# Stony Brook University

# Traffic Scene Understanding using Sound-based Localization, SVM Classification and Clustering

A Thesis Presented
by

**Shreyas Kodasara Rajagopal**

to

The Graduate School
in partial fulfillment of the
Requirements
for the degree of

**Master of Science**
in
**Computer Engineering**

**Stony Brook University**
December 2010

**Stony Brook University**
The Graduate School
**Shreyas Kodasara Rajagopal**
We, the thesis committee for the above candidate for the
Master of Science degree,
hereby recommend acceptance of this thesis.

Dr. Alex Doboli, Advisor of Thesis
Associate Professor, Department of Electrical and Computer Engineering

Dr. Sangjin Hong, Associate Professor,
Department of Electrical and Computer Engineering

This thesis is accepted by the Graduate School.

Lawrence Martin

Dean of the Graduate School

Abstract of the Thesis

# Traffic Scene Understanding using Sound-based Localization, SVM Classification and Clustering

by

**Shreyas Kodasara Rajagopal**

**Master of Science**

in

**Computer Engineering**

Stony Brook University

**2010**

The thesis is about an embedded system application aimed at identifying the semantics of traffic based on acoustic data. Sound localization, classification and clustering are used for scene understanding. The report presents a set of experiments used to simulate different traffic scenarios.

An alternative implementation for sound localization is also explored, where fixed point representation of rational numbers is used instead of floating point numbers. The results for both the implementations are compared in terms of execution speed and accuracy for a Programmable System-on-Chip (PSoC).

*To My Parents, Sister and Artur*

# Contents

# List of Figures

# ACKNOWLEDGEMENTS

This thesis would not have been possible without Dr.Alex Doboli, my Advisor. I offer my sincerest gratitude to him, for his professional support throughout my thesis, wisdom, knowledge and commitment to the highest standards. I attribute the level of my Masters degree to his steadfast encouragement and advice.

I am heartily thankful to my colleagues at the Embedded Systems Lab: Anurag Umbarkar who has been like a mentor to me and has helped me with every aspect of my thesis, Varun Subramanian and Cristian Ferrent for their valuable inputs and support.

My deepest gratitude goes to my family for their unflagging love, support and belief in me. This work would have been simply impossible without them. I am indebted to my parents, sister and brother-in-law, for their love and care.

A special thanks to a special person, Artur Kasperek, a friend, philosopher and guide whose immense support motivated and inspired me in bringing out the best in me in this thesis and throughout my Masters.

# Chapter 1

# Introduction

For human beings, visual and auditory information are important to sense the surroundings. Acoustic information, if interpreted correctly, can be used as an aid for behavior and semantic understanding, and analysis. To understand a specific sound, the system needs to localize the sound source and extract meaningful information from it.

Many research works focus on auditory signal processing but few attempts have been made to use these techniques to understand underlying scene. This thesis focuses on understanding the sound scenes by integrating acoustic signal processing and machine learning techniques. Traffic scenarios are used as a case study for scene understanding. The application is implemented on PSoC1 and PSoC5. PSoC1 is a programmable, mixed-signal SoC that includes 8-bit micro-controller, on-chip SRAM and flash memory, programmable digital blocks, and programmable analog blocks[4]. This makes PSoC a very attractive architecture for this application as it supports integrated implementation of the mixed-signal front end for sound-based lo-

calization. The analog front end of the design consists of signal conditioning, filtering, and analog to digital conversion (ADC). The digital processing includes Hanning windowing, Fast Fourier Transform (FFT), phase calculation, Maximum Likelihood (ML) algorithm[15], and Data clustering and Classification. The hardware implementation of clustering and classification algorithms is done on PSoC5 which operates at 80 MHz and has a 32-bit ARM Cortex core.

The organization of the report is as follows: Chapter 1 gives a description of the problem with the semantic elements to be identified. Chapter 2 describes the different features that can be extracted from sound. Chapter 3 offers an overview of Support Vector machine classifier and clustering. Chapter 4 gives the implementation details of sound-based localization and classifier. Also, an alternative implementation for sound localization using fixed point number representation instead of floating point is explored. Chapter 5 explains the set of scenarios used for validating the implementation. Finally, Chapter 6 presents the conclusions and the future improvements for the application.

## 1.1 Problem Description

The goal of this work is to understand at run time the semantics of traffic scenes based on auditory inputs collected through a network of embedded nodes with sound processing features. Understanding traffic scenes requires the following main types of capabilities:

- *Finding the components of a scene*: This capability identifies the elements of a traffic scene, and their attributes. An important aspect is establishing the needed information that is sufficient to correctly distinguish all relevant elements of a scene.

- *Understanding the relationships in a scene*: this capability finds the relationships and correlations that exist between the elements in a scene. This includes cause - effect relationships, in which a certain element causes or enables a given situation, and correlations in which elements influence reciprocally their characteristics. Getting insight into the origin of the existing relationships is a main function of this capability. In addition to the correlations that result directly from the description of the application, other correlations are produced due to specific conditions and properties of the participating elements.

  For example, the traffic flow can be obstructed by an obstacle on the road (direct correlation), or a set of drivers with specific driving profiles that end up slowing down each other. The second situation cannot be reasoned out from the application description, but should be figured out from the scene characteristics. Disambiguation is a second main function of this capability as multiple causes can produce the same effect. For example, group of vehicles slowing down can be either because of a conservative driver or a pothole present in the road. The sensed information must be used to infer the more likely cause that produces

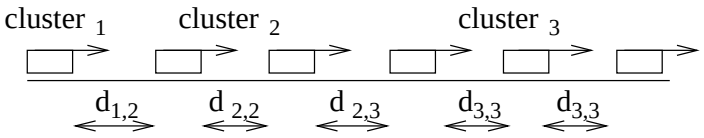


Figure 1.1: An example scenario

the situation.

- *Predicting the evolution of a scene*: the capability refers to the dynamics (evolution) of a traffic scene, including the possible situations that can emerge within a time window.

  *Example:* Let's consider the simple traffic situation in Figure 1.1 to illustrate the three challenges in scene understanding. Figure 1.1 presents five vehicles moving on the road. Scene understanding must first identify the three vehicle clusters, where a cluster comprises of the vehicles moving using the same pattern (e.g., similar speed and speed variations), and which is different from the patterns of other clusters. If the clusters move with different speed then only speed is sufficient for cluster identification. However, if two clusters are moving at the same speed then additional attributes are needed for differentiating the clusters, such as the interspacing $d_{i,j}$ between two vehicles. A possible differentiation criterion is that interspacing is significantly larger than the average of the other interspacings.

Hence, an important challenge in concept identification (including their attributes) is the identification of the necessary and sufficient information that

makes identification possible. Moreover, inferring the information needed for scene understanding helps solving any ambiguities that can occur between different concepts that have some common attributes. Note that concept identification relies not only on finding similarities but also outliers.

The meaning of a traffic scene is defined in terms of a set of basic semantic elements, which are indivisible tokens that can be estimated based on inputs coming from sensors. The basic semantic elements (BSEs) to be identified and analyzed include the following aspects:

- *Vehicle attributes*: some of the typical vehicle attributes include kind, speed, acceleration, position and trajectory.

- *Driver's driving profile*: this includes his/her preferred style of driving depending on traffic and weather conditions. The driver's profile describes the likelihood of changing the speed or trajectory (e.g., switching the lanes).

- *Clusters of vehicles*: clusters are formed by vehicles that travel while having a common set of stationary attributes, such as a constant number of vehicles in the cluster and vehicle speed variations and inter-vehicular spacings that pertain to well-defined (yet unknown) ranges.

- *Cluster attributes*: every cluster is characterized by attributes like size (number of vehicles), speed range, trajectory, time of formation and

time of dispersion. Clusters have also attributes that are different from the attributes of vehicles, e.g., spacing between cars.

- *Cluster-level social behavior*: the way in which the drivers forming a cluster change their driving behavior based on the cluster characteristics, e.g., drivers decide to adapt to the speed of the other drivers in the cluster, or start looking for opportunities to leave the cluster.

- *Cluster dynamics*: vehicle clusters go through modifications, such as a cluster splitting into sub-clusters and different clusters merging into a single clusters. Another kind of interaction is if two clusters automatically correlate their attributes, like speed.

- *Road conditions*: this refers to special road conditions, e.g., the position of potholes, traffic signs, and stopped vehicles.

- *Weather conditions*: this relates to the nature of weather conditions, such as the position of ice and water on the road.

In this thesis, the semantics are identified by extracting features from the sound and then applying machine learning techniques.

# Chapter 2

# Feature Extraction

Feature extraction is an essential pre-processing step to pattern recognition and machine learning problems. It is often decomposed into feature construction and feature selection. Standard machine learning techniques can be used to understand the semantics based on the extracted features. This section discusses about the features that can be extracted from sound signals.

Sections 2.1 to 2.4 discuss about the timbral characteristics that can be extracted from the sound signals [11],[12],[13],[14].

## 2.1 Features from Time Samples

- Volume: RMS of the amplitudes of samples in a small window.

$$RMS = \sqrt{\frac{\sum_{n=1}^{N}(x[n])^2}{N}} \tag{2.1}$$

  where, x[n] is the magnitude of time sample with index n, N is the total number of samples.

- Zero crossing: Number of sign changes of amplitude in a window.

$$Z_t = 0.5 * \sum_{n=1}^{N} |sign(x[n]) - sign(x[n-1])| \tag{2.2}$$

- Low Energy: Percentage of sample amplitudes, smaller than RMS, on each window.

## 2.2 Features from Frequency Spectrum

- Spectral Centroid: Center of gravity of magnitude spectrum of Short-time Fourier Transform (STFT). It is a measure of spectral shape.

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]} \tag{2.3}$$

  where, $M_t[n]$ is magnitude of Fourier transform at frame t and frequency bin n.

- Spectral Roll off: Frequency $R_t$ below which 85% of the magnitude distribution is concentrated.

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n] \tag{2.4}$$

- Spectral Flux: Squared difference between normalized magnitudes of successive spectral distributions. It is a measure of the amount of local spectral change.

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \tag{2.5}$$

- Spectral Bandwidth: Bandwidth of Fourier Transform of frame t.

$$B_t^2 = \frac{\sum_{n=1}^{N} (n - c_i)^2 |M_t[n]|^2}{\sum_{n=1}^{N} |M_t[n]|^2} \tag{2.6}$$

- Band Energy Ratio: It is the following ratio,

$$BER_t = \frac{\sum_{n=1}^{N/4} M_t[n]}{\sum_{n=1}^{N} M_t[n]} \tag{2.7}$$

- Spectral Flatness: Quantifies how tone-like a sound is, as opposed to being noise-like.

$$Flatness = \frac{\sqrt[N]{\prod_{n=1}^{N} M_t[n]}}{\frac{\sum_{n=1}^{N} M_t[n]}{N}} \tag{2.8}$$

- Spectral Crest Factor: Ratio of peak of the spectrum to the RMS value of the spectrum.

## 2.3 Features from Linear Regression of Spectrum

- SpecReg Slope: Slope of linear regression of spectrum. Regression formula can be represented as $M = a + b * n$.

$$Slope(b) = \frac{(N \sum (M * n) - (\sum M)(\sum n))}{\left(N \sum n^2 - (\sum n)^2\right)} \tag{2.9}$$

- SpecReg Y Intercept: Intercept point of regression line and Y axis

$$Intercept(a) = \frac{(\sum M - b\,(\sum n))}{N} \tag{2.10}$$

## 2.4 Features from Modified Frequency Scale

- Mel magnitudes: Obtained by converting frequency spectrum using Mel scale.

$$Mel(f) = 2595 * log_{10}\left(1 + \frac{f}{700}\right) \tag{2.11}$$

- ERB Magnitudes: Obtained by converting frequency spectrum using Equivalent Rectangular Bandwidth (ERB) scale.

$$ERB\,(f) = \frac{107}{5} * log_{10}\left(\frac{10000}{437}f + 1\right) \tag{2.12}$$

- Octave Magnitudes: Obtained by converting frequency spectrum using Octave scale.

$$Oct(f) = \begin{cases} 0 & \text{if } f \leq \frac{55}{128} \\ log_2\left(\frac{128}{55}f\right) & \text{else} \end{cases} \tag{2.13}$$

## 2.5 Localization features

Sound localization is the process of identifying the spatial coordinates of a sound source based on the sound signals received by a microphone array [15]. In this implementation, sound based localization is used to extract the intra-cluster vehicular distance.

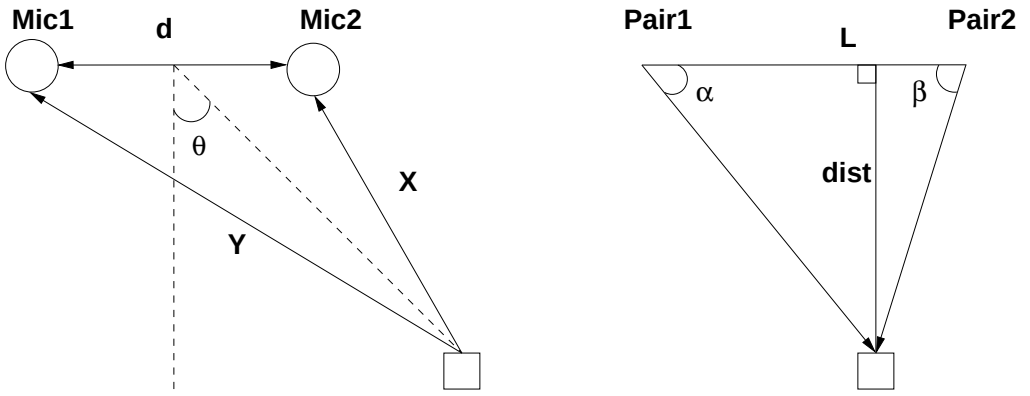


Figure 2.1: Sound Localization: (a) TDOA estimation (b) Process of triangulation

A simple method of localization is to estimate the time delay of arrival (TDOA) of a sound signal between the two microphones. This TDOA estimate is then used to calculate the Angle of Arrival (AoA). Combining the data from two microphone pairs and by using the process of triangulation, we compute the distance of sound source from the microphone pairs as shown in Figure 2.1

In Figure 2.1(a), TDOA = $\frac{\mathrm{Y-X}}{\mathrm{v}}$ , where $v$ is speed of sound and $\theta$ is the AoA. In Figure 2.1(b), 'dist' gives the perpendicular distance between the sound source and line joining the microphone pairs.

The overall process of sound localization is shown in Figure 2.2.

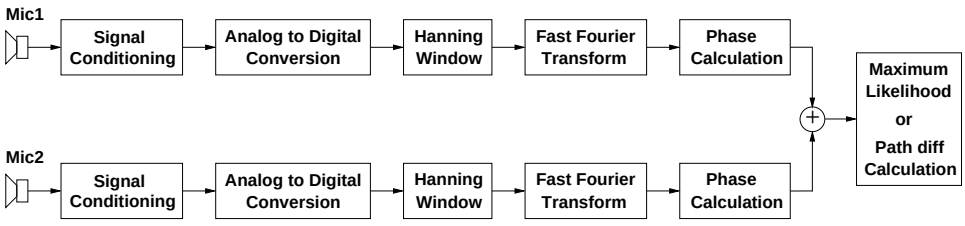


Figure 2.2: Sound Localization Data flow

# Chapter 3

# Classification and Clustering

The main operators used for scene analysis were Support Vector Machine (SVM) based classifier and clustering. The algorithms are written in ANSI C. This chapter gives an overview of the SVM and the implementation method.

## 3.1 Classification

Support Vector Machine (SVM) is a supervised learning system used for classification [3]. SVM is used to predict the class of a feature vector based on the training of the system. SVM performs classification by using a non-linear mapping to transform training data into a higher dimension and constructing hyperplane that optimally separates the data into two categories. The hyperplane is such that the classification error is minimum.

There can be two cases which are discussed in the following sections.

### 3.1.1 Separable Case

Consider the set of training data that consist of two classes,

$$S = \{\mathbf{x_i}, y_i\}\ i = 1\ to\ n, y_i \in \{-1, 1\}, \mathbf{x_i} \in \mathbb{R}^d \tag{3.1}$$

A linear classifier able to separate the positive from the negative examples will be a hyperplane $\mathbb{R}^d$ in characterized by a normal w called weights and an offset b given by:

$$h(\mathbf{x}) \ = \ \mathbf{w}.\mathbf{x} + b \tag{3.2}$$

For a linearly separable data set S, there exists a hyperplane that satisfies all the points in S:

$$\mathbf{w}.\mathbf{x_i} + b \ \geq 0 \text{ for } y_i \ = \ +1 \tag{3.3}$$

$$\mathbf{w}.\mathbf{x_i} + b \ < \ 0 \text{ for } y_i \ = \ -1 \tag{3.4}$$

The above two equations can be written as,

$$y_i(\mathbf{w}.\mathbf{x_i} + b) \ \geq 0 \ \forall i \tag{3.5}$$

The decision function will be of the form:

$$\mathrm{D}(\mathbf{x}) \ = \ sign(\mathbf{w}.\mathbf{x} + b) \tag{3.6}$$

There are an infinite number of ($\mathbf{w}$,$b$) pairs satisfying the above inequality, since for any ($\mathbf{w}$,$b$) satisfying the inequality 3.5 ($a\mathbf{w}$, $ab$),$\forall a > 0$, also satisfies it. To make the solution unique, (w,b) can be rescaled so that the closest

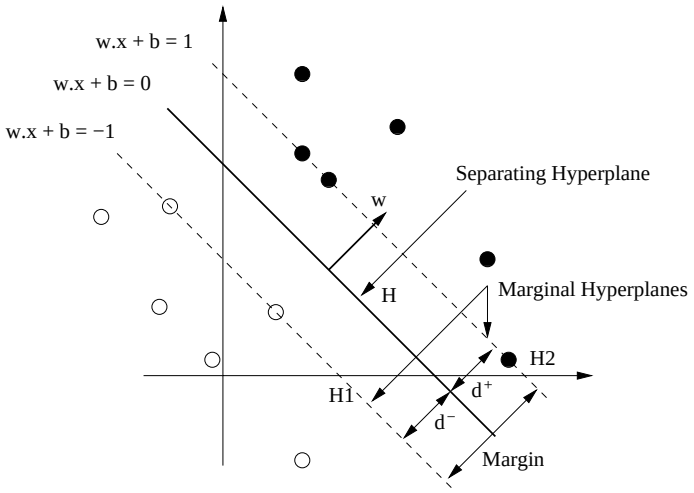


Figure 3.1: Linear SVM for the separable case

points to the hyperplane satisfy $|(\mathbf{w}^T\mathbf{x}_i+b)| = 1$. This normalization leads to the following canonical form for SVM,

$$y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 \, \forall i \tag{3.7}$$

Let $d^+$, $d^-$ be the distances from the separating hyperplane to the closest positive, negative examples, respectively. The margin of the separating hyperplane is defined as $d^+ + d^-$. It can be seen from Figure 3.1 that, the closest positive, negative examples to the separating hyperplane are those points lying on the hyperplanes H1, H2, respectively, and hence the margin size equals

$$d^+ + d^- = \frac{1}{||\mathbf{w}||} + \frac{1}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||} \tag{3.8}$$

The goal of SVM is to find the pair of hyperplanes $H_1$, $H_2$ that maximize the margin, subject to the constraints 3.7. This can be formulated as the

following constrained optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w^T w},$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}^i + b) \geq 1 \; \forall i \tag{3.9}$$

This is a convex optimization problem for which we are guaranteed to obtain its global optimal solution.

### 3.1.2 Non-separable case and the Kernel Trick

In the real world, there are many data sets that are not linearly separable. When the SVM derived above is applied to non-separable data sets, some data points $\mathbf{x}_i$ could be at a distance $\xi_i/||\mathrm{w}||$ on the wrong side of the margin hyperplane (Figure 3.2). To extend the SVM to handle non-separable data, the constraint 3.7 can be relaxed and a further cost can be added for doing so. More precisely, positive slack variables $\xi_i$ are introduced, one for each data point $\mathbf{x}_\mathrm{i}$, and the constraint 3.7 is transformed to

$$y_i(\mathbf{w}^T\mathbf{x}^i + b) \geq 1 - \xi_i, \; \xi_i \; > 0 \; \forall i \tag{3.10}$$

For a misclassification error to occur, the corresponding $\xi$ must exceed unity, hence $\frac{1}{n}\sum_{i=1}^{n}\xi$ is an upper bound of the average loss on the training data. Therefore, a way to assign an extra cost for errors is to add a new term $\frac{C}{n}\sum_{i=1}^{n}\xi$ to the cost function, where C is a parameter to be chosen by the user. A larger $C$ corresponds to a higher penalty to errors, with

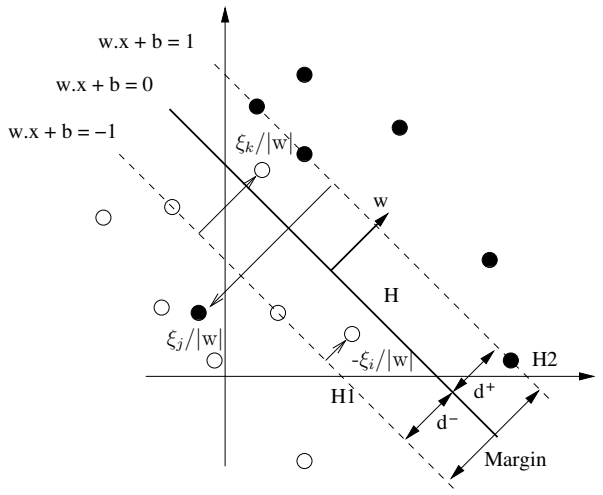


Figure 3.2: Linear SVM for the non-separable case

$C = \infty$ meaning that no error can be tolerated at all. Now the SVM for non-separable case can be casted as the following optimization problem:

$$\min_{\mathbf{w},b,\xi_i} \frac{1}{2}\mathbf{w^T w} + \frac{C}{n}\sum_{i=1}^{n}\xi_i$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}^i + b) \geq 1 - \xi_i,\ \xi_i\ > 0\ \forall i \tag{3.11}$$

Again, this is a convex optimization problem, and the Lagrange multiplier method can be applied to obtain the globally optimal solution.

The Kernel trick is used when the classification boundary is non-linear. It attempts to map the original feature space, into a feature space of higher dimension where the data can be linearly separated using a kernel function. In this new feature space linear SVM can be used for classification.

Radial basis (Gaussian Kernel) is used in the implementation for the transformation and it is given by:

$$k(x, y) = \mathrm{e}^{\frac{-||\mathrm{x}-\mathrm{y}||^2}{2\sigma^2}}$$

## 3.2 Clustering

Data clustering is the process of partitioning a collection of data into a number of cluster based on some criterion function. The criterion function determines the accuracy of clustering. An SVM-based clustering algorithm clusters data with no *a priori* knowledge of input classes [16]. The algorithm initializes by first running a binary SVM classifier against a data set with each vector in the set randomly labelled, this is repeated until an initial convergence occurs. In order to obtain convergence, an acceptable number of KKT violators must be found. This is done by running the SVM on the randomly labeled data with different numbers of allowed violators until the number of violators allowed is near the lower bound of violators needed for the SVM to converge on the particular data set. Once this initialization step is complete, the SVM confidence parameters for classification on each of the training instances can be accessed. The lowest confidence data (e.g., the worst of the mislabelled vectors) then has it's labels switched to the other class label. The SVM is then re-run on the data set (with partly re-labelled data) and is guaranteed to converge in this situation since it converged previously, and now it has fewer data points to carry with mislabelling penalties. This continues until no more progress can be made.

# Chapter 4

# Implementation overview

## 4.1 Sound Localization

The entire implementation for sound localization is first coded in C. This software implementation is run for the sound signal samples, obtained through PSoC1 using serial communication. The hardware implementation is done on PSoC1. The microphone circuitry is mounted on the breadboard of the PSoC Eval board. The analog modules are implemented using reconfigurable analog blocks inside the PSoC while the digital modules are implemented in C code [15].

An alternative implementation to the floating point computations is done using fixed point arithmetic in order to speed up the execution [8]. The floating point computations require very large number of clock cycles to perform basic multiplications and additions. Also, the floating point implementation requires 32 bits to store a single value, whereas the fixed point arithmetic can be fit in half of this value or any other feasible size. In order to accommodate

rational numbers with relatively high accuracy we 7 bits are used to store the decimal part, 8 bits for fraction part and 1 bit for the sign. This representation is called Q7.8 notation (Figure 4.1). For example to store, $\pi$= 3.140625, we will have to store 3 in decimal part and $0.140625 * 2^8 = 0x24$ in fractional part as shown in Figure 4.1:

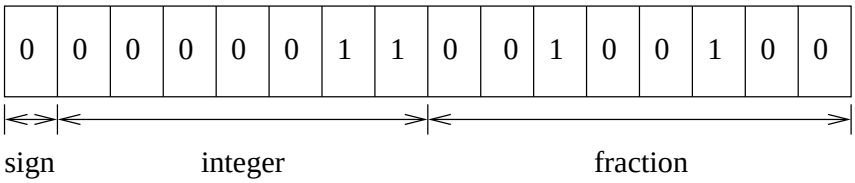


Figure 4.1: Q7.8 representation of $\pi$

Using 8-bits for fraction part gives a precision of $2^{-8} = 0.00390625$ which sufficient to perform the FFT and all other computations required for sound localization. The addition in Q7.8 notation can be performed using standard 2's complement adder without any modifications. Q7.8 multiplication can be performed using standard fixed point multiplication, but at the end of multiplication the 32-bit result has to be shifted right by 8. Then the 16 bits from the middle of 32-bit result are taken, and other bits are discarded.

$$\begin{aligned} \text{-}0.5 * 1.375 &= 11111111.10000_{b(Q7.8)} * 00000001.0110000_{b(Q7.8)} \\ &= 1111111\mathbf{11010100.00000000}00000000_{b(Q15.16)} \quad (4.1) \\ &= 11010100.00000000_{b(Q7.8)} \\ &= 0.6875 \end{aligned}$$

## 4.2 Support Vector Machine

The SVM classifier is implemented in C and compiled using gcc compiler. The SVM optimization problem (Equation 3.11) can be solved using Lagrangian multiplier method. The training set contains a set of feature vectors where each feature vector is a tuple of the form $\mathbf{X} = [\mathrm{x_1}, \mathrm{x_2}, \mathrm{x_{3,}}...\mathrm{x_n}]$ .Here $\mathrm{x_1}, \mathrm{x_2}, \mathrm{x_{3,}}...\mathrm{x_n}$ are the features.

### 4.2.1 Lagrange Multiplier method

First, a Lagrange function is defined using a set of non-negative Lagrangian multiplier [1], [3] $\alpha = \{\alpha_i\}$and $\beta = \{\beta_i\}$, as:

$$\begin{aligned} \mathrm{L_p} &= \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\mathrm{C}}{\mathrm{n}}\sum_{\mathrm{i=1}}^{\mathrm{n}}\xi_\mathrm{i} - \sum_{\mathrm{i=1}}^{\mathrm{n}}\alpha_\mathrm{i}\left(\mathrm{y_i}\left(\mathbf{w}^\mathrm{T}\mathrm{x_i} + b\right) - 1 + \xi_\mathrm{i}\right) \\ &\quad - \sum_{i=1}^{n}\beta_i\xi_i \end{aligned} \tag{4.2}$$

Next, the unconstrained minimum of the Lagrangian function $\mathrm{L_p}$ is computer with respect to $\mathbf{w}$,$b$ and $\xi_i$.

$$\frac{\delta \mathrm{L_p}}{\delta \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i \, y_i \, x_i = 0$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \, y_i \, x_i \tag{4.3}$$

$$\frac{\delta \mathrm{L_p}}{\delta b} = -\sum_{i=1}^{n} \alpha_i \, y_i = 0$$

$$\sum_{i=1}^{n} \alpha_i \, y_i = 0 \tag{4.4}$$

$$\frac{\delta \mathrm{L_p}}{\delta \xi_i} = \frac{\mathrm{C}}{\mathrm{n}} - \alpha_i - \beta_i = 0$$

$$\alpha_i = \frac{\mathrm{C}}{\mathrm{n}} - \beta_i \tag{4.5}$$

The dual of the Lagrangian function is,

$$\mathrm{L_D} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \alpha_{\mathrm{i}} - \frac{1}{2} \sum_{\mathrm{i}=1}^{\mathrm{n}} \sum_{\mathrm{j}=1}^{\mathrm{n}} \alpha_{\mathrm{i}} \, \alpha_{\mathrm{j}} \, \mathrm{y_i} \, \mathrm{y_j} \, \mathrm{x_i} \, \mathrm{x_j} \tag{4.6}$$

The minimum solution of the optimization problem can be obtained by maximizing the dual of the objective function.

$$\max_{\alpha} \mathrm{L_D}(\alpha)$$

$$\text{subject to } \sum_{i=1}^{n} \alpha_i \, y_i = 0, \quad 0 \leq \alpha_i \leq \frac{\mathrm{C}}{n} \ \ \forall i \tag{4.7}$$

Once $\alpha$ is obtained, the optimal classifier is given by,

$$f_{\alpha}(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i (x_i.x) + b\right) \tag{4.8}$$

The feature vectors for which $\alpha_i \neq 0$ are called support vectors.

In the SVM learning step, the optimal Lagrangian multipliers $\alpha$ that maximize the dual $L_D(\alpha)$are obtained using sequential minimal optimization (SMO) [6]. SMO decomposes SVM problem into smallest possible Quadratic programming problem and solves this subproblem analytically. At each step, SMO picks two Lagrangian multipliers according to some heuristic rules, optimizes the two multipliers jointly and updates SVM to reflect the optimal values. For this purpose, Karush–Kuhn–Tucker (KKT) conditions are used.

In the classification step, the unknown feature vector is classified based on the support vectors computed in the learning step.

# Chapter 5

# Experiments

The following 6 scenarios are simulated to identify some of the semantics of the traffic. The test setup for each of the scenarios is shown in Figures 5.1 to 5.4. Two sensing nodes, which are PSoC1, are used for sound localization. The distance between the two nodes, D, is 72 inches. Four different positions are considered to simulate the movement of vehicles. Experiments are done with 1, 2 and 3 sound sources to simulate different cluster size for different scenarios. First, the feature vectors are clustered into into different classes using SVM based clustering and then the clustered data is used to train the SVM classifier.

## 5.1 Single vehicle in favourable driving conditions

In this scenario, only one vehicle is tracked (Figure 5.1). By favorable driving conditions, we mean that there is no sudden change in the speed of the vehicle from one position to the other, that is, it remains constant. With

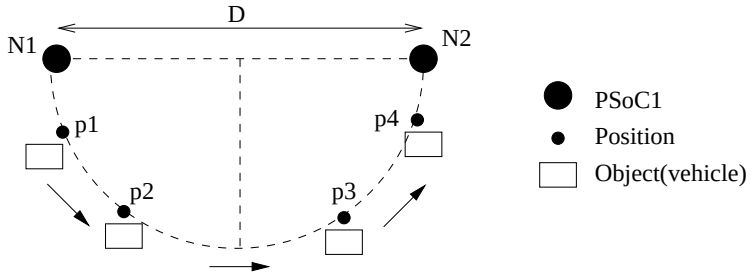


Figure 5.1: Simulation of a single vehicle

the distances from one position to the other known and by assuming the time it takes for the vehicle to move from one position to the other, speed is obtained.

## 5.2 Cluster of vehicles in favorable driving conditions

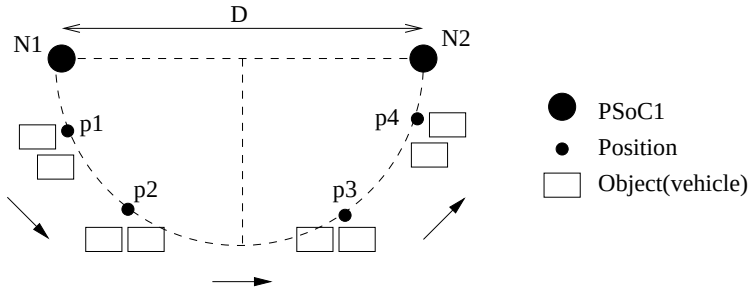


Figure 5.2: Simulation of a cluster of vehicles

A cluster of vehicle is simulated using multiple sound sources, one for each vehicle in the cluster (Figure 5.2). The favorable driving conditions are simulated as described in the previous section.

## 5.3 Single vehicle in bad driving conditions

This scenario is simulated as described in Figure 5.1 but in this case, the speed of the vehicle varies at different positions. In real environment, the variation of speed may be due to potholes, bad weather or road conditions. Even with varying speed, the classifier can identify the class of unknown feature vectors based on the intra-cluster object distance.

## 5.4 Cluster of vehicles in bad driving conditions

A cluster of vehicles in bad driving conditions is simulated using multiple sound sources (Figure 5.2). In this case, the speed of the cluster is changing with time but the speed of all the vehicles within the cluster is the same. Also, the intra-cluster vehicular distance remains same at different sampling positions.

## 5.5 A vehicle joining a cluster of vehicles

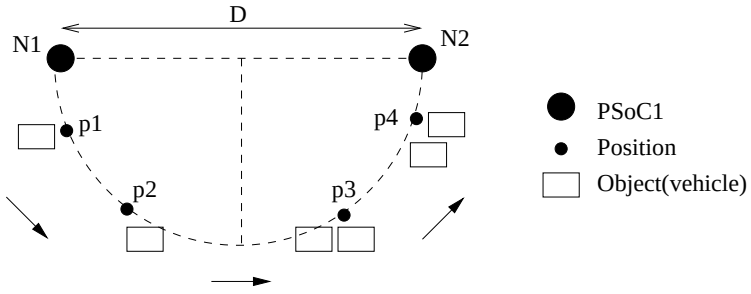


Figure 5.3: A vehicle joining a cluster

Figure 5.3 describes the scenario of a vehicle joining a cluster. A single vehicle is present at positions $p_1$ and $p_2$ and a new vehicle joins it at position $p_3$and creates a cluster of size 2. This is simulated using a single sound source at $p_1$and $p_2$ and two sound sources at positions $p_3$and $p_4$. Experiments are also performed with different cluster sizes.

## 5.6 A vehicle splitting from a cluster of vehicles

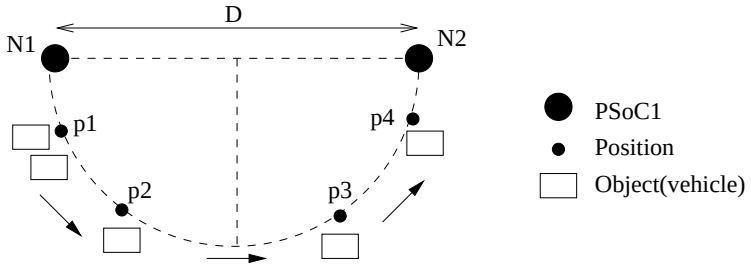


Figure 5.4: A vehicle splitting from a cluster

Figure 5.4 describes the scenario where a vehicle splits from the cluster. This is simulated using two sound sources at position $p_1$ representing two vehicles and a single sound source at positions $p_2$,$p_3$and $p_4$.

## 5.7 Results

The experimental results for sound-based localization, clustering and classification are discussed in this section.

### 5.7.1 Sound localization results

Table 5.1 gives the measured intra-cluster object distances for a cluster of size 3. The second column gives the distances between the first and the second object and the third column gives the distances between the second and third objects at four different positions. The actual $O_1$to $O_2$ and $O_2$to $O_3$ distance used in the experiment was 8 inches.

| Position | $O_1$to $O_2$ (inches) | $O_2$to $O_3$(inches) |
|---|---|---|
| 1 | 8.03 | 8.79 |
| 2 | 7.65 | 4.87 |
| 3 | 7.66 | 8.27 |
| 4 | 8.55 | 7.88 |

Table 5.1: Intra-cluster vehicular distance for a cluster of size 3

Table 5.2 gives the measured intra-cluster object distances for a cluster of size 2. The second column gives the distances between the two object. The actual $O_1$to $O_2$ was 16 inches.

| Position | $O_1$to $O_2$ (inches) |
|---|---|
| 1 | 16.30 |
| 2 | 12.00 |
| 3 | 13.90 |
| 4 | 14.04 |

Table 5.2: Intra-cluster vehicular distance for a cluster of size 2

The possible factors that resulted in localization errors are:

- As the distance between the microphone pair and the sound source decreases, the DoA estimates become coarser.

- Physical parameters such as speaker width and sensitivity of the microphone contributing towards measurement errors.

- Accuracy of experimental setup and error due to elevation of microphone and sound source.

#### 5.7.1.1 Execution time profiling of sound localization

The test setup for sound localization on PSoC is as shown in Figure 5.5 . The microphone pair is situated at least 1 m away from either wall and at least 1 m above the floor. This is done to reduce effect of reverberations. Readings were taken for values of DOA ranging from -90 to +90 in steps of 15 degrees.

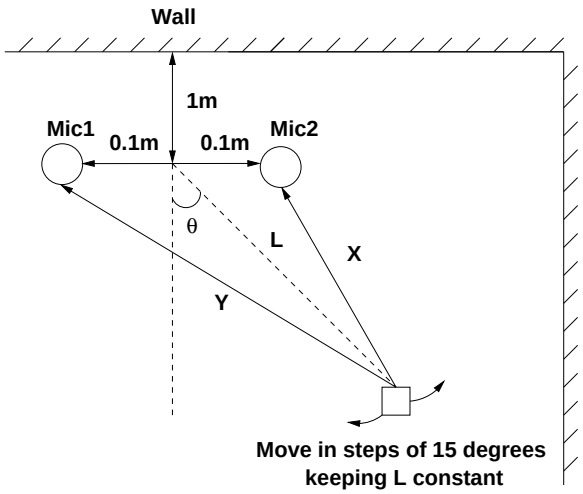


Figure 5.5: Test setup

Table 5.3 and Figure 5.6 give the comparison between floating and fixed point implementations of sound localization in terms of execution time. The

execution times of the performance-critical routines for computing FFT and FFT peak values is are greatly reduced with the fixed point implementation. The reduction in execution time is achieved due to faster fixed point computations when compared to floating point operations. Figure 5.7 compares the results of the fixed point and floating point implementations for localizing a sound source. The floating point implementation does slightly better than the fixed point implementation.

| Routine | Floating point ($\mu s$) | Fixed point ($\mu s$) |
|---|---|---|
| Hanning w. | 19867 | 8857 |
| FFT reorder | 10929 | 90 |
| FFT | 259938 | 110613 |
| FFT peak | 43450 | 2168 |
| CORDIC | 4470 | 580 |
| AOA | 882 | 882 |

Table 5.3: Execution Time

### 5.7.2 SVM-based Clustering and Classification results

The SVM-based clustering and classification algorithms are executed on the PC (Intel Core2Duo, 1.3GHz, 1GB RAM) and PSoC 5 (ARM Cortex core, 80MHz) and the respective execution times for the learning step are 3 ms and 13 ms. The clustering accuracy for a data set with 8 feature vectors used for the experiment is 87.5%. The classification accuracy in identifying the 6 scenarios is 100%.

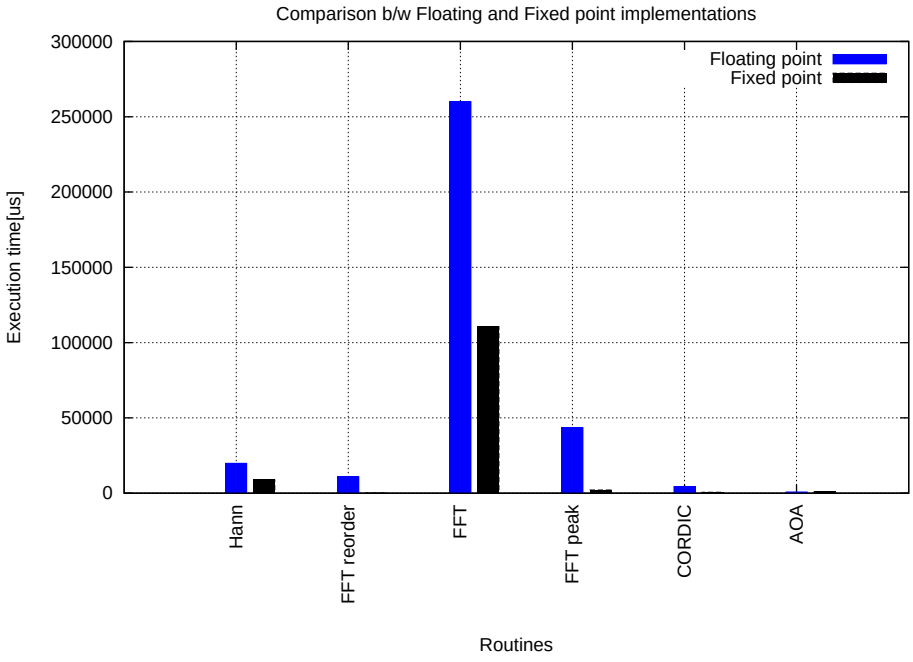


Figure 5.6: Execution time comparison plot

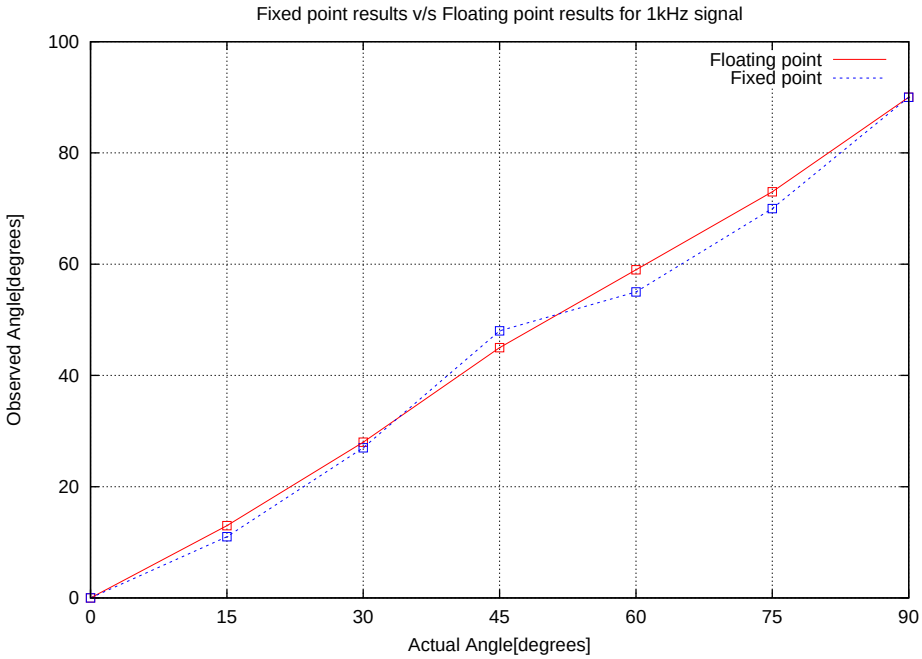


Figure 5.7: Results for fixed point and floating point implementations

# Chapter 6

# Summary

## 6.1 Related work

There has been relatively less emphasis on sound-based scene understanding [2], [10] as compared to its vision-based counterpart [5],[7],[9]. Also, there has been less work on implementing and experimenting the same realized as customized electronic designs. This is important to achieve low cost requirement which in turn is important for large-scale deployment of the application in a real-world environment.

Kim et al. [10] describe a method for audio scene understanding using topic models. The topic models are extensively used in text modeling applications. The work proposes to build an acoustic dictionary for scene understanding. The Mel frequency cepstral coefficients (MFCC) feature vectors are extracted and each of the feature vectors is classified based on a pre-trained acoustic word dictionary built using vector quantization.

Canton-Ferrer et al. [2] focus on activity detection and recognition based

on audiovisual data. The spectro-temporal audio features and localization features are used in conjunction with visual features for tracking, recognition and scene understanding.

## 6.2 Future Scope

In the work presented in this report, only the localization features and speed of the vehicle are extracted using which the clustering and classification is done. The other features described in sections 2.1 to 2.4 could give a better description of the scene and a lot more behaviors could be understood.

The BSEs described in section 1.1 define a simple ontology for traffic applications. They are the basic elements involved in traffic and are used for expressing the interactions and correlations in traffic scenes. Every particular traffic scene is a specific instance of the ontology. Based on this ontology, different traffic scenarios could be understood.

# Bibliography

[1] Christopher J.C. Burges. *A Tutorial on Support Vector Machines for Pattern Recognition.* Data Mining and Knowledge Discovery 2, 121-167, 1998.

[2] C. Canton-Ferrer, T. Butko, C. Segura, X. Giro, C. Nadeu, J. Hernando, and J.R. Casas. *Audiovisual Event Detection Towards Scene Understanding.* Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference, 2009.

[3] Corinna Cortes and Vladimir Vapnik. *Support-vector networks.* Machine Learning, 20:273–297, 1995.

[4] Alex N. Doboli and Edward H. Currie. *Introduction to Mixed-Signal, Embedded Design.*

[5] A. Ess, T. Mueller, H. Grabner, , and L. van Gool. *Segmentation-Based Urban Traffic Scene Understanding. BMVC*, September 2009.

[6] Yihong Gong and Wei Xu. *Machine Learning for Multimedia Content Analysis.*

[7] Martin Heracles, Fernando Martinelli, and Jannik Fritsch. *Vision-Based Behavior Prediction in Urban Traffic Environments by Scene Categorization.* BMVC, 2010.

[8] Paul Holden. *Develop FFT apps on low-power MCUs.* Embedded Systems Programming, 2005.

[9] Seung-Bin Im, Keum-Sung Hwang, and Sung-Bae Clio. *A Bayesian Network Framework for Vision Based Semantic Scene Understanding.* Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium, 2007.

[10] Samuel Kim, Shiva Sundaram, Panayiotis Georgiou, and Shrikanth Narayanan. *Audio Scene understanding using topic models.* Proceedings of the Neural Information Processing Systems (NIPS) Workshop.

[11] Dongge Lia, Ishwar K. Sethi, Nevenka Dimitrovac, and Tom McGeec. *Classification of general audio data for content-based retrieval.* Pattern Recognition Letters 22, 2001.

[12] Ingo Mierswa and Katharina Morik. *Automatic Feature Extraction for Classifying Audio Data.* Journal, 58:127149, 2005.

[13] Fabian Mörchen, Alfred Ultsch, Michael Thies, Ingo Löhken, Mario Nöcker, Christian Stamm, Niko Efthymiou, and Martin Kümmerer. *MusicMiner: Visualizing timbre distances of music as topographical maps.* 2005.

[14] George Tzanetakis and Perry Cook. *Musical Genre Classification of Audio Signals.* IEEE Transactions on Speech and Audio Processing 10(5), 2002.

[15] Anurag Umbarkar. *Improved Sound-based Localization Through a Network of Reconfigurable Mixed-Signal Nodes.* M.S.Thesis, Stony Brook University, 2010.

[16] Stephen Winters-Hilt and Sam Merat. *SVM clustering.* BMC Bioinformatics; 8(Suppl 7): S18., 2007.