high as 92.44% accuracy is achieved for that surveillance system. Overall, the accuracy for the surveillance system inside and outside the swimming pool region is 94%, by taking the average of the second and the third results above.

Performance comparison is also conducted to measure the robustness of the surveillance system. In this project, Partition Region of Interest (ROI) technique prepared by [20] is modelled to measure the accuracy. Three main parameters in that algorithm are declared as follows, Q = 219, H = 15 and G = 5. As explained earlier in Section 3.1, that algorithm has a near-far problem. Hence, the measured accuracy is 41.20%, which is weaker than the proposed methodology in this project by at least twice.

9. Conclusion.

In this paper, an off-time swimming pool surveillance system using thermal imaging system has been proposed, and it consists of automatic intruder detection capability (human detection and water activity detection). It works with a thermal camera, which first captures and divides the images into 2 regions, and later employing the head detection for both regions and water activity for Region 2. The purpose of this surveillance system is to prevent any intruder especially playful children from entering the swimming pool during off-time. With such a system, drowning cases can be reduced. Swimming pools in hotels or residential areas, and mining pools in industrial levels are suitable for such application. This system is able to achieve as high as 92.44% for Region 2 with the help of head and water activity detection. In the future, the surveillance system will be enhanced into an omnidirectional (360°) view to increase the surveillance coverage angle using minimum hardware. The current viewing angle of the FLIR's ThermoVision A20M is $19^\circ \times 14^\circ$. In order to monitor the entire swimming pool premise, several thermal cameras are required and this can increase the budget of the system. The system can be more cost-effective by introducing this omnidirectional view to reduce number of thermal cameras [24]. An image captured by the omnidirectional view has to be unwrapped into a panoramic image so the image information can be retrieved from there. To unwrap an image, there are three main methods available, which include discrete geometry method (DGT), log-polar mapping method and pano-mapping table method [25].

REFERENCES

[1] J. Huge, *Book of Child Care*, Limited Edition, HarperCollins Publishers, Canada, 1986.

[2] Centers for Disease Control and Prevention, *Injury Prevention & Control: Home and Recreational Safety*, http://www.cdc.gov/HomeandRecreationalSafety/Water-Safety/waterin juries-factsheet. html, 2011.

[3] L. J. Chan, *Stay Safe Around Water*, http://thestar.com.my/lifestyle/story.asp?file=/2011/5/25/life parenting/8722359&sec=lifeparenting, 2011.

[4] J. C. Miller, M. L. Smith and M. E. McCauley, Crew fatigue and performance on U.S. coast guard cutters, *U.S. Coast Guard Research & Development Center*, 1998.

[5] C. K. Eveland, D. A Socolinsky and L. B. Wolff, Tracking human faces in infrared video, *Image and Vision Computing*, vol.21, pp.579-590, 2003.

[6] B. Sugandi, H. Kim, J. K. Tan and S. Ishikawa, Real time tracking and identification of moving persons by using a camera in outdoor environment, *International Journal of Innovative Computing, Information and Control*, vol.5, no.5, pp.1179-1188, 2009.

[7] R. Kayne, *What is a Motion Detector?* http://www.wisegeek.com/what-is-a-motion-detector.htm, 2011.

[8] Banner Engineering Corp (2006), *Q & A: Ultrasonics Basics*, http://www.bannerengineering.com /training/faq.php?faqID=34&div=1#content, 2011.

[9] *Ultrasonic Motion Sensors*, http://www.homesecurityguru.com/ultrasonic-motion-sensors, 2011.

[10] *Ultrasound*, http://sensorwiki.org/doku.php/sensors/ultrasound, 2011.

[11] *Kongsberg Wins $3 Million US Coast Guard Underwater Surveillance Contract*, http://www.km.kongsberg.com/ks/web/nokbg0238.nsf/AllWeb/A94BADEE4C4F081FC1257000002770B9?OpenDocument, 2011.

[12] Poseiden. (n.d), *The Poseidon*, http://www.poseidon.fr/fr/system.html, 2011.

[13] Sofradir EC, Inc. (n.d), *How Night Vision Works*, http://www.hownightvisionworks.com/, 2011.

[14] FLIR. (n.d), *What's The Difference between Thermal Imaging and Night Vision?* http://www.flir.com/cvs/americas/en/content/?id=9612, 2011.

[15] Home Security Guru. (n.d), *Passive Infrared Sensors*, http://www.homesecurityguru.com/passive-infrared-pir-sensors, 2011.

[16] Home Security Guru. (n.d), *Microwave Motion Detector Guide*, http://www.homesecurityguru.com/microwave-motion-sensors, 2011.

[17] W. K. Wong, Z. Y. Chew, H. L. Lim, C. K. Loo and W. S. Lim, Omnidirectional thermal imaging surveillance system featuring trespasser and faint detection, *International Journal of Image Processing, CSC Journals*, vol.4, no.6, pp.518-538, 2011.

[18] *Thermal Imager BEYOND COMPARE Frequently Asked Questions*, http://www.hurleyir.com/faqs.html, 2011.

[19] J. Han and B. Bhanu, Fusion of color and infrared video for moving human detection, *Pattern Recognition*, vol.40, no.6, pp.1771-1784, 2007.

[20] W. K. Wong, P. N. Tan, C. K. Loo and W. S. Lim, An effective surveillance system using thermal camera, *Proc. of International Conference on Signal Acquisition and Processing*, Kuala Lumpur, Malaysia, pp.13-17, 2009.

[21] W. K. Wong, Z. Y. Chew, C. K. Loo and W. S. Lim, An effective trespasser detection system using thermal camera, *Proc. of the 2nd International Conference on Computer Research and Development*, pp.702-706, 2010.

[22] W. K. Wong, H. L. Lim, C. K. Loo and W. S. Lim, Home alone faint detection surveillance system using thermal camera, *Proc. of the 2nd International Conference on Computer Research and Development*, pp.747-751, 2010.

[23] W. K. Wong, J. H. Hui, C. K. Loo and W. S. Lim, Off-time swimming pool surveillance using thermal imaging system, *Proc. of IEEE International Conference on Signal and Image Processing Applications*, pp.366-371, 2011.

[24] B. M. Zalili and W. Junzo, Multi-camera tracking system for human motions in different areas and situation, *International Journal of Innovative Computing, Information and Control*, vol.4, no.5, pp.1213-1222, 2008.

[25] W. K. Wong, W. S. Pua, C. K. Loo and W. S. Lim, A study of different unwarping methods for omnidirectional imaging, *Proc. of IEEE International Conference on Signal and Image Processing Applications*, pp.427-432, 2011.