MONITORING SYSTEMS FOR COUNTING PEOPLE BASED ON WIRELESS MULTIMEDIA SENSOR NETWORK

Kalvein Rantelobo[1*], A. Sampeallo[1], J. F. Mandala[1]
H. F. J. Lami[1], B. Bernandus[1], P. H. Rantelinggi[2], N. P. Sastra[3]
[1)] Universitas Nusa Cendana, Jl. Adisucipto, Penfui, Kupang, NTT, Indonesia
[2)] Universitas Papua, Jl. Gn. Salju, Amban, Manokwari, Papua, Indonesia
[3)] Universitas Udayana, Denpasar, Bali, Indonesia
Email: kalvein@staf.undana.ac.id, agusthinus.sampeallo@staf.undana.ac.id, janifm99@yahoo.co.id, h.lami@staf.undana.ac.id, bernandus@staf.undana.ac.id, parma.hadi.id@ieee.org, n.p.sastra@unud.ac.id

Info Artikel

Article history:
Received Nov 14, 2024
Revised Des 15, 2024
Accepted Feb 28, 2025

ABSTRACT

In this work, the visual sensor used in the Wireless Sensor Multimedia Networks (WSMN) aims to monitor and calculate the number of people passing through a room. The contribution of this paper is the use of Raspberry Pi 3 devices that are connected to the internet using Internet-of-Thing (IoT) technology. The proposed scheme can be implemented in the actual environment. From the test results, the system distinguished people entering and leaving the room by processing the image using background subtraction, using the morphological transformation method, and calculating the contour area of the picture. Image processing results can calculate the number of people in the room, and the system can send it to the web server. Subsequently, this paper discussed the energy consumption used by the WSMN and explained test parameters.

Keywords: *WMSN; Visual Sensor; Wi-Fi; People Counting; Raspberry Pi3*

ABSTRAK

Dalam penelitian ini, sensor visual yang digunakan dalam Jaringan Multimedia Sensor Nirkabel (WSMN) bertujuan untuk memantau dan menghitung jumlah orang yang lewat di dalam sebuah ruangan. Kontribusi dari makalah ini adalah penggunaan perangkat Raspberry Pi 3 yang terhubung ke internet menggunakan teknologi Internet of Things (IoT). Skema yang diusulkan dapat diterapkan di lingkungan nyata. Dari hasil pengujian, sistem dapat membedakan orang yang masuk dan keluar dari ruangan dengan memproses gambar menggunakan pengurangan latar belakang, dengan metode transformasi morfologis, dan menghitung area kontur gambar. Hasil pemrosesan gambar dapat menghitung jumlah orang di dalam ruangan, dan sistem dapat mengirimkan data tersebut ke server web. Selanjutnya, makalah ini membahas konsumsi energi yang digunakan oleh WSMN dan menjelaskan parameter pengujian.

Kata Kunci: *WMSN; Sensor Visual; Wi-Fi; Penghitungan Orang; Raspberry Pi 3*

Corresponding Author:
Kalvein Rantelobo,
Department of Electrical Engineering, FST,
Universitas Nusa Cendana,
Jl. Adisucipto, Penfui, Kupang, NTT.
Email: kalvein@staf.undana.ac.id

1. INTRODUCTION

Wireless Sensor Network (WSN) is a network that connects devices or terminals such as sensor nodes, routers, and sink nodes. This device is ad-hoc connected and supports multi-hop communication and even collaboration. The term ad-hoc refers to the ability of devices to communicate directly without the need for network infrastructures such as a router or access point. Multi-hop is a term that refers to the communication of several devices involving intermediate devices; multi-hop applies tools such as routers to forward packets from one node to another in the network. WSN is generally built from sensors with low, small, and low-power specifications so that they can be attempted in large areas in large quantities[1, 2]. Although small, this sensor can detect, process data, and communicate with sensors. WSN communicates between ad-hoc sensors and must be able to manage independently when communicating via wireless networks. The sensor is equipped with resources that can be used for a long time, so the operation does not require much human intervention. One WSN sensor is a camera's visual sensor. Visual sensors can send more information than scalar sensors, such as temperature, fire, and gas. With the application of camera sensors in WSN technology, the visual sensors can be placed in areas with little infrastructure, like minimal electricity resources and other conditions [3].

The most common obstacle associated with sensor network design is the operation of sensor nodes with limited energy resources. Each sensor node uses a battery as an energy source, which must be replaced or recharged (for example, using solar power) when it runs out; whether the battery can be renewed or not significantly affects the strategy applied to energy consumption. For non-rechargeable batteries, the sensor node must be able to operate until the desired time or the battery can be replaced. The length of time a node operates depends on the type of application. For example, monitoring glacial movements may require sensors that can work for several years, while battlefield sensors are needed for a few hours or several days. Energy efficiency is essential in wireless sensor networks. This requirement penetrates every aspect of the sensor node and network design. For instance, choices made in the sensor node's physical layer affect all devices' energy consumption.

In this paper, the WSN application will be developed using the Raspberry Pi 3 Mini PC platform for intelligent space applications. This application is designed to count the number of people entering and leaving the room, in this case, server space. It can send data to the internet with a Wi-Fi 802.11n module. The camera is installed to capture images for further processing. The object of the image was taken by a person/human. Data is processed to find out the number of people in the server room. This person counting system can be useful in implementing internet things for smart rooms or smart buildings, such as automation of room temperature based on the number of people in the room, room lamp automation, and room door automation can be done based on the capacity of people in the room.

2. RESEARCH METHODS

Many studies have been done on wireless sensor networks. In [4], some studies have examined human objects, such as video and images, based on moving objects. Our previous works by [5-7] implemented Linux-embedded OS on wireless sensor networks using visual and detection sensors. In this study, IEEE 802.15.4 ZigBee was used as a communication module, whereas IEEE 802.15.4 ZigBee had limitations in sending data to the internet, which requires the IEEE 802.11 protocol as a gateway. Research in [8] used the motion histogram method to identify and count people to save power consumption in image processing from visual sensors. Whereas in the study conducted by [9], Raspberry Pi was used as a camera data processing platform and used the histogram differences in imagery in counting people, but the testing in the study was limited to people who did not move and face facing the camera. The number of people in the study was obtained by adding face histograms to each person caught on camera. Based on several studies above, two crucial components can be identified in the WSN, namely aspects of data transmission and image processing. Both have an essential role in realising an effective and efficient WSN application. In [10], it was realised that a human counter that is based on the Atmega 16 microcontroller, an ultrasonic sensor that functions to display distance, is used as a data source in this study by comparing the distance value when people were passing and when there were no people, the number of people who enter the room, the results of the calculations were displayed on the LCD. The study in [11] aims to calculate the number of people in static and dynamic images with the Haar-like Features method capable of real-time face recognition. Testing was conducted on static images obtained by uploading image files and dynamic images captured from webcam cameras. The number of people in the image was obtained by detecting people's faces in the image. Based on

several studies above, two critical components can be identified in image processing on WSN, namely aspects of data transmission and image processing. Both of these things have an essential role and become the primary focus of this study. In this research, the implementation of the WSN application will be developed using the Raspberry Pi 3 mini PC platform for smart room applications [12].

The application is designed to calculate the number of people going in and out of a room, in this case, the server room. The application can send data to the internet with an 802.11n Wi-Fi module. The camera is installed to capture images for further processing. The object of the captured image is a person/human. Data is processed to find out the number of people in the server room. This person-counting system can be helpful in the implementation of the Internet of Things (IoT) for smart rooms or smart buildings, such as automating the room's air temperature based on the number of people in the room, automation room lights and automation of the room door can be done based on the capacity of the person in the place.

2.1 The Model Systems

Figure 1 shows the overall appearance of hardware visual sensor-based people calculation system devices within the scope of the wireless sensor network. The software takes several steps to calculate people and send calculation data to the web server. The first stage is the image inputting visual data; from the visual data received, several scenes are carried out to identify people in the image.

The morphological transformation method reduces the noise around the object to obtain a more definite object shape. This morphological transformation method utilises the functions of erosion and dilation on the image results from the previous stage. In general, erosion will reduce the size of objects in the image, while dilation functions to enlarge objects in the image. The morphology process is carried out after the image is converted into a black-and-white binary image using the thresholding method.

The next process is confirming the object's contours by adding the find contours function. Objects are defined by giving a minimum pixel area limit for each contoured purpose. So, if the thing caught is less than the pixel value specified, then the object is not a person. If a person is identified, the object will be tracked into or out of the room and calculated. Calculation results will be sent to the web server, which can be seen in Figure 1.

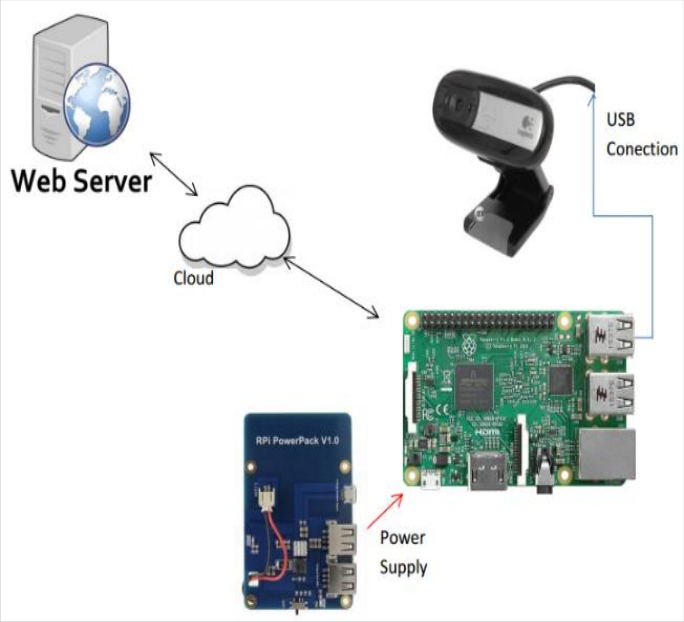


Figure 1. Scheme of the system

In Figure 1, there are three main components, namely:

- Logitech c170 Webcam Camera that functions as an input to this system. The data entered is visual data in the form of images that will be processed by Raspberry Pi 3.
- Raspberry Pi 3 is a data processor inputted by the camera so that the number of people entering and leaving the room has been generated and sent to the web server.
- Powerpack V1.2 RPi is a charging module for Lippo batteries 3.8 A, which will supply power to the Raspberry Pi.

Data is an important reference source used in this study. Two factors need to be considered in this research: data sources and data collection methods. The type of data used in this study is
divided into two, especially primary data and secondary data.

- Primary data, especially the captured image of the camera, the results of image processing, and the energy consumption of the device.
- Secondary data is data obtained from the Raspberry Pi 3 datasheet, camera datasheet, articles from the internet and ebooks that relate to the method of visual sensor-based people calculation systems and wireless sensor networks.

The flowchart of the research procedure in Figure 2 starts from the literature study stage in the form of theories supporting the realisation of this final project so that problems and identification of the hardware and software you want to use can be defined. After the hardware and software are realised, the device can be placed to collect and analyse data with the desired scenario. Next, the data collection phase of the desired scenario has been completed, and then the measurement of device energy consumption and conclusions are taken.

2.2 Overview of Software

This JSN scope calculation system uses IDLE Python 2.7.9 software with Python Language [13]. Python is a programming language that does not use compilers. This language is a simple dynamic language that is built based on a high level of code readability. The initial appearance of IDLE Python 2.7.9.
Besides having high code readability, so the code is easy to understand, the Python programming language has an extensive library, including the OpenCV library. OpenCV is an open-source computer vision library [14]. This library is written by discussing C and C ++ and can run on OS Linux, Windows, and MAC. OpenCV is designed for efficient computing and is focused on real-time applications. OpenCV provides a computer vision infrastructure that can quickly build image applications.
The software does several steps in calculating people and sending calculation data to the web server. The first stage is the image inputting visual data; from the visual data received, several stages are carried out to identify people in the image. Beginning with background and foreground separation with the background subtraction method, Background subtraction is a method that can separate the foreground and background in the image. The background or background in the image is constant or unchanging, while the foreground is a moving object. In this stage, there will be a difference between the objects that enter the scope of the image capture with the background.
The morphological transformation method reduces noise around the object to obtain a clearer object shape. This morphological transformation method utilises erosion and dilation functions in the results from the previous stage. In general, erosion will reduce the size of objects in the image, while dilation serves to enlarge objects in the image. The morphology process is carried out after the image is converted into black-and-white binary images using the thresholding method.
The next process is the process of confirming the contours of the object by adding some function in the software schema, which will be explained in the next section. Objects are defined by giving a minimum pixel area limit for each object that has been contoured. So, if the object that is caught is less than the pixel value specified, then the object is not a person. If a person is identified, the object will be tracked into the room or out of the room and calculated. Calculation results will be sent to the web server. The software flowchart can be seen in Figure 2.

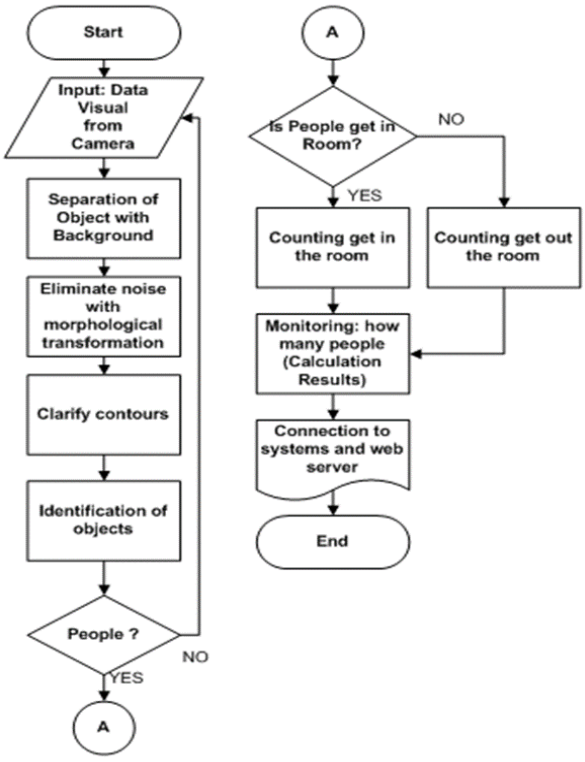


Figure 2: The Flowchart of the proposal schema

3. RESULTS AND DISCUSSION

3.1 Research Results

3.1.1 Hardware and Software Realisation

The overall appearance of hardware visual sensor-based people calculation system devices within the scope of the wireless sensor network that has been realised can be seen in Figure 3. In calculating people and sending calculation data to the web server, there are several steps that the software does. The first stage is the image inputting visual data from the visual data received,several scenes identify people in the image. To begin with, background and foreground separation with background subtraction method Background subtraction is a method that can separate the foreground and background in the image. The background of the image is constant or unchanging, while the foreground is an object that is resonant. In this stage, there will be a difference between the objects that enter the background capture scope and image.

Figure 3. Implementation of Scheme

The noise that appears around the object is reduced by the morphological transformation method to obtain a more definite object shape. This morphological transformation method utilises the functions of erosion and dilation on the image results from the previous stage. In general, erosion will reduce the size of objects in the image, while the dilation functions to enlarge objects in the image. The morphology process is carried out after the image is converted into a black-and-white binary image using the thresholding method.

The software does several steps in calculating people and sending calculation data to the web server. The first stage is the image inputting visual data from the visual data received, which is carried out in several scenes to identify people in the image. The opening with background and foreground separation with the background subtraction method. Background subtraction is a method that can separate the foreground and background in the image. The background of the image is constant or unchanging, while the foreground is an object that is resonant. In this stage, there will be a difference between the objects that enter the background capture scope and image.

The noise that appears around the object is reduced by the morphological transformation method to obtain a more definite object shape. This morphological transformation method utilises the functions of erosion and dilation on the image results from the previous stage. Generally, erosion will reduce the size of objects in the image, while the dilation functions to enlarge objects in the image. The morphology process is carried out after the image is converted into a black-and-white binary image using the thresholding method.

The next process is affirming the contours of the object by adding the function cv2.findContours. Objects were defined by giving a minimum pixel area boundary for each object that has been contoured. If the object that was caught is less than the specified pixel value, then the object is not a person. When people are identified, the object will be tracked until it enters the room or exits the room calculated. The calculation results will be sent to the web server.

Raspberry Pi 3 is a mini-computer that can execute complex programs. The specification of the Raspberry Pi 3 is shown in Table 1. In this case, the Raspberry Pi functions to run the person counting software and, at the same time as the IEEE 802.11n Wi-Fi module, sends the data from the person's calculation to the web server. The initial appearance of the Raspberry Pi can be seen in Figure 4. In Figure 4, you can see the Raspberry Pi display as if it were a computer in general. Raspberry Pi runs on this device on Linux-based Raspbian Jessie with Pixel OS. In Raspberry Pi, the IDLE Python 2.7.9 application has been installed. Python is a high-level programming language that supports the Internet of Things and image processing.

The Video Capture function can be used to take videos from video files or cameras. On testing, a video is taken with the camera. By using the cap = cv2. Video Capture ('0') in the program listing, the camera with ID 0 that has been connected to Raspberry Pi 3 starts capturing the image and stores it on the variable stamp.

The Video Capture function can be used to capture video from video files or cameras. In testing, the video was taken with the camera. By using cap =

cv2.Video Capture ('0') in the program listing, the camera with ID 0 that has been connected to the Raspberry Pi 3 starts capturing the image and is stored in the cap variable, as in Figure 4.

Table 1. Specification of the Rasberry Pi3[2]

Specifications	Data/Information
SoC	Broadcom BCM2837
CPU	4× ARM Cortex-A53, 1.2GHz
GPU	Broadcom VideoCore IV
RAM	1GB LPDDR2 (900 MHz)
Networking	10/100 *Ethernet*, 2.4GHz 802.11n *wireless*
Bluetooth	Blue tooth 4.1 *Classic*, Bluetooth *Low*
Storage	microSD
GPIO	40-pin header, *populated*
Ports	HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, *Ethernet*, *Camera Serial Interface* (CSI), *Display Serial Interface* (DSI)

The Video Capture function can be used to take videos from video files or cameras. On testing, a video is taken with the camera. By using the cap = cv2.Video Capture ('0') in the program listing, the camera with ID 0 that has been connected to Raspberry Pi 3 starts capturing the image and stores it on the variable stamp.
The Video Capture function can be used to capture video from video files or cameras. In testing, the video was taken with the camera. By using cap = cv2.Video Capture ('0') in the program listing, the camera with ID 0 that has been connected to the Raspberry Pi 3 starts capturing the image and is stored in the cap variable, as in Figure 4.

```
#Input camera

cap = cv2.VideoCapture('0')
```

Figure 4. Pseudocode of the function Video Capture for Input Camera (video streaming)

3.1.2 Separation of Object Images with Background (background Subtraction)

The separation of object images from the background begins with the segmentation section, where foreground objects or moving objects are segmented from the background. This is done by taking the picture as background and taking the frame obtained at t, denoted by I (t), to be compared to the background image symbolised by B. Each pixel I (t) is denoted as P [I (t)] and subtracted by each pixel in the same position in the background, denoted as P [B]. In accordance with the following formula.

$$P[F(t)] = P[I(t)] - P[B] \quad (1)$$

By threshing the results of the equation above, the formula is obtained:

$$| P [F (t)] - P [F (t + 1)] | > Threshold \quad (2)$$

Where P [F (t)] is each pixel in the frame that has been reduced by each pixel in the background, and P [F (t + 1)] is each pixel in the frame as a result of reducing the background in the previous t (time). Foreground pixels are moving pixels; therefore, each pixel on the frame in t was previously defined as background, as seen in Figure 5.

```
#Background Subtractor

fgbg = cv2.createBackgroundSubtractor
fgmask = fgbg.apply(frame)

#Implementation threshold on frame
```

Figure 5. Pseudocode of the Function Background Subtractor

The results of the cv2.createBackgroundSubtractor given in the next frame are loaded with the cv2.threshold, cv2.THRESH_BINARY function. With the binary threshold method, each pixel in the image is replaced by a black pixel (0) if the pixel intensity is less than the specified pixel limit or replaced with a white pixel (255) if the pixel intensity is more than the specified pixel limit. In the cv2.threshold program(fgmask, 20.255, cv2.THRESH_BINARY) there is an image input (fgmask), pixel intensity limit (20), and a maximum pixel limit (255).
The purpose of this process is to display the subtractor background results on objects captured by the camera by running the subtractor background function on the camera and displaying the results. This test is carried out in 3 conditions: when there are no objects passing by, when there are objects of people passing by, and when there are immovable objects caught on camera.
Figure 6 (a) is the catch of the image, and Figure 6 (b) is the subtractor background result in that image. From this test can be seen the background subtractor results in Figure 6 (b) is not much different from the background subtractor results in Figure 6 (b), which is done without objects. This proves that the subtractor background function cannot distinguish objects that do not move with the background,

highlighting the limitations of this approach in dynamic environments where stationary objects may be overlooked, leading to potential inaccuracies in object detection and tracking.

(a) (b)

Figure 6. Process the Background Subtractor in the image (a) Image Before the Background Subtractor and (b) After the Background Subtractor When There Are Passing Objects

It can be seen in Figure 6 (a) that the catch of the image with the moving object and Figure 6 (b) is the subtractor background result in the image. From these two images, the image of the background subtractor results in Figure 6 (b) displays a greyish-white pattern on a moving object that is caught on camera. This proves that the subtractor background function can distinguish moving objects from the background. The results of testing the separation of people's images from Background with Background Subtractor can be seen in Table 2.

Table 2. Test Result Separation of Image of Person from Background

Conditions	The Results of Background Subtractor	Information
There is no overall	Black object	All images identified as background
There is a moving object (person)	A greyish-white pattern on a moving object	Movable objects identified
There is a whole black immovable object	Black object	The immovable object has been identified as a background.

In other words, how does the background subtractor method work according to the formula $|P[F(t)] - P[F(t+1)]| > Threshold$ is by Figure 7.

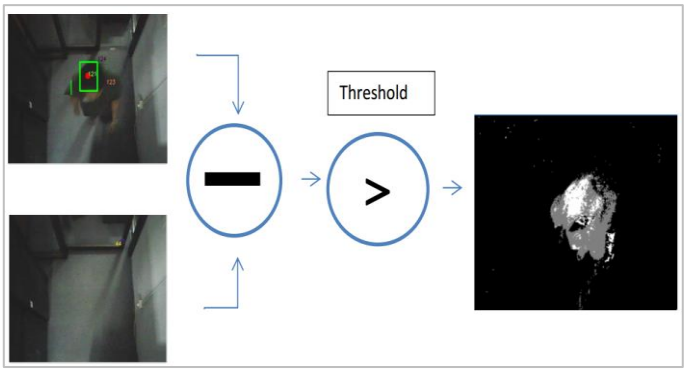


Figure 7. Process of the Background Subtractor

This morphological transformation method utilises erosion and dilation functions in black-and-white images resulting from the previous stage. Erosion is the process of combining object points into parts of background points, while dilation combines background points into parts of object points. Previously, the image has been separated by foreground and background with a background subtractor. The erosion process that is followed by the dilation is called an opening, while the dilation process followed by erosion is called the closing. The morphological transformation method requires two inputs, namely the image of the background subtractor and the kernel value for the opening and closing. This test aims to produce a noise-free image. The test method is done by entering various kernel values in the opening and closing and selecting the image results with the least noise, as seen in Figure. 8.

```
#Implementation threshold on frame
        try:
        ret,imBin=
cv2.threshold(fgmask,200,255,cv2.THRESH_BINARY)
        ret,imBin2                                          =
cv2.threshold(fgmask2,200,255,cv2.THRESH BINARY)
        #Opening
        mask = cv2.morphologyEx(imBin, cv2.MORPH_OPEN, kernelOp)
        mask2    =    cv2.morphologyEx(imBin2,    cv2.MORPH_OPEN,
kernelOp)
        #Closing
        mask   =      cv2.morphologyEx(mask   ,   cv2.MORPH_CLOSE,
kernelCl)
        mask2    =    cv2.morphologyEx(mask2,    cv2.MORPH_CLOSE,
kernelCl)
    except:
        print('EOF')
        print 'UP:',cnt_up
        print 'DOWN:',cnt_down
        break
```

Figure 8. Pseudocode of the Fucntion morphological transformation

In this method, several main functions are used, namely cv2.MORPH_OPEN and cv2.MORPH_CLOSE is used to execute the opening and closing method in the image according to the kernel value input, while the cv2.morphologyEx () function performs the morphological transformation using several inputs. The image of the subtractor background results in the previous step on the main, as well as the kernel value for opening and closing, are shown. Based on Fig. 6, it can be concluded that the value of the opening kernel 5 and closing 3, opening 9 and closing 5, opening 10 and closing 5, opening 9 and closing 6, opening 10 and closing 6 produces a good image that lacks noise and resembles the shape of the original object. The kernel values can be used to search for object contours at a later stage, as the pseudocode in Figure 9.

```
kernelOp = np.ones((9,9),np.uint8)

kernelCl = np.ones((6,6),np.uint8)
```

Figure 9. Pseudocode of Opening and Closing Kernel

Morphological transformation testing in the previous stage resulted in several values of opening and closing kernels that reduced noise in the image and produced a foreground shape that resembled the shape of the original object; the next step was to give a contour to the foreground. A contour is a line that combines all continuous points and has the same colour or intensity. Contours are used to identify objects that are caught on camera. Binary images are used to get a good contour value by applying the threshold to the image and morphological transformation. This test is done by trying each opening and closing kernel value that has produced a good image in the previous testing stage and adding the cv2.findContours function to the program listing, as seen in Figure 10. The best contour results will be used for the identification of the next object.

```
_,              contours0,              hierarchy              =
cv2.findContours(mask2,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
```

Figure 10. Pseudocode of Function findCountour

In the cv2.findContours function, some inputs are needed, namely the morphological transformation image in mask2, cv2.RETR_EXTERNAL, cv2.CHAIN_
APPROX_SIMPLE.cv2.RETR_EXTERNAL function to only use external object contours while cv2.CHAIN_APPROX_SIMPLE functions so that only 2 main coordinates on the right and left side of the object are used; this is done to reduce unwanted computation. The results of the cv2.findContours function can be seen in Figure 11.

3.1.3 Noise Removal with Morphological Transformation

From Figure 12, you can see the results of testing the cv2.findContours function in morphological transformation images with kernel opening values and closing 5 and 3. Where the cv2.findContours function can produce contours on objects entering and exiting the room, but on the contour of objects entering the range, there are still many lines with different contours of the original object.

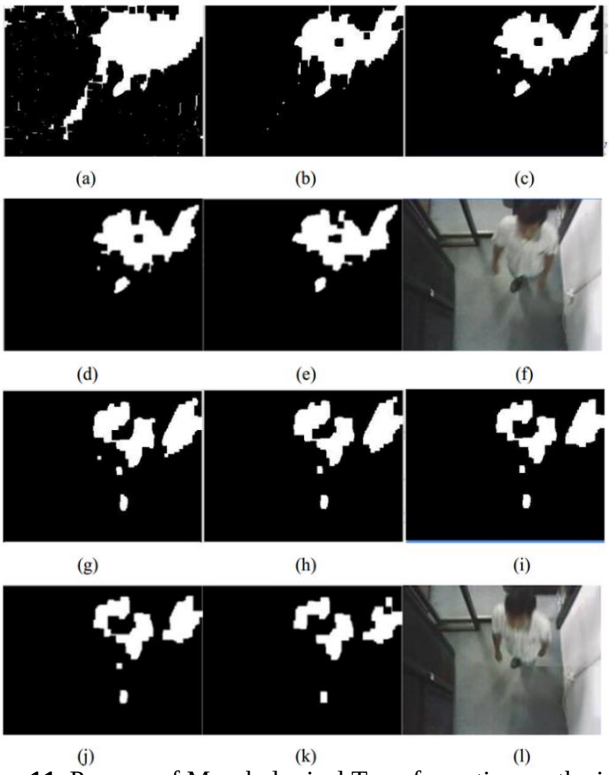
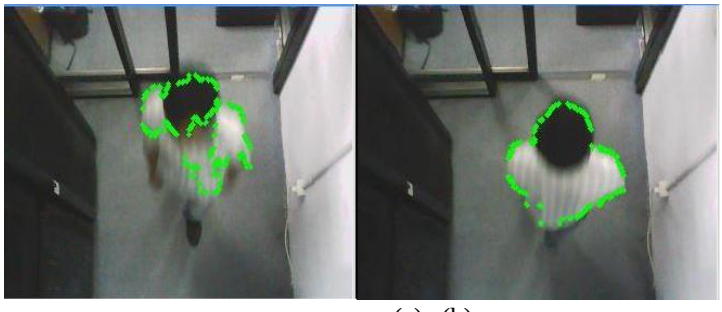


Figure 11. Process of Morphological Transformation on the image



(a) (b)

Figure 12. Process of Explaining Contours in Images (a) Image of object contours entering the room; and (b) an image of an object contours out of the room

3.1.4 Identification of Objects

Two stages of testing are carried out to distinguish objects that are people and non-people, specifically by looking for the area of the pixels of the object and giving a minimum area of pixels. The object tested at this stage is that the person who is not intact entered the scope of the camera, and the object of the person as a whole is included in the scope of the camera capture. By comparing the pixel area of the two objects, the minimum area of pixels is obtained for the whole object. So, if the pixel area of the object being caught is less than the specified pixel area, then the object is not a person who the camera has completely captured, which can be seen in Figure 14.

The minimum area of the pixel area is stored in the area TH variable. From the contour area obtained, 4 coordinate points of the object are drawn, and a green square is drawn on the object by adding program fragments. Functioncv2.boundingRect (cnt) dancv2.Moments (cnt) function to find 4 coordinate points on the object while the cv2.circle and cv2.rectangle function to describe the green square on the object, indicating that an object is a person. The above program results can be seen in Figure 14. In Figure 14 (a) and Figure 14 (b) can be seen as objects that enter and exit the room and are defined as people. It is proven by the appearance of a green square on the object. Whereas Figure 14 (c) shows a comparison of objects that have been defined by people and other objects in the form of chairs, it can be seen in the object seat that there is no green square; this proves that the system can distinguish objects of people and not people.

```
area = cv2.contourArea(cnt)
print 'area', area

areaTH = 1500

M = cv2.moments(cnt)
            cx = int(M['m10']/M['m00'])
            cy = int(M['m01']/M['m00'])
        x,y,w,h = cv2.boundingRect(cnt)
        cv2.circle(frame,(cx,cy), 5, (0,0,255), -1)
        img = cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
```

Figure 13. Pseducode area and taking four Object Coordinate Points

3.1.5 Calculation of People Entering and Exiting the Room

After object identification, then the calculation of people entering and exiting the room is then carried out. This test is done by tracking the defined object as a person until it passes through the virtual line drawn on the interface. At this stage, the number of people who entered and left the room was tested; this was done by running program fragments as in pseudocode in Figure 15. To track objects in and out of the room, first is done by giving the ID to each object that has been identified. In Figure 15, the object ID is called "person." Each person is tracked by matching the "person" contour in the next frame to the specified ID and saving the coordinates. Each "person" detected is given a condition if each person passes the virtual line that has been drawn with the cv2.polylines function as in listing pseudocode in Figure 15, the "person" is calculated in or out of the room. After identifying the object, then the calculation of the person entering and leaving the room is then carried out. It test was done by tracing an object that has been defined as a person to pass through the virtual line drawn on the interface.

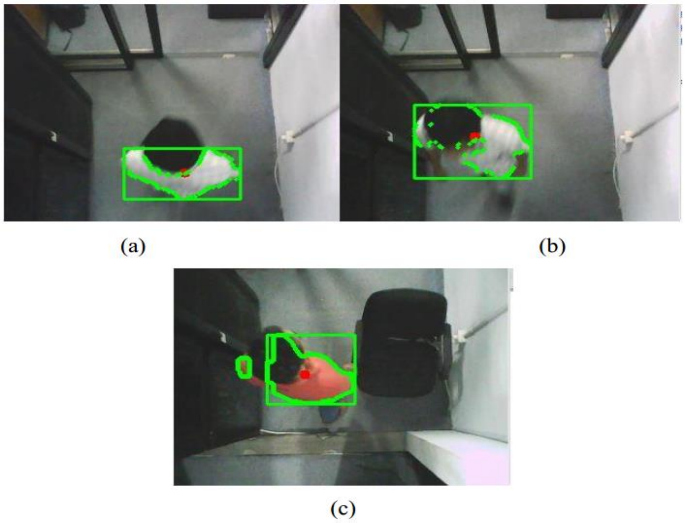


Figure 14. Object Identification Process: (a) Image of Identification Result of Room Entrance (b) Exit the Room and (c) Comparison of Identification of Person and Non-Person Objects.

There are two virtual lines used in the interface, which are blue and red. The incoming person will be known if the person's object passes the blue line and is followed by a red line. At the same time, the outgoing person will be known if the person's object crosses the red line and is followed by a blue line. After ascertaining that the object entered into the scope of the camera capture is a person, the calculation of the person entering and exiting the room is then performed as in Figure 16.

```
                    new = True
                    if cy in range(up_limit,down_limit):
                        for i in persons:
                            if abs(cx-i.getX()) <= w and
                            abs(cy-i.getY()) <= h:
                                new = False
                                i.updateCoords(cx,cy)
                            if i.going_UP(line_down,line_up) ==
                                True:
                                    cnt_up += 1;
                                    print "ID:",i.getId(),'crossed
going up at',time.strftime("%c")
                                elif
i.going_DOWN(line_down,line_up) == True:
                                    cnt_down += 1;
                                    print "ID:",i.getId(),'crossed
going down at',time.strftime("%c")
                                break
                            if i.getState() == '1':
                                if   i.getDir()   ==   'down'   and
i.getY() > down_limit:
                                    i.setDone()
```

Figure 15. Pseudocode of the object tracking and calculation of people entering and leaving the room

3.1.6 Transmission of Calculation Result of People Toward Web Servers

In the next stage, testing of data from the calculation of people goes to the web server. Data calculated by people who the device has obtained are sent to the website Thingspeak.com. Tests are carried out by running program fragments. The header, website, value variables, and communication ports are then inputted using the POST method, as seen in Figure 17. The results of the calculation of people in the room are sent in real-time.

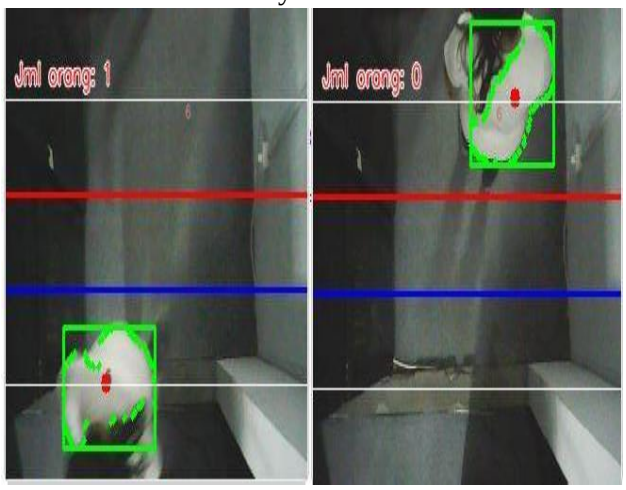


Figure 16. The Process of Calculating the Number of People Entering and Exiting the Room (a) Display of Calculation of People Entering the Room and (b) Exiting the Room

```
############################
    #Sending to the Webserver #
    ############################
    nilai = int((cnt_up)-(cnt_down))
    params = urllib.urlencode({'field1':value,'key':key })

#'field2': down,
    headers = {"Content-typZZe": "application/x-www-form-
urlencoded","Accept": "text/plain"}
    conn = httplib.HTTPConnection("api.thingspeak.com:80")
    try:
        #time.sleep(sleep)
        conn.request("POST", "/update", params, headers)
        response = conn.getresponse()
        print value
        print response.status, response.reason
        data = response.read()
        #time.sleep(sleep)
        conn.close()
    except:
        print "connection failed"
        break
```

Figure 17. Pseudocode is used to send people calculation results to a web server.

The data calculated by people who have been obtained by the device are sent to the website Thingspeak.com [15]. The results of the calculation of people entering and leaving the room are visualised in graphical form on the thingspeak.com website, as in Figure 18.

Figure 18. Graph Calculation of People Entering and Exiting the Room.

3.1.7 Device Energy Consumption Measurement

The energy consumption of the system for calculating people in the scope of wireless sensor networks can be determined by measuring energy consumption when images are captured, processed, and data is sent to a web server. To find out the device's energy consumption during image capture, image processing, and sending data to the web server, first, look for the power value by measuring the voltage and current ten times for 1 second at each of these stages and look for the average. The formula obtains the average voltage current and power values:

$$\bar{V} = \frac{\sum_{i=1}^{n} V_i}{n} \quad (3)$$

$$\bar{I} = \frac{\sum_{i=1}^{n} V_i}{n} \quad (4)$$

$$P = V \times I \quad (5)$$

where: $\bar{V}$: mean of voltage; $\bar{I}$: mean of current; Vi : values of voltage in i; Ii: value of current in i; n : sampling; and P: power in watt.

Device energy consumption is obtained by summing energy consumption during image capture, image processing, and sending data to a web server, according to the formula:

$$E_{WVSN} = E_{capt} + E_{proc} + E_{comm} \quad (6)$$

Where: $EWVSN$: Energy consumption of the device;$Ecapt$: Energy consumption when the device captures the image; $Eproc$: Energy consumption when the device processes the image; and Eco: Energy consumption when the device sends data to the web server.

From the measurements in Section 3.1.7, the power consumption is 0.905 W when the device captures an image, 1,707 W when the device processes the image, and 0.65 W when the device sends data to the web server. With 1 W unit conversion equal to 1 J / s, it is obtaining:

$$EWVSN = Ecapt + Eproc + Ecomm$$

$$EWVSN = 0.905 + 1.707 + 0.65$$

$$EWVSN = 3{,}262\ J\ /\ s$$

So, the energy consumption of the system for calculating people within the scope of the wireless sensor network is 3,262 s / s.

3.2. Discussion

The visual sensor-based people counting system discussed in this article emphasises the effective use of image processing technology to quantify the number of individuals within a given space. Utilising Raspberry Pi, the system processes images in real-time and transmits data via a wireless sensor network, thereby contributing to the advancement of Iot-based automation systems. The background subtraction method is employed to effectively distinguish between the background and objects, although challenges arise in certain scenarios where stationary objects may be erroneously identified as part of the background. Furthermore, morphological transformations are utilised to reduce noise in the

images, with the careful selection of appropriate kernels being crucial for generating clear imagery. Object identification is achieved through contour detection, enabling the system to differentiate human figures from other irrelevant objects; counting is conducted through the implementation of virtual lines, which enhances the accuracy of detecting individuals entering and exiting the space. The resulting count data is transmitted to a web server in real-time and visualised on the Thingspeak.com platform, assisting users in monitoring the acquired information.

4. CONCLUSION

In building an image-based application, the system needs to capture and identify good objects. Some initial processes (pre-processing) are required in order to obtain good image results to increase application performance. This process is the process of separating objects with background, morphology, and contour confirmation. The process of separating objects from the background is done so that only parts of the object can be obtained for the sake of analysis. The results of this process still leave noise that is eliminated by the morphology process (opening and closing). The opening and closing kernel values that give the best results are 9 × 9 and 6 × 6.

Virtual help is used to determine whether an object (person) caught on camera moves into or out of the room. The movement of objects relative to the virtual line determines the person's status (in / out). The decimation of the results of this application is done by utilising a web service called Thingspeak. Thus, the results of monitoring carried out by this application can be accessed from the internet.

Some suggestions for the development of visual sensor-based people calculation systems in the scope of the wireless sensor network further are as follows. (a) The results of calculations of people in the room can be developed to automate the state of the room. Like temperature settings and door automation; and (b). Image processing can be refined to identify the faces of people entering and leaving the room, improving the function of the device as room security.

ACKNOWLEDGMENTS

The research is supported through DP2M Grants by Indonesian Ministry of Research and Higher Education (under Grant No. 195/SP2H/LT/DRPM/2019). We are very grateful to Mr. Meksianis Ndii for fruitful discussion; Lab. Teknik Elektro FST Universitas Nusa Cendana; UKM Robotec Udayana; and Kelompok Studi Robot Fakultas Teknik Universitas Udayana.

REFERENCES

[1] S. Birtane, O. K. Sahingoz, and H. Korkmaz, "Vibrational genetic algorithm-based deployment of wireless sensor networks with heterogeneous nodes in irregularly shaped areas," *IEEE Access*, 2024, doi: https://doi.org/10.1109/ACCESS.2024.3395421.

[2] S. Lata, S. Mehfuz, and S. Urooj, "Secure and reliable WSN for Internet of Things: Challenges and enabling technologies," *IEEE Access*, vol. 9, pp. 161103-161128, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3131367.

[3] Z. Nurlan, T. Zhukabayeva, M. Othman, A. Adamova, and N. Zhakiyev, "Wireless Sensor Network as a Mesh: Vision and Challenges," *IEEE Access*, vol. 10, pp. 46-67, 2022, doi: https://doi.org/10.1109/ACCESS.2021.3137341.

[4] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: A survey," *Pattern Recognition*, vol. 51, pp. 148-175, 2016, doi: https://doi.org/10.1016/j.patcog.2015.08.027.

[5] N. P. Sastra and G. Hendrantoro, "Energy Efficiency of Image Transmission in Embedded Linux based Wireless Visual Sensor Network," *Journal of Communications Software and Systems*, vol. 11, no. 3, pp. 146-154, 2015, doi: http://dx.doi.org/10.24138/jcomss.v11i3.103.

[6] N. P. Sastra, W. Wirawan, and G. Hendrantoro, "Virtual view image over wireless visual sensor network," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 9, no. 3, pp. 489-496, 2011, doi: http://doi.org/10.12928/telkomnika.v9i3.740.

[7] D. M. Wiharta and G. Hendrantoro, "On the Accuracy of Particle Filter-Based Object Tracking," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 11, pp. 265-276, 2015, doi: https://doi.org/10.14257/ijmue.2015.10.11.25.

[8] T. Teixeira and A. Savvides, "Lightweight people counting and localizing for easily deployable indoors wsns," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 4, pp. 493-502, 2008, doi: https://doi.org/10.1109/JSTSP.2008.2001426.

[9] B. Hemangi and K. Nikhita, "People counting system using raspberry pi with opencv," *International Journal for Research in Engineering Application & Management (IJREAM), ISSN*, pp. 2494-9150, 2016. [Online]. Available: https://www.ijream.org/papers/IJREAMV02I01894.pdf.

[10] D. I. S. Saputra, "Rancang Bangun Alat Penghitung Jumlah Pengunjung di Toko Adhelina Berbasis Mikrokontroler Atmega 16," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 4, no. 1, pp. 16-21, 2015. [Online]. Available: https://media.neliti.com/media/publications/265926-rancang-bangun-alat-penghitung-jumlah-pe-38a1e6b1.pdf.

[11] R. Kusumanto, W. S. Pambudi, and A. N. Tompunu, "Aplikasi Sensor Vision untuk Deteksi MultiFace dan Menghitung Jumlah Orang," *Semantik 2012*, pp. 1-8, 2012. [Online]. Available: https://publikasi.dinus.ac.id/index.php/semantik/article/view/45.

[12] R. Pi, "Raspberry pi-teach, learn, and make with raspberry pi," *línea]. Available: https://www. raspberrypi. org/[Último acceso: 20 Agosto 2016]*, 2015.

[13] python. [Online]. Available: https://www.python.org/downloads/

[14] Lbrary *opencv*. Accessed: 04-Sep-2018. [Online]. Available: https://opencv.org

[15] IoT *Analytics - ThingSpeak Internet of Things*. [Online]. Available: https://thingspeak.mathworks.com/