

Automatic collection of fuel prices from a network of mobile cameras

Yi Fei Dong¹, Salil Kanhere¹, Chun Tung Chou¹, Nirupama Bulusu²

¹ School of Computer Science & Engineering,
University of New South Wales, Australia
{ydon,salilk,ctchou}@cse.unsw.edu.au

² Portland State University, USA
nbulusu@cs.pdx.edu

Technical Report
UNSW-CSE-TR-0803
February 2008

THE UNIVERSITY OF
NEW SOUTH WALES



School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

It is an undeniable fact that people want information. Unfortunately, even in today's highly automated society, a lot of the information we desire is still manually collected. An example is fuel prices where websites providing fuel price information either send their workers out to manually collect the prices or depend on volunteers manually relaying the information. This paper proposes a novel application of wireless sensor networks to automatically collect fuel prices from camera images of road-side price board (billboard) of service (or gas) stations. Our system exploits the ubiquity of mobile phones that have cameras as well as users contributing and sharing data. In our proposed system, cameras of contributing users will be automatically triggered when they get close to a service station. These images will then be processed by computer vision algorithms to extract the fuel prices. In this paper, we will describe the system architecture and present results from our computer vision algorithms. Based on 52 images, our system achieves a hit rate of 92.3% for correctly detecting the fuel price board from the image background and reads the prices correctly in 87.7% of them. To the best of our knowledge, this is the first instance of a sensor network being used for collecting consumer pricing information

Keywords: Automatic data collection, Computer-vision-based sensing, Consumer pricing information gathering, Participatory sensor networks, Vehicular sensor networks.

1 Introduction

The technology of wireless sensor networks (WSNs) has been applied to a plethora of application domains, e.g., farming [24], structure monitoring [19], military [1], environmental monitoring [14], home health care [3], home environment control [23] etc. In this paper, we propose a novel application of using WSN to automatically collect prices of fuel¹ from still images of road-side price board of service stations. To the best of our knowledge, this is the first instance of a WSN being used for collecting consumer pricing information.

Price dispersion of homogenous goods appears to be a fixture of our economy. It is not uncommon for us to find two shops within a close proximity selling an identical item at different prices. For instance, on 18 January 2008, around 9pm, two service stations in Roseville, Sydney, that are less than 1 km apart, were selling unleaded fuel at 1.359 and 1.439 Australian dollars per litre. With the presence of price dispersion, the availability of consumer pricing information can therefore be advantageous and this has been confirmed by a number of studies by economists on the effect of information on the on-line economy. For example, the study in [4] showed that consumers who used on-line price comparison service could save up to 16% in buying electronic goods on-line. Also, the on-line economy has also been shown to drive down insurance prices in [5].

Given the existence of price dispersion in fuel, a number of websites exist to enable consumers to compare fuel prices. Examples of these include **Gaswatch**, **GasBuddy**² in the United States; **fuelprice**³ in the UK; **RACV**, **fuelwatch** and **motormouth**⁴ in Australia. These websites use a couple of different methods to collect their fuel price data. Some of these websites send their workers out, once or twice a day, to collect the data; however, this is labour intensive and it is difficult to track price changes since different service stations update their price at different times of the day⁵. Some other websites rely on volunteers manually relaying the fuel price information. Although the use of volunteers can provide more frequent updates, the need to manually relay the fuel price data may discourage many people from volunteering. Are there other alternatives other than manual collection? One possibility is for the service stations to broadcast their fuel prices. However, there is no incentive for the service stations to do this since availability of information can drive up the competition. Alternatively, one may also consider deploying infrastructure at each service station to monitor the prices; however, the capital investment can be costly.

Given the above discussion, we therefore see the efficient collection of fuel price information (or more generally of a lot of information at the street level)

¹Unfortunately, different parts of the English speaking world use different words to refer to fuel for automobiles. In North America, it is commonly known as "gasoline" while it is called "petrol" in the United Kingdom and Australia. We have chosen to use "fuel" in this paper to avoid bias towards a particular community.

²<http://www.gaspricewatch.com>, <http://www.gasbuddy.com/>

³<http://www.petrolprices.com/>

⁴<http://www.racv.com.au>, <http://www.fuelwatch.wa.gov.au/>, http://motormouth.com.au/default_nf.aspx

⁵Price of fuel does change a few times a day at many service stations across Sydney.

to be a technical challenge. We believe that a method to overcome this technical challenge is to use the *Sensing Data Market (SenseMart)* framework that the authors of this paper have proposed in [9]. The SenseMart framework consists of two key ideas. Firstly, it leverages the existing sensing and communications infrastructure in collecting sensing data. Secondly, it provides a platform for people to exchange and share sensing data. This paper proposes a WSN architecture that allows volunteers to automatically collect, contribute and share fuel price information, thus lowering the cost or barrier for sharing. Our proposed system (which will be detailed in Section 2) leverages on the ubiquity of cameras, mobile Internet connectivity, GPS (Global Positioning System) and GIS (Geographic Information System). Through the use of GPS and GIS, our system knows that when the vehicle of a contributing user is getting close to a service station and triggers the camera automatically. These images are then processed by a computer vision algorithm to extract the fuel price, the details of which will be described in Sections 3 and 4.

This paper makes the following contributions:

- We propose a new application of WSN to collect consumer pricing information.
- We propose a novel WSN architecture — whose components include cameras, mobile phones, GPS, GIS, computer vision algorithms as well as the sharing of sensing data — that can automatically collect fuel prices from images of service station, thus lowering the barrier of sharing sensing data.
- We implemented a working prototype which automatically detects and classifies fuel prices from images of service station billboards. Based on 52 images, our system achieves a hit rate of 92.3% for correctly detecting the fuel price board from the image background and reads the prices correctly for 87.7% of them.

2 System Design

Our system has two principal modes of operation: (i) fuel price collection and (ii) user query. In this paper, we primarily focus on the former. The most important goal of our system is to automate the process of collecting the fuel prices. This is achieved by automatically triggering the mobile phones of contributing users to take pictures of roadside fuel price boards when they approach service stations while driving. Our system employs sophisticated computer vision algorithms to scan these images and retrieve the fuel prices. To reduce the complexity of the computer vision tasks, our system relies on contextual information that is made available by GPS and GIS software. Figure 2.1 presents a pictorial overview of our system. As depicted in the picture, the data collection process involves three steps: (i) capturing images of the fuel price boards, (ii) uploading the images to the central server and (iii) extracting fuel prices from the images. Each of these tasks is executed by a distinct component of the system. In the following, we discuss the design and implementation of the three main components that make up our system and describe their operation.

Camera Sensor: The primary function of the camera sensor is to automatically capture images of the price boards of approaching service stations. In

our system, we have assumed that the mobile phones of contributing users will serve this purpose. Note that, almost all current mobile phones are equipped with in-built cameras. We assume that the phone is mounted on the dashboard in front of the front passenger, with the camera lens pointing towards the road, i.e. leftwards. Consequently, the camera can capture pictures of roadside objects on the left of the car (Note that in Australia we drive on the left side of the road). In the future, it is expected that cars would be fitted with cameras for implemented sophisticated ITS (Intelligent Transport System) applications. Our system can be readily interfaced with these in-car cameras. In this instance, the images captured by the cameras could be transferred wirelessly via Wifi or Bluetooth to the mobile phone.

Figure 2.2 depicts a logical representation of the various components that contribute to realizing the automated capturing of images. The control unit, which oversees the operation is implemented as a daemon in the mobile phone. The control unit periodically polls the in-built GPS receiver of the phone to obtain the current location coordinates of the car. It should be noted that a large majority of the new mobile phones are equipped with GPS receivers. Alternately, it is fairly straightforward to connect an external GPS receiver to a phone via Bluetooth. Our system also requires that GIS software such as Google Maps, TomTom or Nokia Maps is installed on the phone. GIS systems use the GPS location coordinates and a map database of the city road network to estimate the current street position of the mobile. GIS systems can also call to attention any approaching Points of Interest (POI) such as restaurants, shopping malls and service stations. For example, TomTom alerts users of approaching service stations and also indicates the provider information (i.e. brand, e.g.: Caltex, Mobil, etc). Our system takes advantage of these capabilities of the GIS software for automating the data collection process. The control unit queries the GIS system to obtain the location coordinates and the brand of the next approaching service station along the path of the car. Using this location data, the control unit first determines if the service station is on the appropriate side of the road. Recall, that as explained earlier the mobile phone is mounted with the camera lens facing the left-side of the road. Next it determines if the car is within an acceptable distance for taking pictures (a configurable threshold which is set to 10 meters) and then automatically triggers the camera sensor to take several pictures. The camera is switched off once the car has passed the service station. The images are tagged with contextual information provided by the GPS and GIS software such as the service station location coordinates, brand information and time of capture. This information significantly simplifies the complexity of the image processing algorithms that are executed at the server. The images together with the meta-data are passed on to the data upload unit, which is responsible for uploading the pictures to the central server. It is important to note that the entire process is automated and does not require any user intervention.

Data Transport: Our system leverages on the existing communication infrastructure for effecting the transfer of images to the server. Current mobile phones have ubiquitous Internet connectivity via the GSM/GPRS/3G/HSDPA cellular network and in some cases also via in-built 802.11 interfaces. The data upload unit of the mobile phone establishes a TCP connection with the central server using any of the available underlying access technologies. The images are uploaded to the server using this reliable channel. An alternative is to upload

the images as multimedia SMS messages.

Central Server: The central server implements the computer vision algorithms for processing the images and extracting the fuel prices. A standard desktop computer is sufficient for implementing this functionality. We have implemented a TCP server daemon, which receives the images uploaded by the mobile users. The meta-data (location coordinates, service station brand and time) is extracted and stored separately. The images along with the fuel brand information are passed on to the image processing engine. Figure 2.3 represents the sequence of steps involved in the price extraction process. The first step involves detecting the existence of a fuel price board in the image, which by itself is a significantly challenging task. However, note that, each fuel brand uses a specific color combination for their fuel price boards. For example, BP uses a green color code, whereas Mobil boards are blue in color. To reduce the complexity of the problem, we utilize the knowledge of the brand of the service station to assist in the detection process. Recall that this information is included as meta-data with the uploaded pictures. For each brand, we employ a tailored color thresholding that is able to capture regions within the images, which have a similar color scheme as that of the price board. In certain situations, surrounding objects within the image may have colors resembling the board resulting in more than one potential candidate regions, e.g.: the blue sky in the background image may be similar to the color of the Mobil board. In this situation, we use two post-processing techniques to narrow down the search. In the first instance, we make use of certain features of the dimensions of the price boards to exclude some of the candidate regions selected by our color thresholding algorithm. Further refinement is achieved by comparing the color histogram of all candidate regions with that of a sample image of the price board of that particular brand. The detection process concludes with the identification of the precise location of the price board within the image. The image is then cropped such that it solely contains the price board. The cropped image is then normalized to a standard size and resolution. The goal of the next step is to extract the individual fuel price characters from the image. To achieve this, we first convert the color picture to a binary image and then employ connected component labeling to extract out the individual numeral characters of the fuel prices. The final step uses a Feedforward Backpropagation Neural Network algorithm for classifying the price numeral characters. The neural network utilizes a training template containing numeral characters compiled from a number of sample images of fuel price boards. A detailed description of all of the aforementioned steps can be found in the subsequent section.

Once the prices are extracted they are stored in a database which is linked to a GIS road network database populated with the locations of the service stations. This GIS database is consistent with that installed on the mobile phones. Since, each image is tagged with the time and location, the server updates the fuel prices of the appropriate service station in the database if the current image has a timestamp later than that of the stored prices. The past history of fuel prices at each service station is also recorded for analyzing pricing trends.

As discussed earlier, this paper primarily focusses on the data collection aspect of our system. We now briefly describe how our system deals with user queries. The central server implements a simple query resolver daemon for servicing pricing queries from users. A simple client program implemented on the

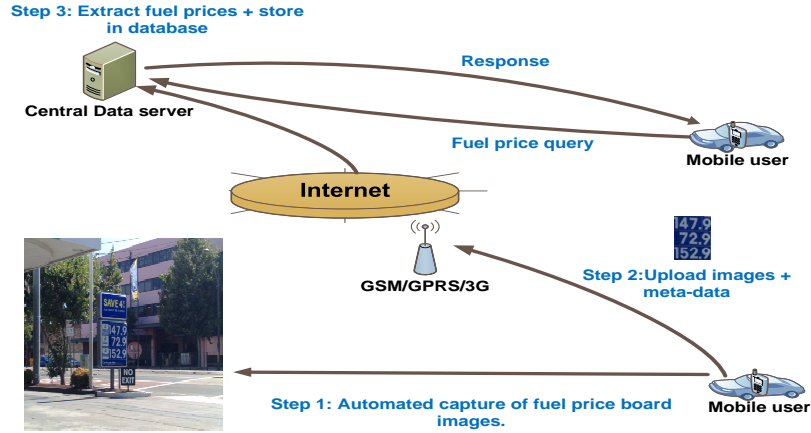


Figure 2.1: System Overview

mobile phone allows users to query for the cheapest refueling option in their vicinity. The client program obtains the current location coordinates of the user by querying the GPS receiver. The users' query tagged with his location information is then sent to the central sever. The server searches the fuel price database for the cheapest service station in the vicinity of the user and sends back a reply to the users mobile phone, which contains the price, location coordinates and brand of the chosen service station. In our future work, we also intend to implement support for more sophisticated queries such as "find the cheapest fuel station collocated with a convenience store". In addition, we plan on providing for alternate modes of access to the pricing information, such as via SMS and a web-based interface.

3 Computer Vision Algorithm for Extracting Fuel Prices

Despite recent advances, computer vision still can not deal with complex environmental problems, though it can work accurately in controlled environments. In the following we present some of the key environmental challenges that are encountered by our system:

1. Some of the pictures may contain extraneous objects that overlap with the fuel price board. For example, in Figure 3.1(a) ¹ a white car has masked the lower part of the board.
2. The color of some objects in the background may be similar to the color of the price board, E.g., the blue color of the wall in the background in Figure 3.1(b) is similar to that of the fuel price board. Consequently, the

¹This and other images in Section 3 are best viewed on a colour screen. Full size versions of these images can be found at the following link: http://www.cse.unsw.edu.au/~ydon/dcoss08_images/

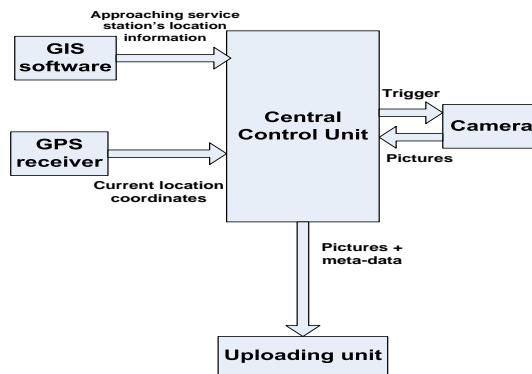


Figure 2.2: Mobile sensor components

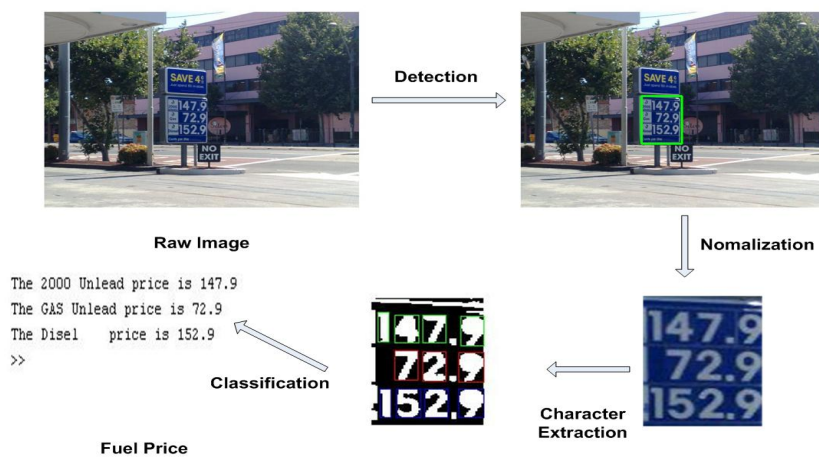


Figure 2.3: Overview of the computer vision algorithm



Figure 3.1: Examples illustrating the challenges encountered by computer vision

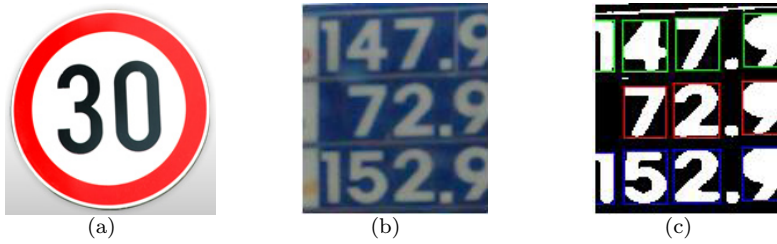


Figure 3.2: Problems encountered in classifying fuel price characters

wall may be wrongly interpreted as the price board if color is used as the key feature in the detection process.

3. Since the images are captured while in motion, it is likely that some of them appear blurred or unfocussed. Figure 3.1(c) illustrates one such case.
4. Recognition and detection of road signs from digital images is a mature area of research [12, 21, 2]. However, a road sign differs significantly from a fuel price board as evidenced by comparing Figures 3.2(a) and 3.2(b). A typical image of a fuel price board not only contains several small characters but also is more likely to have extraneous objects that could be wrongly interpreted as characters. For example, the cropped white label to the left of the price in the top row may be wrongly interpreted as "1". Figure 3.2(c) illustrates an instance where all individual characters are successfully extracted.
5. Computer vision algorithms used for extracting text from images consist of two important steps: (i) Detection of the target area and (ii) Extraction of the characters. It is likely that the quality of the images required for the successful completion of the individual tasks may vary significantly. For example, it may be fairly easy to detect the price board in a particular image. However, the text extraction may fail for the same image.

To overcome these and other challenges, our system makes use of certain contextual information by employing the assistance of GPS and GIS. Our algorithm consists of two key steps, which are described in the subsequent sub-sections.

3.1 Fuel Price Board Detection

Given an image, the task of detecting the presence of a fuel board and precisely identifying its location is quite difficult to solve with the current state-of-the-art in computer vision. Hence, our system relies on GPS and GIS to provide additional contextual information, which significantly simplifies the problem. As explained in Section 2, the GIS system includes the service station brand as meta-data with the image. Each fuel brand uses a specific color combination for its price boards, e.g.: BP price boards are green and Mobil boards use blue as the backdrop. We use this color information as a key feature for isolating objects in the picture which can be potential fuel price boards. In this paper we have only focussed on BP and Mobil, two of the largest fuel providers in Sydney. In our future work, we intend to encompass all the major providers in Australia. There are two prominent schemes for representing color images: RGB (Red, Green and Blue) and HIS (Hue, Intensity and Saturation). The HIS color space is immune to changes in the lighting condition, but is computationally intensive since it involves nonlinear transformation [12]. On the contrary, RGB is sensitive to changes in the lighting conditions, but is computationally efficient. Since our ultimate goal is to implement the computer vision algorithms directly on the mobile phones, we have chosen to use the RGB color space.

Figure 3.3 depicts the step-by-step evolution of the board detection algorithm for one particular example image. The original image, containing a Mobil board that serves as the input to the detection algorithm is shown in Figure 3.3(a). We use the Mobil board which is blue in color as an illustrative example to explain our algorithm. A similar logic was used to identify BP boards.

Since RGB is an additive color space, a blue component image can be readily obtained by boosting the blue level of each pixel and subtracting the corresponding green and red components as indicated by the following equation,

$$B(x, y) = 2 * fb(x, y) - fr(x, y) - fg(x, y) \quad (3.1)$$

where $fb(x, y)$, $fr(x, y)$ and $fg(x, y)$ are, respectively, the functions representing the blue, red and green levels of each pixel in the image [12]. Figure 3.3(b) represents the blue component image of the original image in Figure 3.3(a). A similar approach can be employed for other colors.

Color thresholding [13] is one of the earliest techniques for detecting objects in digital images. Thresholding aims to classify all the pixels of an image into *object pixels*, which correspond to regions potentially containing the object and *background pixels*. Since the Mobil board is blue, we employ color thresholding with the criteria that a pixel in the blue component image is an object pixel if $B(x, y)$ exceeds a certain threshold, B_{th} . The output of the color thresholding produces a binary (i.e. black and white) image, with all object pixels taking on a value of 255 (note that, each pixel is represented by 8 bits) and background pixels being zero. The above process can be represented mathematically as,

$$B(x, y) = \begin{cases} 255, & \text{if } B(x, y) \geq B_{th}, \\ 0, & \text{else,} \end{cases} \quad (3.2)$$

In our first attempt at implementing color thresholding, we tried to use a singular value of the threshold B_{th} , which was calculated using a few test images. However, when we used this value for testing our dataset (details in Section 4),

Average Intensity Range	$AI \leq 3$	$3 < AI \leq 9$	$9 < AI \leq 15$	$15 < AI \leq 70$	$70 < AI$
B_{th}	0	20	50	80	120

Table 3.1: Thresholds for images with different average intensities

we observed very poor results. This was expected given that the RGB color space is quite sensitive to lighting conditions. As an improvement we used a set of test images and manually classified them into 5 distinct groups representing different lighting conditions. We then computed the range of the average intensities (AI) of the blue component images for each group. Next, we determined the value of B_{th} for each group that achieved a 100% positive detection of the price board. Table 3.1 shows the range of AI values for each of the groups and the corresponding threshold value, B_{th} .

Figure 3.3(c), shows the binary image produced by the color thresholding. Since, each row of the fuel prices is separated by a line, the fuel price board actually appears as three sub-divided regions in the binary image. These adjacent regions are merged together to form the complete price board by using morphological reconstruction [13], which is shown in Figure 3.3(d). Finally, connected component labeling [13] is employed to group the pixels into components based on pixel connectivity, i.e, all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. The first step of connected component labeling is edge detection. Comparing with other edge detection operators, sobel edge detection operator gives more importance to the points fall in the line [13]. Thereby it affords better edge detect effect. As a result, the sobel operator is applied here to obtain edges in Figure 3.3(d), which produces a binary image BW of the same size as Figure 3.3(d) with 1's where the sobel operator finds edges in Figure 3.3(d) and 0's elsewhere. Once the edges is obtained, we need to calculate the connected components in BW . The component filtering mask coefficients are defined in Equation 3.3.

$$connect_mask = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.3)$$

The connected component of image of BW is obtained by performing two-dimensional convolution of image BW and $connect_mask$. Finally the obtained components are then labeled as different numbers. Figure 3.3(e) shows the output of the the connected component labeling. One can readily observe that it contains several regions, which can be potential price boards. Our algorithm uses two post-processing techniques, which make use of certain known features (color and dimension) of the fuel price boards to narrow down the choices and correctly detect the price board.

Dimension Comparison: The first step makes use of the *a priori* knowledge of the dimensions of a typical price board to exclude regions, which are obviously either too small or too large. However, determining a set of constraints



(a)



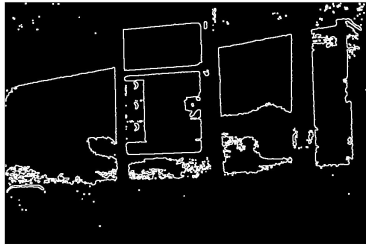
(b)



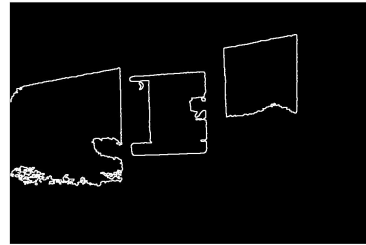
(c)



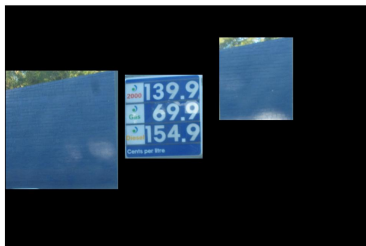
(d)



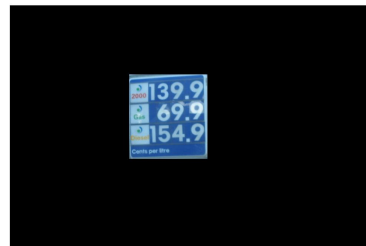
(e)



(f)



(g)



(h)

Figure 3.3: Intermediate steps involved in the detection process

based on dimensions can be tricky since the pictures are not captured from a static location. The camera can be at variable distances from the price boards when the pictures are captured. Further, the orientation of the camera lens relative to the price board may be different in each image. We have analyzed a set of test images of the price boards to encompass all these variables. Our test set includes close-up and far-away views of the price boards and pictures captured with the camera lens at different angles to the plane of the price board. Our analysis concludes that a candidate region of width W_i and height H_i can only qualify as a potential fuel price board, if the following constraints are met:

$$W_i \geq \frac{W_{image}}{30} \quad (3.4)$$

$$1 \leq \frac{H_i}{W_i} < 2.5 \quad (3.5)$$

where W_{image} is the width of the image containing the candidate region. Figure 3.3(f) is the output of the first post-processing step. As compared to the input image in Figure 3.3(e), the number of candidate regions have now been reduced to 3.

Color Histogram Comparison: The second post-processing technique compares the histogram distribution of the candidate regions to that of a template of the price board. The Chi square distance [28] between h_i , the histogram of the candidate region and h_j , the histogram of the price board template is computed as follows,

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{m=1}^K \frac{[h_i(m) - h_j(m)]^2}{h_i(m) + h_j(m)} \quad (3.6)$$

where K is the number of histogram bins. As χ turns out to be a large number, we normalize this distance as follows,

$$D_{norm} = \frac{\chi}{K \times WI} \quad (3.7)$$

where K is the number of histogram bins and WI is the width of the image. In our system, we set K as 60, since it provides for good accuracy and low complexity. WI is set to the standard value of 480 pixels. To determine an appropriate threshold value for the normalized distance, D_{norm} , we used a test set containing 10 cropped images that include the fuel price board and 10 images that do not contain one. The normalized distance of each image in the set was computed by comparing its histogram with that of the price board template using Equation 3.7. Based on the test results, we conclude that if a candidate region has $D_{norm} \leq 2.5$, then it is very likely that this region corresponds to the fuel price board. Figure 3.4(a) shows the template used in our tests and Figure 3.4(d) represents its color histogram. Figure 3.4(b) is a cropped image from the test set that contains the fuel price board and its color histogram is plotted in Figure 3.4(e). D_{norm} for this particular image is 1.6. Figure 3.4(c) represents an image from the test set that does not contain the price board and its corresponding histogram is shown in Figure 3.4(f). D_{norm} for this image is 3.9, considerably higher than the threshold of 2.5. Figure 3.3(h) is the output of

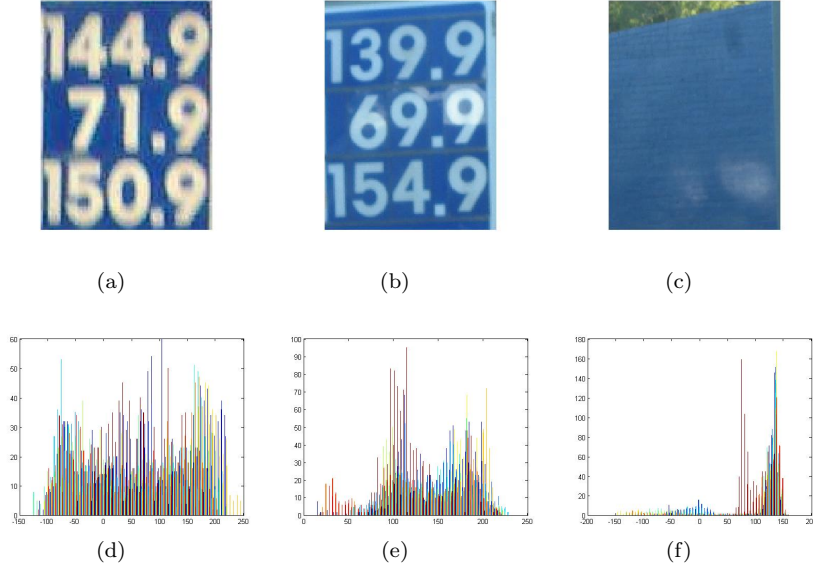


Figure 3.4: Color histograms for a few example images

applying the histogram comparison. The fuel price board has now been correctly identified. This concludes the detection phase. The next step is to extract the fuel prices from the board.

3.2 Fuel Price Classification

The last section describes an algorithm on detecting the fuel price board from the background of the image. This section describes the procedure to extract and classify the numeral characters which make up the fuel prices in a price board. The procedure consists of two steps: extraction of the characters and followed by classification by a neural network.

Character extraction: In Figure 3.5(a), it is shown for a successful detected fuel price board. Variations of detected fuel price boards due to differences in size, perspective angle make automatic fuel price characters extraction extremely challenging. Hence, we normalize the detected fuel price boards to a standard resolution (129-by-112 pixels). The fuel price boards are usually designed such that there is sufficient contrast between the characters and the background color. As a result, an image of the fuel price board does not contain excessive noise. Therefore, it is sufficient for us to use a binary image (instead of full color image) for character extraction which can significantly reduce the amount of processing. We first convert Figure 3.5(a) into a gray scale image which is shown in Figure 3.5(b). As Otsu's method is widely used to find optimal threshold of an image with bimodal histogram due to its simplicity in computation [25], we adopt it to select a proper threshold for the gray scale image. According to this threshold the gray scale image is then converted to a binary image. The binary image of Figure 3.5(b) is shown in Figure 3.5(c). In

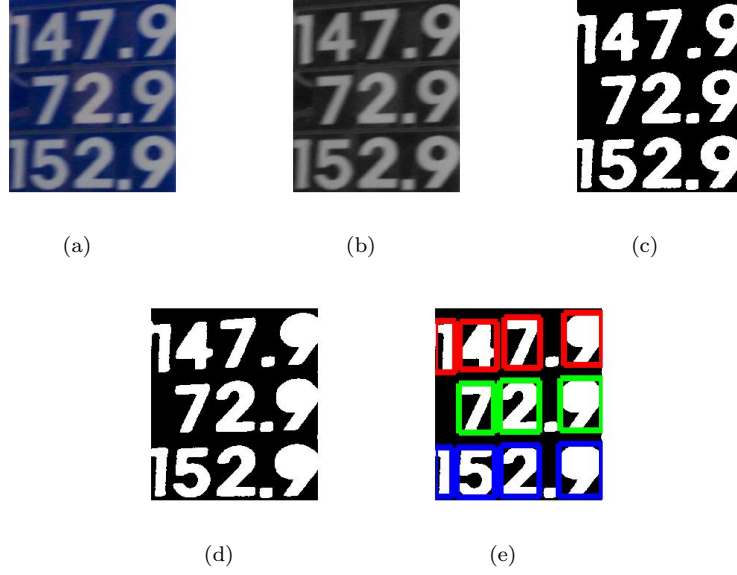


Figure 3.5: Character extraction process

order to match the character template which is shown in Figure 3.6, the holes in the binary image are filled by using morphological reconstruction [13]. The filled binary image is shown in 3.5(d). For character extraction, we employ the bounding box algorithm in [29]. If the width and height of the fuel price board are defined as W_{board} and H_{board} respectively, the width of the narrowest character "1" is at least $\frac{1}{20}W_{board}$. Moreover, a character's height should be between $\frac{1}{5}H_{board}$ and $\frac{1}{3}H_{board}$. As a result, we define the constraints for a character as Equation 3.8 and Equation 3.9.

$$W_c \geq \frac{W_{board}}{3} \quad (3.8)$$

$$\frac{H_{board}}{5} \leq H_c \leq \frac{H_{board}}{3} \quad (3.9)$$

where W_c and H_c are the width and height of the character respectively. The result of cropped characters are shown in 3.5(e). The cropped (or extracted) characters are then normalised to a standard size of 50-by-70 pixels. For the construction of feature vectors for character recognition, we divide each 50-by-70 pixel character into 35 10-by-10 pixel regions and compute the average intensity in each region. Thus, each extracted character will be represented by a 35×1 feature vector.

Character Recognition: For character recognition, we employ the Feed-forward Back-propagation Neural Networks (FFBPNN) due to its simple implementation and robustness to interference [17]. The FFBPNN is first trained by using characters extracted from 20 sample fuel price boards which are shown in Figure 3.6. The feature vector construction procedure is identical to the one described in the last paragraph. Thus, each character in the training set is rep-



Figure 3.6: some neural network template image before processing

resented by a 35×1 feature vector.

For the identification of fuel type, we have, for the time being, chosen to use the *a priori* knowledge of the order in which different types of fuel are arranged for a particular chain of service stations. For example, all Mobil stations in Australia arrange the prices of fuel in the following order: unleaded fuel, Liquid Petroleum Gas (LPG) and diesel. Our future work will integrate fuel type identification into our system instead of depending on *a priori* knowledge.

4 Evaluation and discussion

In order to evaluate our system, we have set up a data set with 52 images of 5 Mobil and 3 BP service stations. These images were captured by either a 5-megapixel Nokia N95 mobile phone camera or a 4-megapixel Canon IXUS 400 camera. The cameras were held by a passenger sitting in the front passenger seat of a moving car. Our data set covers many different conditions — relative distance of the price board from the vehicle (close or far away), weather conditions (sunny, cloudy) and daylight disparities — in order to test the generality of our computer vision algorithm. Each image in the data set contains one fuel price board. There are 3 fuel prices per board and on average there are 11 numeral characters per board. The size of each image is 640-by-480 pixels. Our system is implemented in Matlab and tested on a 2.0-GHz Pentium centrino duo laptop.

We first evaluate the performance of our price board detection algorithm (Section 3.1) using our data set. We classify an image as "close" if the price board occupies more than $\frac{1}{8}$ of the size of the image; otherwise, it is classified as "far away". The performance of our detection algorithm is shown in Table 4.1 where a hit (or a true positive) is obtained when a price board is correctly detected; otherwise, it results in a miss or a false positive. The overall hit rate of 96% is very high considering the many possible different backgrounds that these images can have. In addition, both "close" and "far away" images include images that are taken under many different lighting conditions. This shows that our algorithm is robust against lighting variations.

In our next step of evaluation, we passed those 48 images whose price board we have successfully detected to the fuel price classification module described

Size of the board	No. of fuel price board	Hit Rate	Miss or False Positive
Close	31	29/31= 93.5 %	2/31= 6.5 %
Far away	21	19/21=90.5%	2/21= 9.5%
Overall	52	92.3%	7.7 %

Table 4.1: Test results of automatic fuel price board detection on 52 images

in Section 3.2. Out of these 48 images, 15 of them contain a price board that is so blurry that even human cannot read the fuel prices. We input these 15 blurry images to our algorithm and it did not return any prices, thus giving a true negative rate of 100%. We will now focus on the remaining 33 images which consist of 15 Mobil and 18 BP fuel price boards. There are a total of 330 characters and 99 fuel prices (a fuel price consists of either 3 or 4 characters) in these 33 fuel price boards. We measure the correct extraction rate and correct classification rate of both the characters and prices. The results are shown in Table 4.2. The overall correct classification rate for fuel prices is about 88% and is pretty high. Note that the classification result for Mobil boards is always lower than that of BP, because the classification algorithm mistook the white border of the Mobil price board as a character in some of the cases.

5 Related Work

Wireless sensor networks: Most of the existing research work on WSNs utilises small form factor and power constrained nodes [18] for sensing and communications. However, the WSN community is beginning to appreciate the value of using vehicles, mobile phones, GPS, cameras etc to collect everyday information. A vehicular WSN aims to exploit sensors mounted on a vehicle to collect sensing data. An example of a vehicular WSN is Cartel [15] which collects information on traffic conditions, WiFi availability and potholes on the road. Other work in vehicular sensor networks focus on routing or efficient information dissemination, see [20] or references therein. The wide-spread availability of mobile phones and cameras forms the basis of Participatory Sensing [6] for collecting data on our urban environment, e.g. air pollution [22], cyclist experience [11] as well as the on-line proceeding of a recent workshop in the area ¹. In addition, the Nokia Sensor Planet² initiative aims to explore the use of camera phones as mobile sensing devices. Our work also leverages on the Participatory Sensing concept but with an emphasis on users contributing and sharing data [9]. Our work can make use of the methods for authenticating the location and timing of participatory sensing data discussed in [6]. However, our work differs from existing work in vehicular WSNs and Participatory Sensing in that we focus on the challenges in automatically collecting consumer pricing data.

¹http://urban.cens.ucla.edu/sensys07/index.php?title=Accepted_Papers_and_Workshop_Agenda

²<http://research.nokia.com/research/projects/sensorplanet/index.html>

Brand	No. of fuel price boards	No. of characters	No. of fuel prices	Extraction rate of characters	Classification rate of characters	Extraction rate of fuel prices	Classification rate of fuel prices
BP	18	165	54	165/165=100%	164/165=99.3%	53/54=98.1%	49/54=91%
Mobil	15	165	45	151/165=92%	146/165=89%	39/45=91.1%	38/45=84.4%
Overall	33	330	99	96%	94%	94.6%	87.7%

Table 4.2: Test results of neural network based character recognition method based on 43 fuel price boards

Detection and recognition of objects from digital images: Automatic detection of road traffic signs is a mature area of research [2, 27, 7, 10]. There have also been several recent successful attempts at detecting text from a diverse set of digital sources such as video, newspapers, advertisements and photographs [16, 26]. However, most of these algorithms require a very large set of training samples (up to 1000 images in some cases). Consequently the training process takes up a large amount of time. Given the difficulty in populating such a large training set for our system, we have chosen to use a much simpler and computationally efficient approach, which can achieve a high success rate. The simplicity of our algorithm will also help our future direction, since we wish to migrate some of the computer vision tasks to run on the resource constrained mobile devices.

Commercial OCR (Optical Character Recognition) software has been successfully used in some instances to realize text recognition [16, 7, 8]. An OCR system makes certain assumptions about the source of the text and works well only with standard layouts and fonts. However, it is unable to cope with unstructured environments. Given that our system has to work in an outdoor environment, which is known to involve a large number of variables (lighting condition, weather, distance, etc), we cannot use commercial OCR software.

6 Conclusions and Future Work

This paper has proposed and described a mobile WSN that allows road-users to automatically collect, contribute and share fuel price information. It represents, to the best of our knowledge, the first WSN application for collecting consumer pricing information. Our proposed system is based on three key ideas. Firstly, it provides a platform for people to contribute and share sensing data, which can be viewed as the BitTorrent of sensing data. Secondly, our proposed system leverages on the existing sensing and communication infrastructure and as a result, lowers the barrier for a volunteer to share sensing information. Thirdly, the use of computer vision algorithms to extract the fuel price information from the contributed images. In this paper, we have focused on defining the architecture of our proposed system and describing the computer vision algorithm. Our preliminary experimental results show that our system is able to correctly classify 87.7% of the fuel prices from 33 images of BP and Mobil service stations. Our future work aims to produce a complete prototype of the proposed system which integrates GPS, GIS and automatic camera triggering. We also plan to test our algorithm on other service station chains. In addition, all the computer vision processing is currently performed on the server and we plan to test how much of the processing can be performed on the current generation of mobile devices. Lastly, we would like to point out that our proposed system is also applicable for other applications. For example, GIS providers can utilise user contributed images of street level view to update their POI instead of relying on manual update.

7 Acknowledgements

We would like to thank Maria Myung-Hee Kim and Chen Zhang for coding the classification algorithm, Gina Lam for taking the images, and Rajib Rana and Assad Mehmood for helpful discussion. This research is funded by the Australian Research Council Discovery Grant DP0770523.

Bibliography

- [1] Ákos Lédeczi et al. Countersniper system for urban warfare. *ACM Transactions on Sensor Networks*, 1(2):153–177, 2005.
- [2] C. Bahlmann, Y. Zhu, Visvanathan Ramesh, M. Pellkofer, and T. Koehler. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In *Proceedings of Intelligent Vehicles Symposium*, pages 255–260, 6-8 June 2005.
- [3] Chris R. Baker et al. Wireless sensor networks for home health care. In *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)*, volume 2, pages 832–837, 2007.
- [4] M.R. Baye, J. Morgan, and P. Scholten. The value of information in an on-line consumer electronics market. *Journal of Public Policy and Marketings*, 3:481–507, 2003.
- [5] J.R. Brown and A. Goolsbee. Does the internet make markets more competitive? Evidence from the life insurance industry. *Journal of Political Economy*, 1:17–25, 2002.
- [6] Jeffrey A. Burke, D Estrin, Mark Hansen, Andrew Parker, Nithya Ramanathan, Sasank Reddy, and Mani B. Srivastava. Participatory sensing. In *WSW06 at SenSys 06*, New York, NY, USA, October 31, 2006. ACM.
- [7] Xiangrong Chen and A.L. Yuille. Detecting and reading text in natural scenes. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 366–373, 27 June-2 July 2004.
- [8] Xilin Chen, Jie Yang, Jing Zhang, and A. Waibel. Automatic detection and recognition of signs from natural scenes. *IEEE Transactions on Image Processing*, 13(1):87–99, 2004.
- [9] Chun Tung Chou, Nirupama Bulusu, and Salil Kanhere. Sensing data market. In *Proceedings of Poster Papers of 3rd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2007)*, June 2007. Available online: <http://www.dcooss.org/dcooss07/dcooss07posterproceedings.pdf>.
- [10] A. de la Escalera, J.M. Armingol, and M Mata. Traffic sign recognition and analysis for intelligent vehicles. *Image and Vision Computing*, 21(3):247–258, March,2003.

- [11] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G-S. Ahn, and A. T. Campbell. The bikenet mobile sensing system for cyclist experience mapping. In *Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pages 87–101, 2007.
- [12] M.A. Garcia, M.A. Sotelo, and E.M. Gorostiza. Traffic sign detection in static images using matlab. In *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 2, pages 212–215, 16–19 Sept. 2003.
- [13] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 1992.
- [14] Wen Hu, Van Nghia Tran, Nirupama Bulusu, Chun Tung Chou, and Sanjay Jha. The design and evaluation of a hybrid sensor network for cane-toad monitoring. In *Proceedings of Information Processing in Sensor Networks (IPSN 2005/SPOTS 2005)*, April 2005.
- [15] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. Cartel: a distributed mobile sensor computing system. In *Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys)*, pages 125–138, New York, NY, USA, 2006. ACM.
- [16] A.K. Jain and Bin Yu. Automatic text location in images and video frames. In *Fourteenth International Conference on Pattern Recognition*, volume 2, pages 1497 – 1499, 16–20 Aug. 1998.
- [17] John JingFu Jenq and Wingning Li. Feedforward backpropagation artificial neural networks on reconfigurable meshes. *Future Generation Computer Systems*, 14(5-6):313–319, 1998.
- [18] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2006.
- [19] Sukun Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *6th International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 254–263, 25–27 April 2007.
- [20] Uichin Lee, Biao Zhou, Mario Gerla, Eugenio Magistretti, Paolo Bellavista, and Antonio Corradi. Mobeyes: smart mobs for urban monitoring with a vehicular sensor network. *IEEE Wireless Communications*, 13(5):52–57, October 2006.
- [21] Carlos Filipe Paulo and Paulo Lobato Correia. Automatic detection and classification of traffic signs. In *Eighth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 11–11, 6–8 June 2007.
- [22] Eric Paulos, R.J. Honicky, and Elizabeth Goodman. Sensing atmosphere. In *Workshop on Sensing on Everyday Mobile Phones in Support of Participatory Research*, 2007. Available online: http://urban.cens.ucla.edu/sensys07/index.php?title=Accepted_Papers_and_Workshop_Agenda.

- [23] Vipul Singhvi, Andreas Krause, Carlos Guestrin, Jr. James H. Garrett, and H. Scott Matthews. Intelligent light control using sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys)*, pages 218–229, New York, NY, USA, 2005. ACM.
- [24] Tim Wark et al. The design and evaluation of a mobile sensor/actuator network for autonomous animal control. In *Proceedings of Information Processing in Sensor Networks (IPSN 2007/SPOTS 2007)*, April 2007.
- [25] A.K.C. Wong and P.K. Sahoo. A gray-level threshold selection method based on maximum entropy principle. *IEEE Transactions on Man and Cybernetics Systems*, 19(4):866–871, Jul/Aug 1989.
- [26] V. Wu, R. Manmatha, and E.M Riseman. Textfinder: an automatic system to detect and recognize text in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:1224 – 1229, Nov. 1999.
- [27] Wen Wu, Xilin Chen, and Jie Yang. Dtection of text on road signs from video. *IEEE Transactions on Intelligent Transportation Systems*, 6:378 – 390, Dec. 2005.
- [28] Nong Ye, Connie M. Borrer, and Darshit Parmar. Scalable chi-square distance versus conventional statistical distance for process monitoring with uncorrelated data variables. *Quality and Reliability Engineering International*, 19(6):505–515, 2003.
- [29] Bo Yuan, Leong Keong Kwoh, and Chew Lim Tan. Finding the best-fit bounding-boxes. *Document Analysis Systems VII*, 3872/2006:268–279, 2006.