



Proceedings of the Asia-Pacific Advanced Network 2014 v. 38, p. 99-114.

<http://dx.doi.org/10.7125/APAN.38.14>

ISSN 2227-3026

Design and Development of a Remote Station for Real-time Monitoring of Urban Flooding

Alvin E. Retamar*, Felan Carlo C. Garcia , Jasmin Jane M. Yabut, Joven C. Javier

Solutions and Services Engineering Division, Advanced Science and Technology Institute
ASTI Bldg., CP Garcia Ave. University of the Philippines Technology Park,
Quezon City, Philippines

E-mail: ning@asti.dost.gov.ph

* Author to whom correspondence should be addressed; Tel.: +632 426 9764

Abstract:

This paper presents the design and development of urban flood monitoring stations that use pressure sensor to determine flooding levels. GPRS and SMS were used to communicate data from the remote stations to a server located at a data center. A web-based visualization tool has been developed to allow access to data in real-time. Based on experiments, the sensor measurements have a difference of 0.872cm to 3.067cm with actual values. The higher differences tend to be associated with higher water levels while lower differences were noted for lower water levels. With further experimentation, these differences can be used as correction factors to get to a more accurate reading, specially for purposes of R&D or modeling, though for issuance of warnings to the public, these differences are small enough especially since warnings are mostly based on qualitative descriptions such as knee-deep or waist deep flood level. Image capture was also experimented on and is envisioned to be incorporated with the station to provide a means to verify water level by providing a visual reference. Likewise, the image capture device will serve as a redundant sensor that can also determine flood levels thru image processing. Currently, two prototypes are deployed along Espana Avenue, an area known for flooding. Data from the stations show the typical behavior of flood water but currently, there is no reliable means to verify the accuracy of sensor readings, until the image capture device is installed. The data are viewable thru a web-based visualization tool.

Keywords: Urban flooding, Flood monitoring, Image processing

1. Introduction

Flooding in Metro Manila is of special interest because it is where business, workers, and residents tend to concentrate. When flooding happens, traffic follows, commuters get stuck and complain, and Metro Manila essentially becomes paralyzed. To date, the Metropolitan Manila Development Authority (MMDA) has identified 183 flood-prone streets in Metro Manila.

In view of this, the Department of Science and Technology (DOST) has funded a project titled “Flood Sensor Development, Installation, and Monitoring of Urban Flooding in Metro Manila”. It is being implemented by the Advanced Science and Technology Institute (ASTI), a research and development agency under the DOST with primary focus on Information and Communication Technologies (ICT) and Electronics. Primarily, the project aims to develop remote stations that can monitor street-level flooding in real time at predetermined intervals. While the project itself cannot prevent flooding as it only gathers data, it is ASTI's belief that any science-based solution should first emanate from data gathering so that effective solutions are founded on evidence as provided by data. Having an archived dataset can be used for other R&D endeavors that can provide science-based solutions to this perennial problem. This include flood modeling, the results of which can be used as engineering basis for infrastructure projects that aim to solve flooding.

In the short term, since flood levels are being monitored in real-time, information and warnings can be issued to the public. One practical application that will have immediate impact to commuters is in decision making based on the information that is shared by the project. Knowing that a certain street is flooded, commuters can decide on taking another route, stay put at their current location, or if already on the road, park or stay at a safe place and do something worthwhile. Essentially, information empowers and allows commuters to plan ahead and decide what's best for them.

This paper presents the design and development of a system of urban flood monitoring stations and the data that were gathered thus far. The paper also includes a discussion on data transmission, archiving and distribution mainly through a web-based visualization tool. One important feature that is currently being developed is the use of a camera that can serve as a redundant sensor to determine flood levels by employing image processing and providing a visual reference to verify readings from the sensor. Results and measurements presented herein are based on laboratory experiments and the data gathered from two prototype deployments in Earnshaw and San Diego Streets corner Espana Avenue in Manila. These two sites were identified as deployment sites given their history and was approved upon consultation with the MMDA. This paper will concentrate on the pressure-type sensor since this is what is currently deployed in the prototype sites.

2. Methods

In this section, an overview will be presented first and then cascade to important elements that are essential for the system to achieve its objective.

2.1. System Overview

Figure 1 shows an overview of the project.

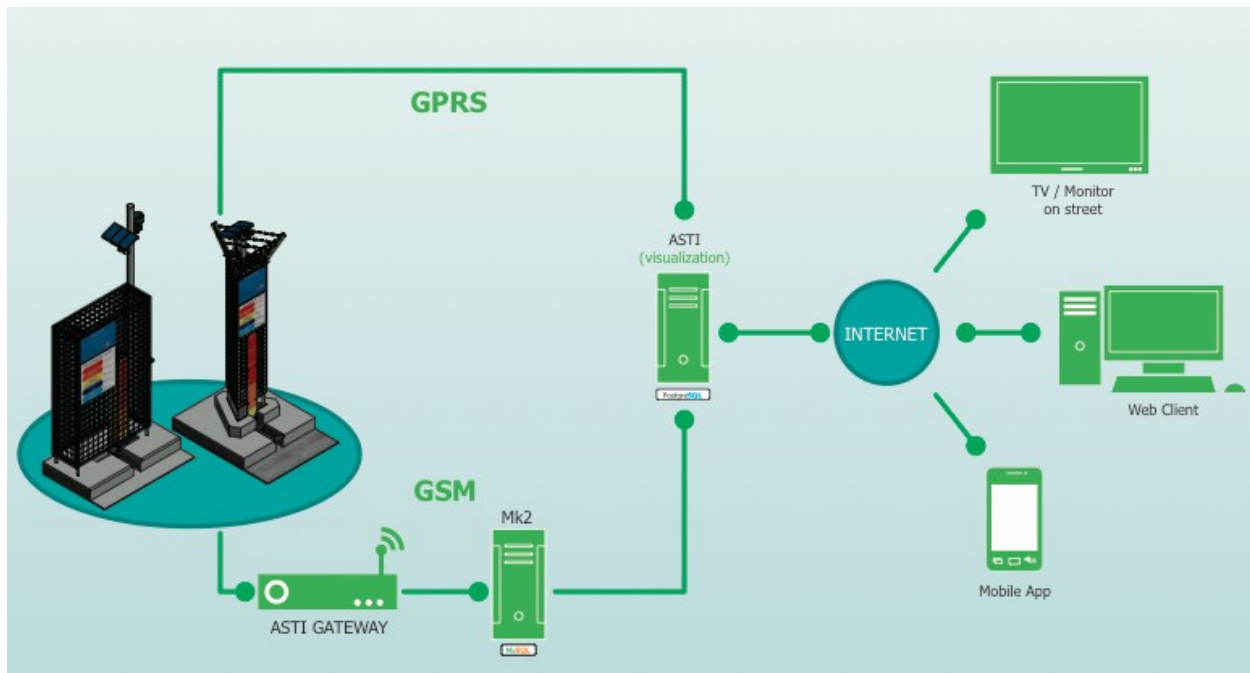


Figure 1. System Diagram for the Project “Flood Sensor Development, Installation, and Monitoring of Urban Flooding in Metro Manila”

In the above figure, data gathered by the stations are transmitted either by General Packet Radio Service (GPRS) or Global System for Mobile (GSM) Communications. In both cases, data are collected in a server at ASTI, the only difference being an Short Message Service (SMS) Gateway acts as an intermediate equipment to gather data via GSM. Once data has been stored in databases, information can be shared to the public via the web which can be accessed thru a computer or a mobile device. For people on the road, on-street video monitors which displays flood information are also available. For data users that intend to use it for R&D activities or need it in real time, data can be shared automatically by using Application Programming Interfaces (APIs) that were developed by ASTI.

The above system can be broken down into three (3) major sub-systems namely: Instrumentation, data archiving, and data distribution. These are discussed in the succeeding sub-sections.

2.2. Instrumentation Development

The urban flood station is composed of: a datalogger (the ASTI-developed Advanced Remote Data Acquisition Unit or arQ); sensors, in this case a pressure-type sensor that translates pressure measurements to water level, plus a rain gauge to correlate flooding with amount of precipitation; a protective fence; a camera that captures images which then undergoes image processing to determine water level thereby acting as a secondary measurement tool (currently under development). Likewise, the photos are envisioned to serve as a reference to verify accuracy of sensor measurements; three (3) 5-watt solar panels to provide power to the remote station; a housing unit to protect the datalogger and provide a mounting platform for the solar panels; a staff gauge that houses the pressure sensor and also provides a visual reference for flood level; flood level signage based from MMDA to provide a standard information tool for flood level warning to the public, and; a signage for warning against theft and vandalism. Refer to Figure 2 for a diagram of the station.

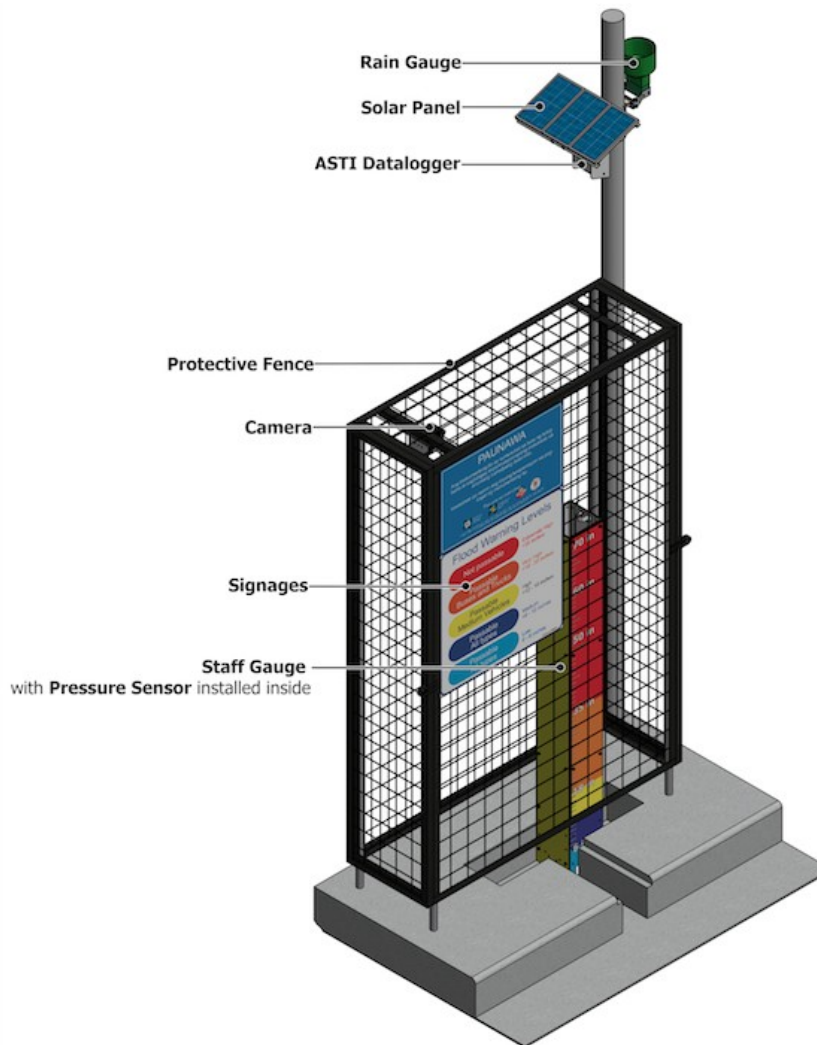


Figure 2. Diagram of Urban Flood Monitoring Station

2.2.1. Advanced Remote Data Acquisition Unit (arQ)

The arQ was developed by ASTI to serve as the controller for all remote stations that the institute has developed. The arQ is currently protected as a Utility Model with the Philippine Intellectual Property Office (IPO) with registration number 2/2012/000525. It is a customizable datalogging device that can be configured to store, record, acquire and send data. For this project, the arQ was configured to effectively interface with a different set of sensors namely a pressure sensor and an image capture device. This entailed performing some changes on the arQ's firmware and removing some features such as the satellite module. Likewise, the arQ was configured to primarily transmit data using GPRS with GSM as secondary mode of transmission using a switching algorithm in case there is failure to send data via GPRS. The arQ also sends health status of the station such as, but not limited to, battery level and signal strength in the area. This helps developers in troubleshooting and preventive maintenance. The arQ can also be polled via SMS to get data on demand. A registration and authentication algorithm allows mobile phones to subscribe to the arQ and perform data queries. Presented below is a diagram of the arQ.

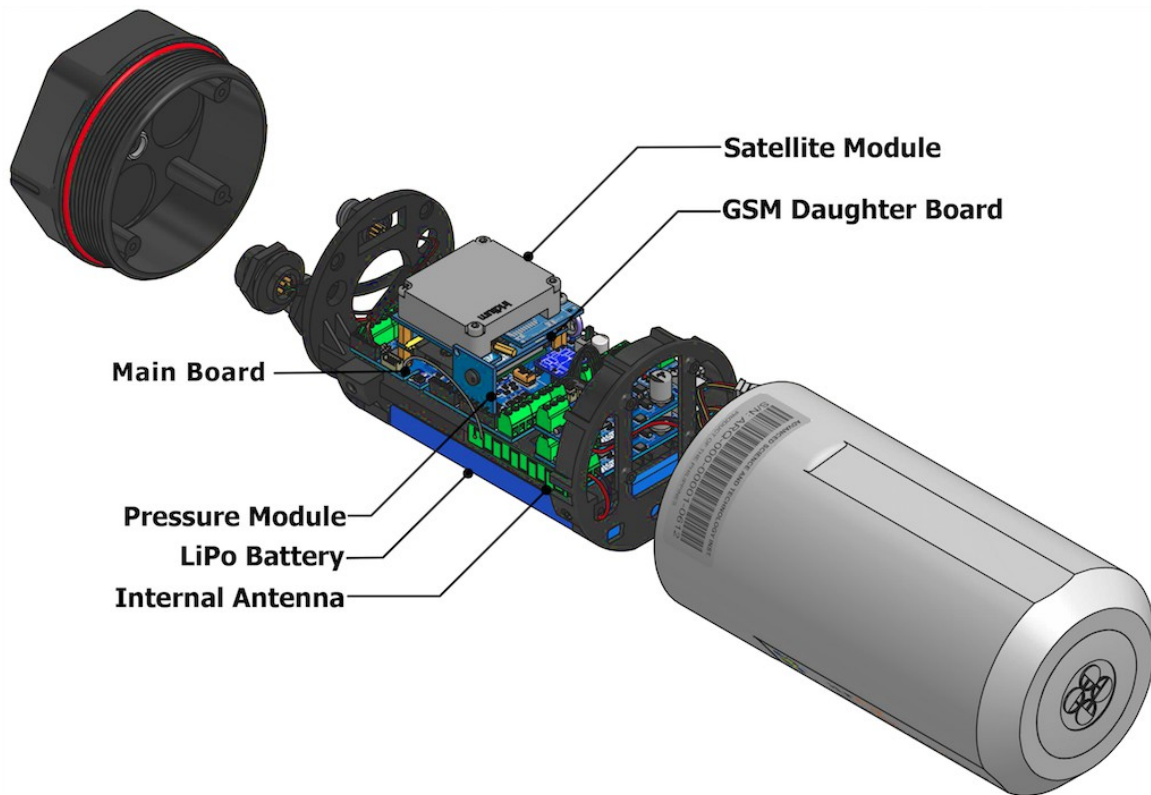


Figure 3. Diagram of arQ

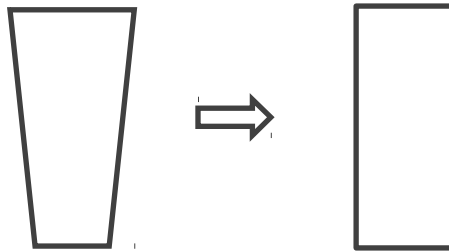
2.2.2. Image Processing

Image processing is given importance in this paper because it provides redundancy in terms of having a second sensor that can determine flood water level. Likewise, this is the first time that such technology will be incorporated in an ASTI developed remote monitoring station to determine water level. In the future, the results of this project's implementation of image processing will be used as platform for succeeding applications where it is appropriate. Though it is still under development, image processing or machine vision is seen as an important technology in future R&D activities.

Detailed process that was used in experiments are presented below.

a. Image Extraction / Perspective Transformation

The initial image taken from the station has a perspective distortion due to the location of the camera which is essentially looking down at the staff gauge. In the image below, the region of interest (ROI) consists solely of the staff gauge. This image that has a perspective view is extracted and transformed into a planar projection[1].



Each pixel on the ROI is treated as a projection in 3d space. Likewise, each pixel is treated as having 3d coordinates of (x,y,z) . This is denoted as P_a :

$$P_a = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$

P'_a is then specified as the planar projection of the pixels in the ROI as a 2d projection consisting of the coordinates $(x,y,1)$:

$$P'_a = \begin{bmatrix} P'_x \\ P'_y \\ 1 \end{bmatrix}$$

A matrix H_{ab} is composed where H_{ab} is the map matrix describing the linear transformation between two projective planes:

$$H_{ab} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

A matrix transformation is performed to transform the image into a planar projection:

$$P'_a = H_{ab}P_a$$

The above operations were performed using Python + OpenCV to successfully transfer the ROI to a planar projection.

b. Pre-processing

The transformed image is pre-processed to enhance the pertinent features such as the lines and edges that will be used for further analysis. The pre-processing stage consists of 2 processes: Adaptive Thresholding, and Sobel Edge Detector.

b.1. Adaptive Threshold

Prior to the application of a threshold, the image is converted to a grey-scale color-space after which a threshold function is then applied:

$$b(x, y) = \begin{cases} 0 & \text{if } I(x, y) < T(x, y) \\ 1 & \text{else} \end{cases}$$

where $b(x,y)$ is the resulting binary pixel after a threshold check of the image. The function $T(x,y)$ is the adaptive mean threshold computed from the target pixel along with its neighbors and has shown far more effectiveness for isolating outlines on images with inconsistent illumination instead of setting a fixed global threshold [2]. Finally, the image is processed with a simple blur filter to smoothen the outlines.

b.2. Sobel Edge Detection

A Sobel edge detection algorithm is applied to the binary image to isolate the edges of the image. The Sobel edge detection algorithm uses two convolution kernels to detect changes in vertical contrast (h_x) and horizontal contrast (h_y):

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Performing kernel convolutions on the image results into two gradient approximation of the image containing the isolated vertical and horizontal edges, namely G_x and G_y . These two gradient approximations are treated as vector quantities and its gradient magnitude is computed as:

$$G = \sqrt{G_x^2 + G_y^2}$$

The resulting magnitude is an image consisting of the combined isolated vertical and horizontal edges[3].

c. Feature Extraction: Hough Transform

After pre-processing the image, a Hough Transform is applied on the image. The Hough Transform works with the premise that every line can be express as a polar coordinate (r, θ) . For every edge pixel point $i(x, y)$ in the image, it is possible to get all (r, θ) values and store it in an accumulator. Once all the edge pixels are checked, the (r, θ) with the most occurrence can be determined[4]. For this application, the (r, θ) with the most occurrence is the vertical line from the top of the image to the interface with the water level. Since the image's pixel height to its real world length was mapped, the water level can be computed as follows:

$$\text{Waterlevel} = (\text{MaximumPixels} - \text{PixelLength}) * k$$

where k is the ratio of the image's pixel height to its real world length.

In summary, the staff gauge image is extracted from a picture to remove the unnecessary background and noises. Since the staff gauge has a geometric distortion due to the angle the photo was taken, a geometric transform algorithm was employed to transform the staff gauge image from initially a trapezoidal shape into a rectangular projection. Once transformed, the color of the staff gauge was converted into grayscale color space to remove unnecessary color noises in the image. A horizontal edge detection algorithm was then applied to detect the edge of the water, if there is any, on the image. Once this edge was found, the height of this water edge was computed by subtracting the edge's pixel coordinate from pixel length of the staff gauge. This distance is then multiplied by a scale factor to convert it into water level.

2.3. Data Transmission

Data is sent from the station using two methods: GPRS and GSM. GPRS is the default sending method for the station. Using GPRS, the station is configured to send the data via TCP socket to the IP address and port of the server. The server listens to this port via a Python script which receives, parses and inserts the data on the database. When GPRS sending is not available, the station automatically switches to SMS. The station is configured to send the data as a message to the number of a GSM Gateway which then parses and inserts the message on a MySQL database.

A python script is then used to transfer the data into the PostgreSQL database used by the web application.

2.4. Data Archiving

Two databases are used in this application namely PostgreSQL and MongoDB. A combination of SQL and NOSQL databases is employed to handle specific tasks in the application.

PostgreSQL is used as a standard storage system for its reliability and data security. PostgreSQL shines in historical queries of the data, however it is not fit for web applications requiring realtime updates and push notifications. In small scale system, continuously polling PostgreSQL for new updates in the data will not cause any noticeable bottleneck in the system. In a production system where the data continuously grow, eventually this method will bottleneck the system and would require scaling into a hardware with more resource.

MongoDB shines in applications where there is a need for instant data availability. MongoDB inserts the data into the RAM and inserts it later in the hard disk for the backup. The time interval between inserting the data from the RAM into the hard disk makes mongodb non-ideal for very critical data handling. However for data caching and realtime applications, MongoDB shines because of its fast data read and write mechanisms. It also provides a lightweight publish-subscribe system. This feature is taken advantage of in the web application by subscribing to a channel where MongoDB can publish the latest inserted data. Compared to long-polling methods, this frees-up system resources while ensuring data is pushed to the web application in realtime.

2.5 Data and Information Distribution

From the databases, a server application pushes the data to the web site (client) via web sockets in order to provide realtime updates and push notifications between the server and the client. The data is then visualized using a combination of HTML5, CSS3 and JavaScript libraries for the consumption of the users. Figure 4 shows snapshots of the visualization tool.

The visualization tool provides interfaces for displaying real-time data from the sensors and a 24-hour historical reading of the water level and rain amount. The markers change colors depending on the water level currently measured in order to provide an immediate visual warning for the users. The color scheme is based on the flood warning levels that MMDA is using. The visualization tool was also designed to be responsive to various mobile devices for ease of access to the users. ASTI also hosts its own map server for the visualization tool hence there is no need to access commonly used maps for the map tiles.

Other than the visualization tool, data can also be shared automatically through the use of APIs developed by ASTI. This scheme works best with partner institutions who need real time access to data. Finally, LED monitors are also going to be put in place at strategic locations to provide flood information to commuters who are already on the road.

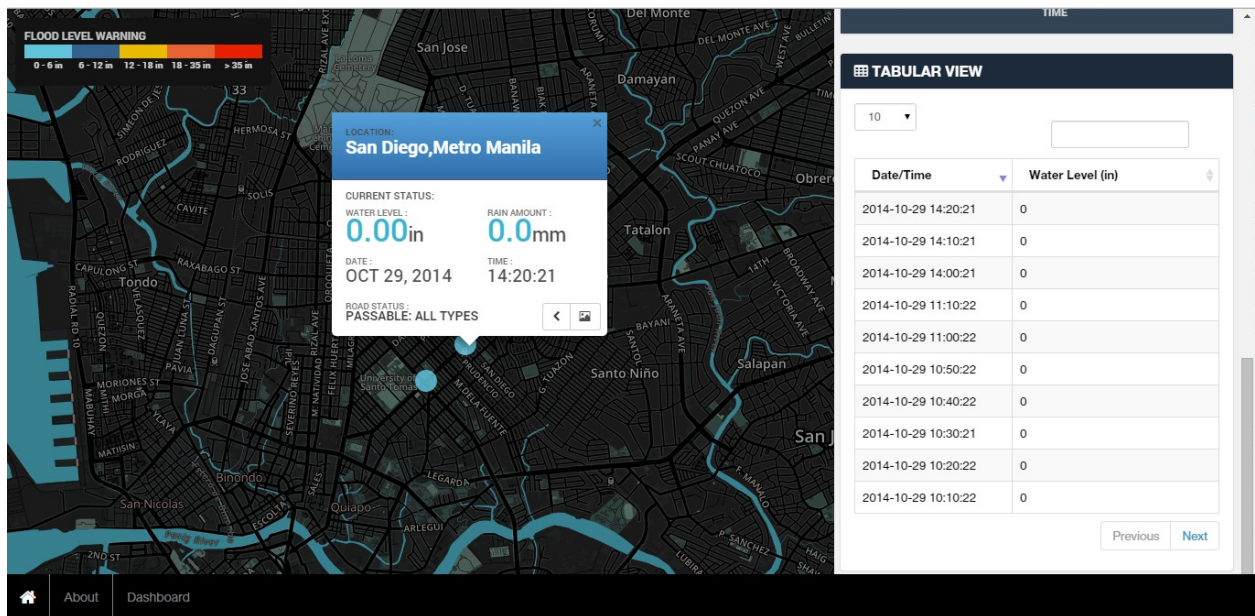
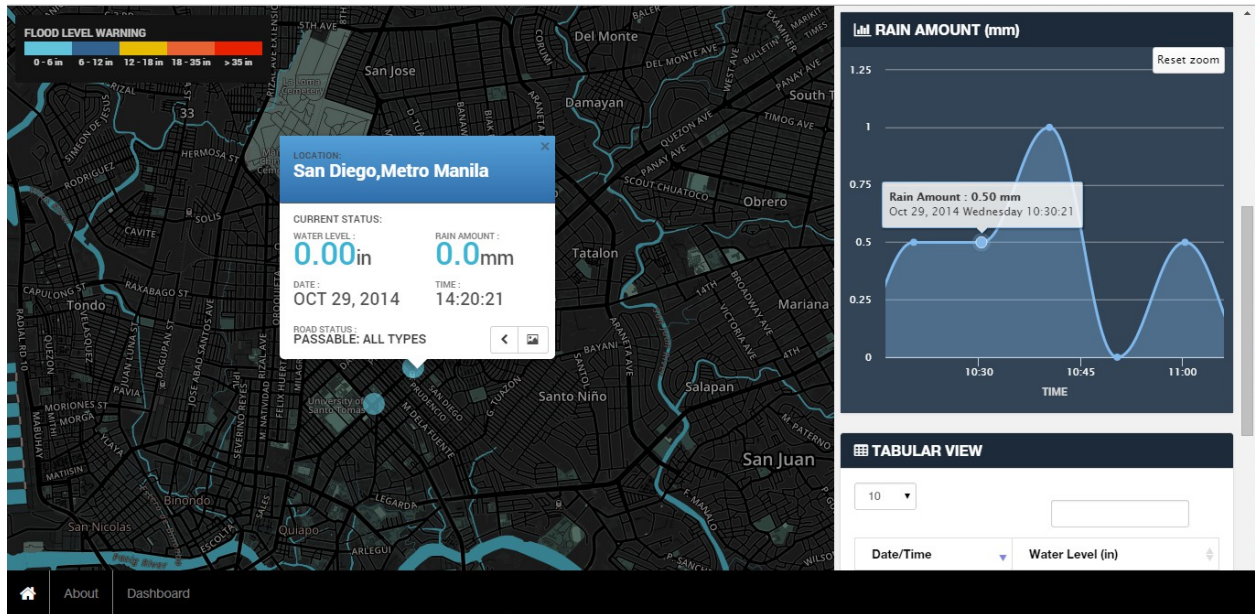


Figure 4. Snapshots of Web-based Visualization Tool

3. Results

The performance of the station will, in a large part, depend on the accuracy of measurements that were taken. Prior to deployment, lab experiments were conducted using a set-up that includes an arQ datalogger and a pressure sensor which are mounted on a tripod. Water is poured unto a container and the actual height is read then compared with the sensor readings. An algorithm was used which to automatically convert pressure readings to water level height. Ten (10) samples are taken for every water level reference. Results are shown in the tables below:

Measured Water Level: 23 cm			Measured Water Level: 60 cm		
Pressure (Pa)	Temperature (C)	Height (cm)	Pressure (Pa)	Temperature (C)	Height (cm)
89	33.36	23.909	3476	31.42	58.450
89	33.63	23.909	3475	31.4	58.440
89	33.6	23.909	3476	31.39	58.450
90	33.59	23.817	3477	31.39	58.460
90	33.52	23.817	3476	31.35	58.450
89	33.54	23.909	3476	31.35	58.450
89	33.53	23.909	3476	31.35	58.450
89	33.5	23.909	3477	31.35	58.460
90	33.52	23.817	3477	31.35	58.460
90	33.5	23.817	3476	31.38	58.450
	Average	23.872		Average	58.452
	Difference	0.872		Difference	1.548
	% Difference	3.791%		% Difference	2.579%

Table 1a. Sensor Measurements for 23 cm and 60 cm Reference Water Level

Measured Water Level: 80 cm			Measured Water Level: 120 cm		
Pressure (Pa)	Temperature (C)	Height (cm)	Pressure (Pa)	Temperature (C)	Height (cm)
5386	31.3	77.929	9263	31.13	117.469
5388	31.31	77.950	9262	31.1	117.459
5387	31.31	77.939	9262	31.14	117.459
5388	31.32	77.950	9262	31.07	117.459
5388	31.3	77.950	9262	31.08	117.459
5389	31.29	77.960	9262	31.08	117.459
5388	31.29	77.950	9261	31.08	117.449
5388	31.26	77.950	9262	31.09	117.459
5389	31.3	77.960	9263	31.08	117.469
5398	31.26	78.052	9260	31.07	117.439
	Average	77.959		Average	117.458
	Difference	2.041		Difference	2.542
	% Difference	2.55%		% Difference	2.117%

Table 1b. Sensor Measurements for 80 cm and 120 cm Reference Water Level

Measured Water Level: 150 cm			Measured Water Level: 180 cm		
Pressure (Pa)	Temperature (C)	Height (cm)	Pressure (Pa)	Temperature (C)	Height (cm)
12162	31	147.020	15094	30.93	176.919
12161	31.96	147.025	15094	30.93	176.937
12162	30.99	147.035	15093	30.91	176.927
12162	30.99	147.035	15093	30.93	176.927
12162	30.99	147.035	15094	30.9	176.937
12161	30.96	147.025	15094	30.91	176.937
12162	30.96	147.035	15094	30.91	176.937
12162	30.96	147.035	15094	30.94	176.937
12162	30.98	147.035	15093	30.93	176.927
12162	30.95	147.035	15094	30.92	176.937
	Average	147.032		Average	176.933
	Difference	2.968		Difference	3.067
	% Difference	1.978%		% Difference	1.703%

Table 1c. Sensor Measurements for 150 cm and 180 cm Reference Water Level

From the lab experiment results, it is clear that there are slight deviations from the sensor readings. However, with further experimentation, these small deviations can be used as correction factors to get as close as possible to the correct value. For R&D purposes, it may be necessary to use correction factors to achieve higher data accuracy. For purposes of providing information and warnings to the public, however, the small difference is acceptable specially since these information and warnings are normally qualitative rather than quantitative. Commuters generally appreciate qualitative information as these are more descriptive and can be associated right away with every day things. For example, describing flood level as knee-deep is better appreciated than saying it is 60cm deep. Given that knee-deep is a non-standard measurement since it differs from person to person, then small differences in measurement by the sensors becomes acceptable for purposes of public warning.

Presented below are data gathered from the deployed stations along Espana Avenue during recent flooding events.

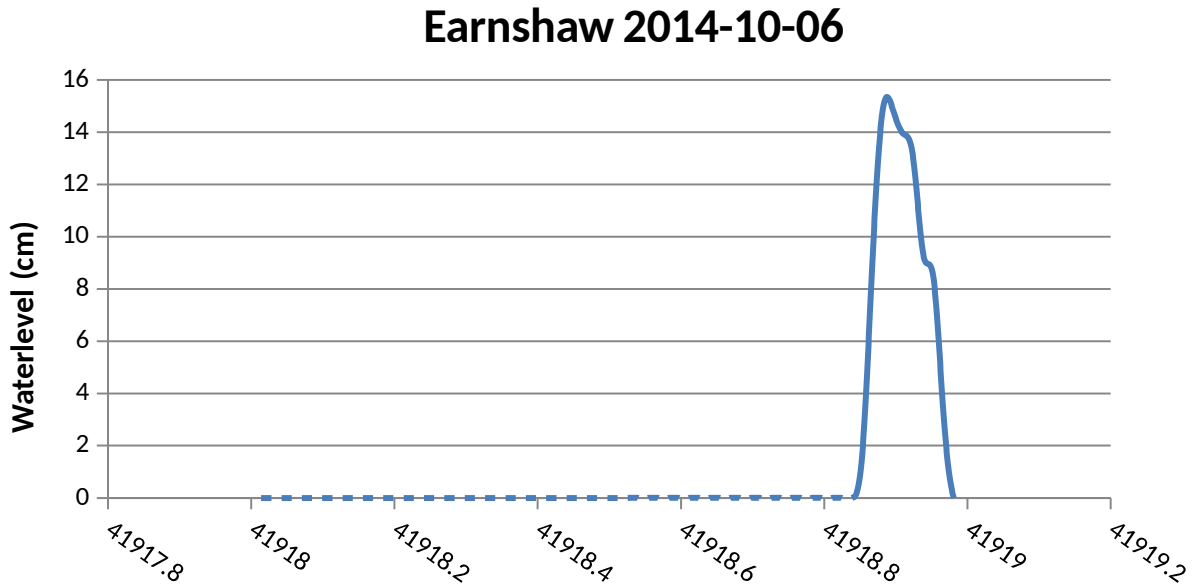


Figure 5a. Graph of Water Level at Earnshaw Station, October 6, 2014

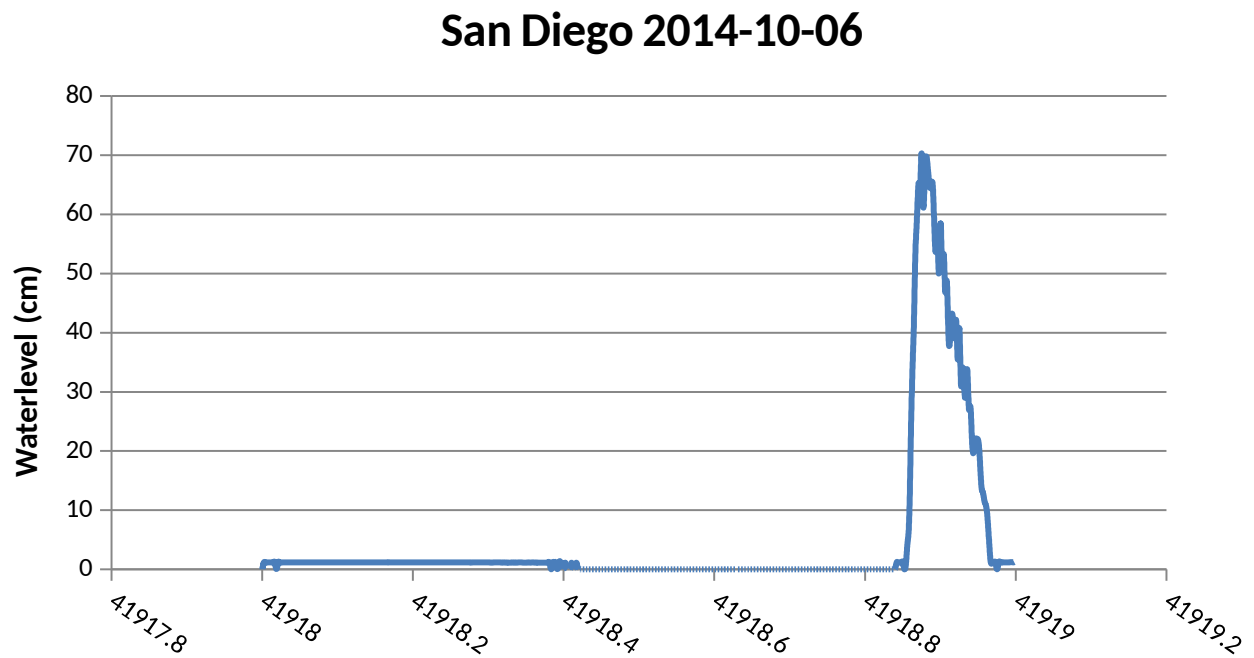


Figure 5b. Graph of Water Level at San Diego Station, October 6, 2014

Comparing the two graphs, we can see that both events happened roughly at the same time., i.e. between 8PM to 11:30PM of October 6, 2014. The two stations recorded maximum water level at around 9PM after which the water began to recede. The two graphs essentially showed the typical behavior of flood water. Unfortunately, given that this is an actual flooding event, it is impractical and unsafe to send observers at the site to conduct measurements and verify sensor readings. Though the sensors have been tested at the laboratory and it can be inferred that the results are likely within the same range of difference from the actual flood level, the best way to verify is still to have some means to observe the water level at the time when the sensor takes readings.

To solve this, a camera is proposed to be installed such that it can take a photo of a reference object, such as a staff gauge, at the same time the sensor takes measurements. In addition, the camera serves as a redundant sensor which can take measurements through image processing. Currently though, developments are still on going with the use of a camera to improve the accuracy of measurements taken specially for the farthest pixels, i.e. low water levels, which tend to have a significant difference. The error can be attributed to transforming the image from a 3d projection into a planar projection. The lower base of the staff gauge contains less pixel information than the top and applying the transformation algorithm essentially stretches the pixels from the lower base which then introduces a high degree of noise. This becomes prominent and detectable by the edge detection algorithm which results to error in the reading. Based on experiments, maximum error observed is around 13%. As for higher water levels, where more pixel information is available, the error is only 2.9% and goes as low as 0.1%. It should be noted that currently, the setup only uses a camera with a VGA resolution (640 x 480) hence one workaround to minimize the error is to use another camera with a higher pixel count.

In addition, the camera that was used for experiments lasted only two weeks and all five (5) test cameras began to fail, hence, it was not practical to deploy one at an actual site even if the sole purpose is to just take photos of a reference object to verify sensor readings. Given this situation, the developers are looking for another camera, preferably with a higher resolution and one that has higher reliability. Development will continue upon the arrival of a new image capture device.

4. Conclusions

The results of the laboratory experiments show that the measurements taken by the sensors have minimal difference with actual level, ranging from 1.703% to 3.791% difference. With further experimentation, these differences can be used as correction factors to make the readings closer to actual. Getting more accurate readings is likely important if used in R&D endeavors but for issuance of public warnings and information, the results are acceptable especially since public warnings are qualitative in nature rather than quantitative. Warnings such as knee-deep flood level is more understood and better appreciated by the public rather than giving quantitative information such as 60cm flood level. Given that qualitative information such as knee-deep is

non-standard measurement since it varies from person to person, then small variations in quantitative measurements are acceptable given the variability of the counterpart qualitative information.

The results from the two (2) deployed stations along Espana that were taken from a recent flood event showed the typical behavior of flood water. However, there is no means to verify the accuracy of the readings given the impracticality and unsafeness of stationing an observer at the site during a flood event. To solve this, a camera is proposed to be installed at the station to take image captures of a reference object such as a staff gauge. Likewise, through image processing, the camera provides redundancy as a secondary sensor for measuring water level.

Acknowledgments

This paper is based on the project titled “Flood Sensor Development, Installation, and Monitoring of Urban Flooding in Metro Manila” which is being implemented by ASTI and is funded and monitored by DOST through the Philippine Council for Industry, Energy and Emerging Technology Research and Development (PCIEERD).

References and Notes

1. L. S. Nguyen, B. Schaeli, D. Sage, S. Kayal, D. Jeanbourquin, D. A. Barr and L. Rossi. 2009. Vision-based system for the control and measurement of wastewater flow rate in sewer systems. IWA Publishing Water Science & Technology—WST 2009
2. T. Romen Singh, Sudipta Roy, O. Imocha Singh, Tejmani Sinam, Kh.Manglem Singh. 2011. A New Local Adaptive Thresholding Technique in Binarization. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 2, November 2011
3. Sobel Edge Detection by James Matthews. (n.d.). retrieved from <http://www.generation5.org/content/2002/im01.asp>
4. R. Szeliski. 2010. Computer Vision: Algorithms and Applications. 2010 Microsoft Research

© 2014 by the authors; licensee Asia Pacific Advanced Network. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).