NON INVASIVE INFANT SLEEP APNEA DETECTION

CO321 CO324 CO325 UNIFIED PROJECT

TEAM

E/14/158: GIHAN JAYATILAKA
E/14/339: SUREN SRITHARAN
E/14/379: HARSHANA WELIGAMPOLA
E/14/237: PANKAYARAJ PATHMANATHAN

Deaprtment of Computer Engineering
Unviersity of Peradeniya

2018

Contents

List of Figures

List of Tables

1. Introduction

1.1 What is sleep Apnea

Sleep Apnea is a serious disorder caused by the interruption of breathing during sleep. This can cause the people to stop breathing for several time even hundreds if not treated properly. It can affect people of any age. But when the babies are affected with the condition they tend to not get up and keep on sleeping which may risk their lives. There are two types of sleep apnea

1. Obstructive sleep apnea
2. Central sleep apnea

1.1.1 Obstructive Sleep Apnea

This is caused by the partial or complete halt of airway despite the ongoing efforts to breathe. When the muscles in the throat relaxes causing the soft tissue in the back of the throat to collapse blocking the airway. This results in partial or complete pauses of breathing which may last at least up to 10 seconds. Most commonly it lasts from 10s to 30s.Brain responds to the causing an arousal from the sleep thus returning the breathing back to the normal state. If this pattern can happen for hundreds of times during the sleep causing a fragmented sleep. Most people with the condition will snore loudly or frequently with periods of silence when air flow is reduced.

OSA can happen to any age group but is very common with middle and older age. About 80 perccent of the people with OSA remains . The OSA is measured by AHI(apnea hypopnea index). This is the average apneas and hypopneas that happens during an hour of sleep. Based on that it can be categorized into three namely mild OSA(5 - 15 AHI), moderate OSA (15 - 30 AHI), severe OSA (more than 30 AHI).

Risk groups

- Overweight
- Males with long necks
- People with abnormalities on tissues in neck
- Adults and children with Down Syndrome
- Children with large tonsils

Effects

- Fluctuating oxygen levels
- Increased heart rate
- Chronic elevation on daily blood pressure
- Mood changes
- Impaired concentration
- Increased risk of strokes

1.1.2 Central Sleep Apnea

This is caused by the breathing interruption caused by the way brain functions. In this case the brain ceases to try breathing. This happens because the brain fails to send signals to muscles. This may also occur by existing conditions such as heart failure or stroke. So treatment is done by treating the existing condition and also by using a device to assist breathing or to supplement oxygen.

Symptoms

- Abnormal breathing patterns
- Abrupt awakening with a short breath
- Insomnia
- Hypersomnia
- Difficulty in concentration
- Mood changes
- Morning headaches
- Snoring

Causes

Central Sleep Apnea can occur by many conditions that affects the ability of the brain-stem. Cause varies with the type of central sleep apnea.

- Cheyne-Stokes breathing - this is associated with heart failure and stroke. Characterized by gradual increase and decrease in breathing effort.
- Drug-induced apnea - Taking certain medications such as opoid or codeine sulphate may cause breathing to become irregular
- High-altitude periodic breathing - this occurs when you are exposed to very high altitude

1.2 Available solutions

1.2.1 Pulse oxymetry

Pulse oxymetry is a technique of measuring the oxygen concentration of the blood. If a child is suffering from OSA his/her blood oxygen level drops down suddenly. Sometimes the blood oxygen drop could be a result of some other problem as well. But still, since the biggest problem with OSA is the reduction of the oxygen supply to the bran, pulse oxymeters are a good enough OSA detection technique.
Pulse oxymeters should be connected to the skin with underlying veins present. Usually, they are connected to the fingers in adults. But for infants, they are connected to the earlobe.

1.2.2 Acoustics

Acoustic techniques used to detect OSA consists of microphones and sound processing. The microphones tries to sense the sound of the breathing of the infant. These sound signals are then processed to identify anomalies in the pattern.
The disadvantage of this technique is that the sound of breathing is of low strength and therefore the noise in the background can make it very difficult to analyze the breathing sound.

1.2.3 Video processing

There are several published work on diagnosing OSA with video processing.

- **Noninvasive Monitoring System for Early Detection of Apnea in Newborns and Infants - Querétaro State University, UAQ Querétaro, México**
  The approach is to find the rate of breathing by the fourier tranform of the

time series of the grey scale values of a set of pixels. This algorithm is sufficient for cases where the only movement in the video is breathing. External disturbances will cause wrong results for this algorithm.

- **Automated Detection of Newborn Sleep Apnea Using Video Monitoring System - Indian Institute of Technology, Kharagpur, India**
  The algorithm is based on keeping track of the total **intensity** of a region of interest. The problem with this algorithm is that it does not take into account any actual shape of the infant. The intensity of the region can change drastically with lighting condition changes. These changes are taken as false positives in the algorithm.

1.3 Our solution

We propose a non invasive solution based on video processing. The infant is observed by a video camera which is connected to a single board computer (Raspberry pi) which analyzes the video feed to diagnose breathing anomalies. The camera is turned to a proper orientation for the observation using a robotic arm.

1.3.1 Advantages of our solution

- Our solution is 100% non intrusive.
- The camera node could operate independently with the inbuilt battery cells. There is no need even for internet connection for the primary functionality.
- Our camera can automatically detect the baby and orient itself to the correct position intelligently using the robotic arm.
- The *breathing detection algorithm* is automated than the previous work found. (The existing algorithms require human intervention to detect interesting regions etc:)
- The *breathing detection algorithm* is accuracte than the existing algorithms.

2. Implementation of the solution

2.1 Overview

The solution consists of hardware and software components as per following figures.

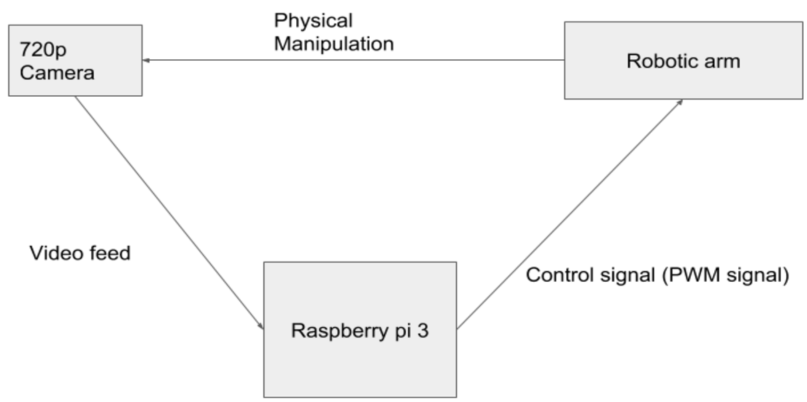

Figure 2.1: Hardware block diagram

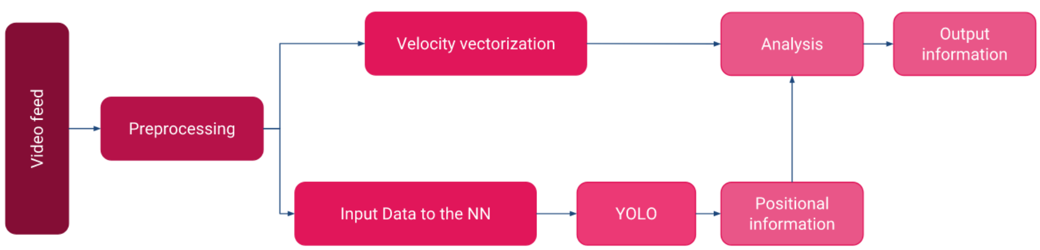

Figure 2.2: Software block diagram

2.2 Hardware

The project consists of two hardware components – The embedded device which is deployed and the main server.

2.2.1 Embedded device

2.2.2 Server

The server used is as following.

- $URL : server2.teambitecode.com$
- $Processor : Intel^{(R)} Xeon^{(R)} CPU E3-1270 V2 @ 3.50GHz$
- $Memory : 512MB$

2.2.3 Node

The node consists of the Raspberry pi, camera and the robotic mount.

Raspberry pi

RASPBERRY PI 3 MODEL B -Single-board computer with wireless LAN and Bluetooth connectivity.

Specifications

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- CSI camera port for connecting a Raspberry Pi camera

Why Raspberry pi?

- The solution requires a high definition video input. Raspberry pi has a native plug and play HD camera.
- The video processing requires a powerful processor and Raspberry pi has a quadcore processor and 1GB of RAM.
- The algorithm requires several libraries to run on. There are functioning compilers to compile these libraries to the ARM architecture of Raspberry pi.

Camera

The Raspberry Pi Camera Module v2 replaced the original Camera Module in April 2016. The v2 Camera Module has a Sony IMX219 8-megapixel sensor.
It supports 1080p30, 720p60 and VGA90 video modes, as well as still capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi.

The 720p mode is used in our project.

2.3 Software

The system uses linux based operating systems, python based numerical computation packages and communication systems.

2.3.1 Server

OS	Ubuntu
Webserver	Apache, Dijango
MQTT Broker	Mosquitto

Table 2.1: Server software

Ubuntu

Ubuntu 16.04.1 LTS is running on the server. A command line version of the operating system is installed.

Why Ubuntu? Ubuntu is one of the most stable distributions of linux suitable for entry level users. The LTS version used in the project will be supported by Cannonical Inc. and the community over a long period of time.
Since we use the OS with only the shell UI, it consumes very little resources of the server. This allows us to cater more users simultaneously.

Django

Django a python based web framework library is chosen as the server side programming language.

Why Django?

- Since the project is meant for a smaller time line the usage of Django gives an edge when it comes to productivity

- Since all the data analysis algorithms are in python it becomes easier to integrate them with the backend development code
- The availability of fully tested functions helps purpose of designing the web page.
- Mistakes such as SQL injection, cross site request forgery and click jacking can be avoided without taking any dedicated actions. Thus it increases the security aspect of the server side with less effort.
- The availability of enough debugging tools for python makes the testing phase easier

Mosquitto

Eclipse Mosquitto is an open source message broker that implements the MQTT protocol. Mosquitto is lightweight and is suitable for use on all devices from low power single board computers to full servers.

Why Mosquitto? The MQTT protocol provides a lightweight method of carrying out messaging using a publish/subscribe model. This makes it suitable for Internet of Things messaging with low power.
Furthermore it supports other security features such as client authorization, ACL (Access control list) and TLS.

2.3.2 Node

The software packages running on the node are,

OS	Raspbian
Interpreter	Python 2.7
Numerical packages	Numpy Tensorflow
Video handling	OpenCV
Communication	PahoMQTT

Table 2.2: Node software

Raspbian

This is the native OS for Raspberry pi.

Python 2.7

Python was chosen as the main programming language to implement the algorithm because,
1.Python syntax is simple enough to implement any mathematical algorithm.

2.Numerical computation packages like numpy and tensorflow has interfaces to fasten up the computation.

This is not the latest version of python. But both tensorflow and opencv has better interfaces with python 2.7.

Numpy

A python based library for data manipulation and numerical computation was used in the data analysis part due to the following advantages.

- It is a package on continuous development. So in the long run the matter of dealing with syntax can result in a better performance
- It's a rigorously tested, well maintained and widely used library for computation. So the community support available plays a huge advantage in the designing procedure.
- Simpler syntax enables high productivity with minimal code
- Provides better and stable performance even in the light of bigger matrices.

2.4 Algorithms

Our solution consists of a novel algorithm for detecting the breathing pattern of the baby. In addition the artificial intelligence algorithms are used to identify the baby and turn the camera for a good orientation.

2.4.1 Detecting the baby

Given different lighting condition the detection of baby by the camera becomes the most primitive task. Developing an object detection algorithm by hand seemed an inefficient way. Because an object detection algorithm must be something of a continuous development. Due to the limited availability of resources we have decided to make use of an existing library YOLO- Real time object detection api. It in real time gives detects objects with an confidence interval thus enabling us to know the availability and location of the baby.

2.4.2 Detecting the breathing pattern

The algorithm we propose have several steps.

- The video is taken in as an array of 8 bit unsigned integers.

Figure 2.3: The original video

- The video is fed to the **Canny edge detection algorithm**
  The opencv implementation of this algorithm is used. The result is a black and white video stream on where the white corresponds to the edges and black to the rest.

$$E_{(t)H\times W} \text{ is the edge matrix.}$$

$$E_{(t,x,y)} = \begin{cases} 1 \text{ if } E_{(t,x,y)} \text{ is an edge.} \\ 0 \text{ if } E_{(t,x,y)} \text{ is not an edge.} \end{cases}$$

Figure 2.4: The edges of the video

- The region of interest A_0 is chosen.
  $(x, y) \in A_0$
  $x \in [x_0, x_1], y \in [y_0, y_1]$
  As of now our algorithm requires a manual input for this region.
  We hope to automate this parameter during the time from 3rd phase evaluation to the final evaluation.

Figure 2.5: The region of interest

- The centroid $C_0(t)$ of the edges in A_0 is calculated for every t
  $C_0(t) = (x_{c_0(t)}, y_{c_0(t)}))$

$$x_{c_0(t)} = \frac{\sum_{(x,y)\in A} E(t,x,y) \times x}{\sum_{(x,y)\in A} E(t,x,y)}$$

$$y_{c_0(t)} = \frac{\sum_{(x,y)\in A} E(t,x,y) \times y}{\sum_{(x,y)\in A} E(t,x,y)}$$

  Special case:
  $(x_{c_0(t)}, y_{c_0(t)}) = (\frac{x_0+x_1}{2}, \frac{y_0+y_1}{2})$ when

$$\sum_{(x,y)\in A} E(t,x,y) = 0$$

- Then the velocity of the centroid $\underline{v}_{(t)}$ is calculated by,
  $\underline{v}_{(t)} = (x_{c_0(t)} - x_{c_0(t-1)})\underline{i} + (y_{c_0(t)} - y_{c_0(t-1)})\underline{j}$

- The direction along which the velocity of the centroid $\underline{v}_{(t)}$ lie is calculated using the **Principle component analysis** as follows,
  Write $v_{(t)}$ as a row vector

$$v_{(t)} = \begin{pmatrix} \underline{\mathrm{v}}_{(t)}.\underline{i} & \underline{\mathrm{v}}_{(t)}.\underline{j} \end{pmatrix}$$

$$v_{(t)} = \begin{pmatrix} \mathrm{v}_{x(t)} & \mathrm{v}_{y(t)} \end{pmatrix}$$

  Make a matrix by taking 10 such readings and arranging them as rows,

$$V_{(t)} = \begin{pmatrix} \mathrm{v}_{x(t)} & \mathrm{v}_{y(t)} \\ \mathrm{v}_{x(t-1)} & \mathrm{v}_{y(t-1)} \\ \mathrm{v}_{x(t-2)} & \mathrm{v}_{y(t-2)} \\ .... & ... \\ ... & ..... \\ \mathrm{v}_{x(t-9)} & \mathrm{v}_{y(t-9)} \end{pmatrix}$$

  Find the mean of these rows,

$$\overline{v_{x(t)}} = \frac{1}{10}\sum_{i=0}^{9} v_{x(t-i)}$$

$$\overline{v_{y(t)}} = \frac{1}{10}\sum_{i=0}^{9} v_{y(t-i)}$$

  Then find the difference matrix,

$$D_{(t)} = V_{(t)} - \overline{V_{(t)}} = \begin{pmatrix} \mathrm{v}_{x(t)} - \overline{v_{x(t)}} & \mathrm{v}_{y(t)} - \overline{v_{y(t)}} \\ \mathrm{v}_{x(t-1)} - \overline{v_{x(t)}} & \mathrm{v}_{y(t-1)} - \overline{v_{y(t)}} \\ \mathrm{v}_{x(t-2)} - \overline{v_{x(t)}} & \mathrm{v}_{y(t-2)} - \overline{v_{y(t)}} \\ .... & ... \\ ... & ..... \\ \mathrm{v}_{x(t-9)} - \overline{v_{x(t)}} & \mathrm{v}_{y(t-9)} - \overline{v_{y(t)}} \end{pmatrix}$$

  The covariance matrix $C_{(t)}$ is calculated by,

$$C_{(t)} = D_{(t)}^{T}.D_{(t)}$$

  $C_{(t)}$ is decomposed into
  $C_{(t)} = P_{(t)} D_{(t)} P_{(t)}^{-1}$ using eigen value decomposition.

$$P_{(t)} = \begin{pmatrix} \mathrm{w}_{1x(t)} & \mathrm{w}_{2x(t)} \\ \mathrm{w}_{1y(t)} & \mathrm{w}_{2y(t)} \end{pmatrix}$$

  and

$$D_{(t)} = \begin{pmatrix} \lambda_{1(t)} & 0 \\ 0 & \lambda_{2(t)} \end{pmatrix}$$

Here the $P_{(t)}$ has the eigen vectors,

$$\underline{w}_{1(t)} = w_{1x(t)}\underline{i} + w_{1y(t)}\underline{j}$$
$$\underline{w}_{2(t)} = w_{2x(t)}\underline{i} + w_{2y(t)}\underline{j}$$

$D_{(t)}$ has their corresponding eigen values $\lambda_{1(t)}$ and $\lambda_{2(t)}$

The bigger value of $\lambda_{1(t)}$ and $\lambda_{2(t)}$ is chosen (let it be $\lambda_{1(t)}$) and the corresponsing eigen vector $\underline{w}_{1(t)}$ gives the direction of the breathing.

The unit vector along this direction is calculated for the next steps,

$$u(t) = \frac{\underline{w}_{1(t)}}{\|\underline{w}_{1(t)}\|}$$

- Now we have $\underline{u}(t)$ and $\underline{v}_{(t)}$. Projecting the velocity vector in the unit vector of direction gives a scalar parameter $s_{0(t)}$ that can be used to determine breathing.

$$s_{0(t)} = \underline{u}(t).\underline{v}_{(t)}$$

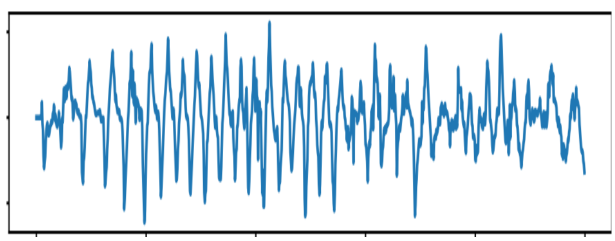

Figure 2.6: $s_{0(t)}$

- $s_{(0)t}$ undergoes two smoothing techniques to give,

$$s_{1(t)} \leftarrow \text{ adaptive filtered } s_{0(t)}$$
$$s_{2(t)} \leftarrow \text{ gaussian blurred } s_{1(t)}$$

$$s_{1(t)} = s_{0(t)} \times 0.8 + s_{0(t-1)} \times 0.2$$

Gaussian blurring is done with 15σ radius.

- The graph $s_{2(t)}$ - t looks as follows,

- The peaks are found using the technique,

$$s_{2(t)} > s_{2(t-1)} \text{ and } s_{2(t)} \le s_{2(t+1)} \Rightarrow s_{2(t)} \text{ is a peak.}$$

  The peaks are marked as following.

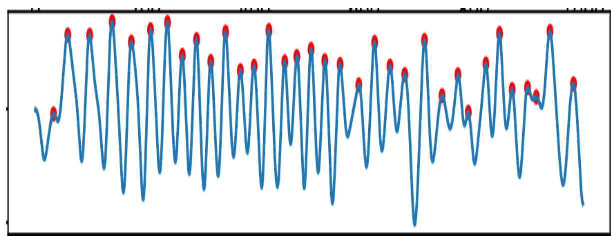

Figure 2.7: $s_{2(t)}$ with peaks marked

- The breathing intervals are calculated as the time between two peaks.

Breath number	Time for cycle / (ms)
1	1200
2	1230
3	1050
4	1050
...	...
...	...
...	...
...	...

Table 2.3: Breathing time interval report

2.5 Implementation

2.5.1 Device

Configuration Mode

Initially is device must be configured for proper performance. Configuration compromises of two steps.

- Camera orientation -
  The orientation could either be done manually or using the automatic orientation algorithm. The method (automatic/manual could be specified in the configuration file.

- Communication parameters -
  The configuration file contains other parameters(User name , Device ID, Access token and other settings) required for MQTT communication as well.

The parameters within the configuration file must be configured for proper operation

Operation Mode

In the operation mode the device reads the image, processes it, and sends processed information to the server. The following steps are repeated :

- Obtain the video feed from the raspberry pi camera.
- Extract image frames from video feed.
- Process image using aforementioned algorithms.
- Convert processed information into json format
- Encrypt json text using AES.
- Publish encrypted message to particular topic as MQTT message.

The algorithm outputs periodical information. This includes a timestamp (specifies starting point), breathing pattern (array of values) and an analysis of the pattern (risk of apnea).

2.5.2 Server

Mosquitto (MQTT broker)

The broker uses an authorization plugin which authenticates the device and identifies whether it has permission to publish - subscribe to a file. Initially the broker checks the whether the user-name-password is valid. Once the client has been authenticated, the broker checks whether the client has permission to publish - subscribe to the requested topic. If the client has permission the action takes place. Note that the broker uses TLS to ensure that the password is secure. Furthermore the broker logs other details such as the time at which connection - disconnection occurs, invalid - unauthorized connection attempts, etc. All these information are pushed to a client that runs within the django framework.

Django

The django framework perform 3 major tasks:

1. Receive data
2. Database
3. Web service

2.5.3 Database

Despite the use of the SQL-LITE database that comes with django by default on deployment a better database is essential. Thus MySQL is used as the database.

2.5.4 Web service

The front end includes many widgets such as

- Graphs - Figure 2.3, 2.4
- Analysis reports - Table 2.3
- Device status information - Battery level, signal strength.
- Patient information - These are manually added on setup of a device

Django provides these contents upon the request of the client by querying the database. Furthermore it notifies the user in real time upon events such as new connections, disconnections, apnea alerts, etc.

3. Project Information

3.1 Budget

No.	Item	Unit Price /(LKR)	Quantity	Total
1.	Raspberry Pi 3 Model B	10,000.00	1	10,000.00
2.	Raspberry Pi Camera	800.00	1	800.00
3.	Servo motors	400.00	2	800.00
Total				11,600.00

Table 3.1: Budget

3.2 Time line

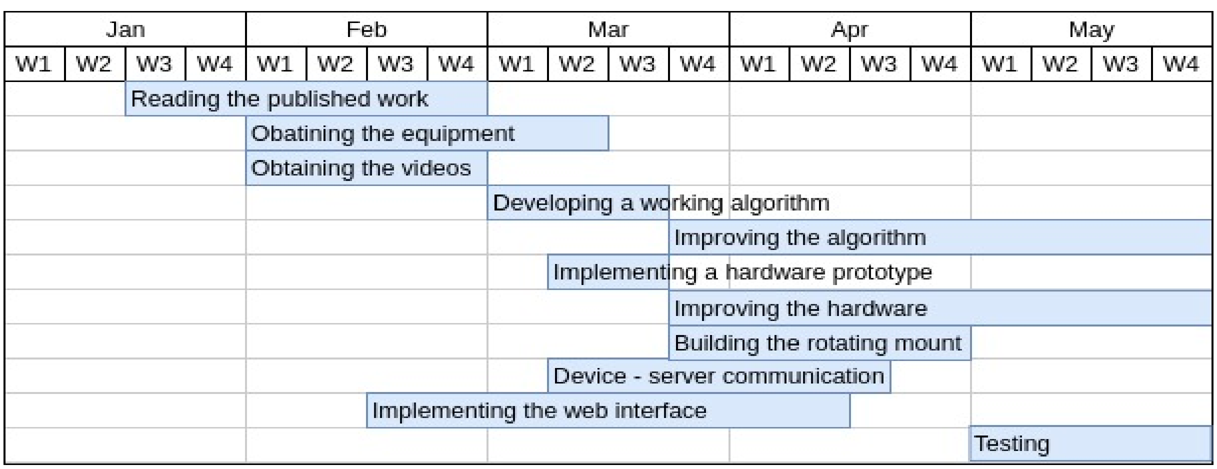

Table 3.2: Timeline of the project

END