

© 2014 Yunlong Gao

THE EMERGENCY TRANSCRIBER: A SITUATION-AWARE  
RECORDING SYSTEM FOR NOISY ACOUSTIC ENVIRONMENTS

BY

YUNLONG GAO

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Computer Science  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:

Professor Tarek Abdelzaher

# ABSTRACT

The thesis presents a novel situation awareness tool for sensing classification. We proposed a general scheme for sensing, and applied that to build an acoustic tool for teams of first responders and emergency personnel. It constitutes an audio interface for reliably recording and disseminating situation progress as extracted from the team’s audio communications. The tool that we built is intended for emergency teams operating in noisy acoustic environments, where standalone speech recognition systems fail to deliver desired accuracy. Such teams typically follow predefined collaborative workflow as dictated by the relevant engagement protocols, specifying their roles and communications. Given the critical nature of the situation, the vocabulary used is often constrained and dependent on the current stage of the workflow being executed. Treating a traditional speech recognition component as a noisy sensor, the novelty of our tool lies in exploiting knowledge of the workflow to correct the noisy measurements. The intellectual contribution in this exploitation lies in the joint estimation of the current state of the workflow together with the correction of sensed data, given only the noisy (speech) measurements and an overall workflow description. Evaluation shows that the tool provides a significant accuracy enhancement compared to the standalone speech recognition, effectively coping with the noisy environment of emergency teams.

*To my parents, for their love and support.*

# ACKNOWLEDGMENTS

This work would not be accomplished without the support of many people.

In particular, I would love to thank my advisor, Professor Tarek Abdelzaher, for his support, trust and instructions. He helped me in every aspect of this work, including the algorithm development, data analysis, simulation and paper writing. Whenever I came across some difficult situations during the research, he always provided me with insightful suggestions and encouraged me to think creatively and try different approaches.

I would also like to Thank Dr. Lu Su, who was senior Ph.D. student under Professor Abdelzaher. He acted as a big brother and helped me quite a lot in both research, study and life. I would also like to thank Shaohan Hu, who worked closely with me and gave me encouragement and help whenever I need them.

Last but not the least, I would like to thank many other people who also joined this project and made it happen: Professor Lui Sha, Dr. Richard Berlin, Mr. Renato Mancuso, Mr. Minje Kim, and Mr. Po-Liang Wu.

# TABLE OF CONTENTS

LIST OF ABBREVIATIONS . . . . .	vi
CHAPTER 1 INTRODUCTION . . . . .	1
1.1 Situation Awareness and Workflow . . . . .	1
1.2 Situation-Awareness in Emergency Environment . . . . .	2
1.3 Speech Recognition in Noisy Environment . . . . .	4
1.4 Thesis Contribution . . . . .	5
1.5 Thesis Organization . . . . .	5
CHAPTER 2 PROBLEM DESCRIPTION AND SYSTEM MODEL	
OVERVIEW . . . . .	6
2.1 Problem Description . . . . .	6
2.2 System Model Overview . . . . .	7
CHAPTER 3 MATHEMATICAL FORMULATION . . . . .	12
3.1 Basic Algorithm . . . . .	12
3.2 Enhancement to Missing Measurements . . . . .	16
3.3 Practical Issues . . . . .	20
CHAPTER 4 SIMULATION . . . . .	21
4.1 Simulation Settings . . . . .	21
4.2 Simulation Results . . . . .	22
CHAPTER 5 CASE STUDY EVALUATION . . . . .	29
5.1 Experimental Settings . . . . .	29
5.2 Experimental Results . . . . .	32
CHAPTER 6 RELATED WORK . . . . .	37
6.1 Sensing Classification . . . . .	37
6.2 Hidden Markov Model . . . . .	37
6.3 Speech Recognition . . . . .	38
CHAPTER 7 CONCLUSION . . . . .	39
REFERENCES . . . . .	40

# LIST OF ABBREVIATIONS

ASR	Automatic Speech Recognizer
HMM	Hidden Markov Model
CPR	Cardiopulmonary Resuscitation
EPI	Epinephrine
ED	Emergency Department

# CHAPTER 1

## INTRODUCTION

### 1.1 Situation Awareness and Workflow

Situation awareness is defined as the ability of an actor to understand the context of its activity. While humans are able to continuously integrate background data to sensory inputs and easily understand their context, machines are still a long way apart from mastering this ability. This is in largely because what determines the context (situation) are subtle changes in variables that cannot be directly and accurately sensed, but needs to be largely inferred. Situation awareness is fundamental in all those domains where it is crucial to understand the state of the environment and predict its changes in the near future. Common examples are: military command and control [1], industrial plant operation [2], air traffic control [3] [4], and emergency services [5] such as health-care [6] and fire-fighting [7].

It is important to consider that in those contexts where situation awareness can be remarkably beneficial, human interactions do not evolve freely. In fact, domain-specific operators are trained to take actions according to a predetermined *workflow* that imposes a finite set of possible decisions and actions at each relevant change of state. We refer to workflow-based human interactions as *structured* human interactions. In addition to those critical domains mentioned above, a wide range of situations evolve according to structured human interactions. In sports, for instance, players on the field follow precise strategies set by the coach during the game and repeatedly trying to achieve the scoring goal; a similar model applies to card/board games where card holders adjust and actuate his/her strategy according the situation in the game; in business/political meetings, topics are discussed based on a tight schedule and a finite set of decisions can be taken for each of them.



In this work, we propose a general methodology to augment the capabilities of sensing devices to improve situation awareness of structured human interactions in noisy environments. Note that the noise hereby does not only refer to acoustic noise, but sensing-task-specific. For instance, it could be the visionary distortion when it comes to vision recognition task. Specifically, we exploit the additional information embedded in the domain-specific workflow to continuously correct the unreliable stream of sensor data. In particular, a number of sensing objects are embedded within a workflow, which consists of states/stages inside. And each state has certain possible objects associated with it, and transits to another abiding to certain probability. In another word, the workflow defines the possible range and the relationship between different objects.

Due to the noisy environment and the inherent unreliability of sensors, the sensing results cannot truly represent the reality of the sensing objects. In this work, we propose our scheme to exploit the workflow information to reconstruct the inaccurate sensing measurements. In this way, it is possible to produce instantaneous results that exhibit the maximum likelihood in terms of probability of being in state  $x$  given that a sensor has produced the output  $y$ . We show how the problem can be formulated using the Hidden Markov Model (HMM) and consider the emergency medical environments as a concrete application for our technique.

## 1.2 Situation-Awareness in Emergency Environment

In this section, we present a specific example of situation-awareness in the emergency department of hospitals and motivate the intuition of our user study.

While a patient is in an emergency department (ED) of a hospital, the surrounding environment can be quite noisy because of the oral communications between physicians, the utterance of the patients, the noise generated by medical devices, etc [8]. Various studies have shown that the sound levels in the emergency department are sufficiently high (on average between 61 and 69 dB(A)) [9] to raise concerns regarding the communication of speech without errors—an important issue everywhere in a hospital and a particularly critical issue in emergency departments (EDs) because of the fact that

doctors and nurses frequently need to work at an urgent pace and to rely on hearing and understanding each other's oral communication [10] [8].

Despite the chaotic and stressful characteristics of the emergence department, in such environment, physicians are surrounding the patient and performing medical operations following a known predefined collaborative workflow according the diagnosis to the patient, thus specifying their roles and communications. A transition in the workflow is determined by either a sudden change in the conditions of the patient or the interactions among physicians. Given the critical nature of the situation, the vocabulary used is often constrained and dependent on the current stage of the workflow being executed.

Unfortunately, however, since humans are responsible for taking the right decision at the right moment, mistakes are possible. Such mistakes can arise from a failure to correctly follow the workflow and/or due to misunderstandings among the physicians. It has been estimated that \$11,529 emergency department malpractice claims have been filled between 1985 and 2007 in United States, representing a total of \$664 millions in liability [11]. Each claim reports an event in the emergency department that has been identified as the cause of injury to an adult patient. Out of the total considered claims, 18% of the cases revealed that no evident mistake affected the conditions of the patient; 37% of the cases referred to diagnostic errors that caused injury to patients due to wrong or not timely treatment; and 17% highlighted that improper performance of procedures has caused damage to bones, internal organs or infections.

In the emergency department particularly, the current operation manner is that the head nurse keeps track of what the physicians have said [12]. Due to the chaotic nature of the working environment, and the human memory decay, it is possible that the nurse may miss recording the commands that has been executed by some physicians and thus lead to possible overdose and misoperation. It is easy to understand that being able to deploy a method to correctly track the workflow of an emergency situation would result in a valuable benefit for both patients and physicians. In fact, this could provide: (a) early detection of procedural mistakes; (b) accurate logging of events to either validate patients claims or support physician's defence from invalid claims; (c) a less error-prone interaction among physicians thanks to a centralized view of the followed steps in the procedural workflow.

### 1.3 Speech Recognition in Noisy Environment

Speech/voice recognition is a well-established realm that many academic endeavours have been invested into [13] [14] [15] [16]. However, due to the inherently chaotic acoustic environment, standalone speech recognition systems may fail to deliver an acceptable level of accuracy. There are also many other works which focus on building up resilient speech recognition systems in noisy environments, such as [17] [18]. Most of the existing work focuses on eliminating recognition mismatch probabilities via the methods of giving more priority to high signal to noise ratio (SNR) portions of the speech in decision making, exploiting class-dependent processing, and including auditory models in voice processing, etc.

In comparison to that, we propose to treating a traditional speech recognition component as a noisy sensor and to correct the noisy measurements relying on the workflow knowledge. The intellectual contribution in this exploitation lies in the joint estimation of the current state of the workflow together with correction of sensed data, given only the noisy (speech) measurements and an overall workflow description.

The challenge of this work lies in the fact that the measurements are inaccurate, while the states associated with the sensing results are hidden. We first model the state transition relationship via Hidden Markov Model [19]. Next, we use a confusion matrix to characterize the properties of the sensor and the sensing environment. We then describe an algorithm which is able to find the optimal sequence of *guessed data* (objects) so that not only we are able to (a) reconstruct the inaccurate sensing measurements, but also (b) keep track of the state transitions.

We first explore the advantages and limitations of this approach via simulation, where abstract workflow states are associated with objects and an "object sensor" is used to do classification based on features of objects. The sensor in the simulation has certain reliability and might cause errors. We then construct a user study in the emergency environment, applying one of the most common workflow – adult cardiac resuscitation algorithm [20] that occurs in emergency departments to show that our approach can recover correct data from noisy measurements by exploiting workflow topology. From our evaluations, it emerges that we are able to achieve an accuracy in the workflow state detection of about 80% while relying on a standalone sensor

with an accuracy of around 40%.

## 1.4 Thesis Contribution

The research topic of this work falls into cyber-physical systems. In particular, we explored possible solutions to increase the sensing accuracy given the sensing environment is noisy and the sensors are unreliable, and we built a real system in our user study to enhance the voice recognition accuracy in the physical world by using our scheme. To the best of our knowledge, our work is the first one to exploit the workflow information to achieve this goal.

We propose to exploit known workflow information to enhance the accuracy in the achievable situation awareness of structured human interactions. And the measurement in return can enhance the inference of the workflow state transition relationships in the workflow.

Moreover, we formulate the general problem as a maximum likelihood problem, which lays the foundation for the further analysis. And we propose an optimal algorithm to solve this problem, which enables us to find the most probable state transition sequence as well as the speech recognition results.

Last but not the least, we build a workflow-aware recoding system for noisy emergency environment that sensitively increases the achievable state-tracking abilities of state-of-the-art speech recognition techniques for emergency personnels.

## 1.5 Thesis Organization

The thesis is organized in the as follows. In Chapter 2, we model our system and then present the terminologies that are used in later sections. In Chapter 3, we formulate the problem rigorously and proposed our algorithm to find the optimal path with recovered sensing measurements. We explored the benefits and the limitations of our scheme in Chapter 4 via simulation, and we discuss the case study of emergency transcriber system in Chapter 5. We introduce the related work in Chapter 6 and we finally conclude this thesis in Chapter 7.

# CHAPTER 2

## PROBLEM DESCRIPTION AND SYSTEM MODEL OVERVIEW

In this chapter, we give a detailed description of the general problem we target on, and introduce the system model that we develop for tackling the problem.

### 2.1 Problem Description

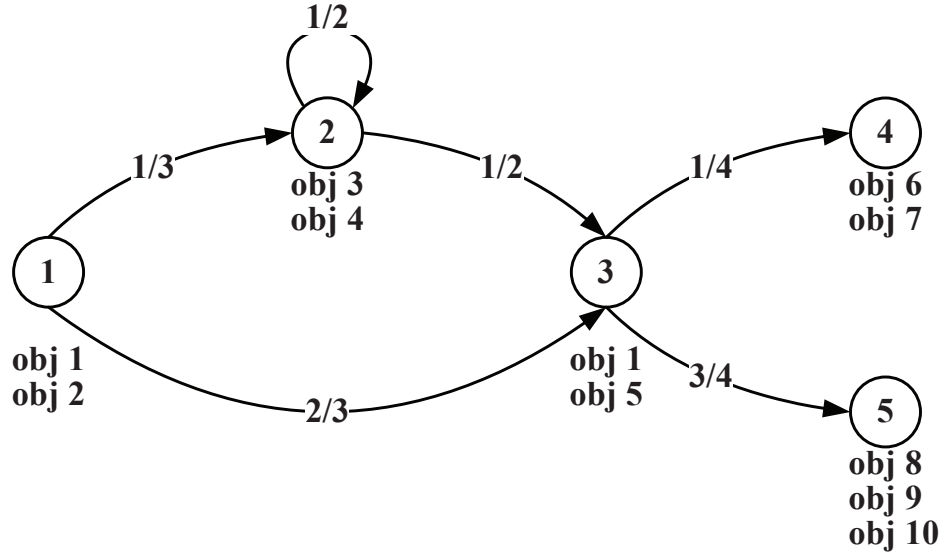


Figure 2.1: An example workflow

In this thesis, we target on the workflow-based sensing problems. Our target lies in two folds: on one hand, we try to augment the situation-awareness capability of structured human interaction, on the other hand, we aim to correct the unreliable sensing data in the noisy environment.

In order to enable the reader with better understanding of what the workflow looks like, we devise a simple example for clearer illustration. Fig. 2.1 shows, as a graph, a simple workflow topology defined for a set of sensing

tasks. The nodes represent abstract states/stages as defined in the workflow, and they are hidden from the sensors. The task of the sensors is to measure the objects accurately while inferring the state transition. Each state is associated with certain objects, which are the classification targets of the sensor. Each state associates with the objects abiding to certain probabilities, which will be further explained in the next section. The directed edges indicate the transition probabilities among states. For instance, at *State1*, *Object1* and 2 are the potential classification targets for the sensor(s). Upon completion at *State1*, the sensing task transits to *State2* (or 3) with probability 1/3 (or 2/3), respectively.

The goal of the sensing tasks is to correctly classify the objects, as well as keeping track of the sequence of states traversed. Let us take a sample sequence of objects as an example. Suppose the sensing results come from the standalone sensor is *Object2*, *Object1*, followed by *Object3*. However, this sequence cannot be met under the constraint of the workflow, since no state transition generates from *Object1* to *Object3*. This error could be caused by the sensor unreliability or the noisy environments which interrupts the sensing classification. The hypothesis example illustrates the motivation of our work, namely, we would like to make use of the workflow information to correct the unreliable sensing results and keep track of the sequence of states those have been traversed.

The challenges stem from the fact that the environments within which the sensing tasks are carried out are usually noisy, and that sensor themselves have inherent unreliability, potentially leading to incorrect decisions for individual classification tasks. It is also challenging to keep track of the state traversal when there is overlapping of objects between different states, since it generates confusion for differentiating them. For example, both *State1* and *State3* have the potential to generate *Object1*. Even though we assume that the sensing result is accurate, without extra information, we cannot tell which state the current object belongs to.

## 2.2 System Model Overview

In this section, we formulate the above problem as an optimization program. Next we give a high-level overview of our system model.

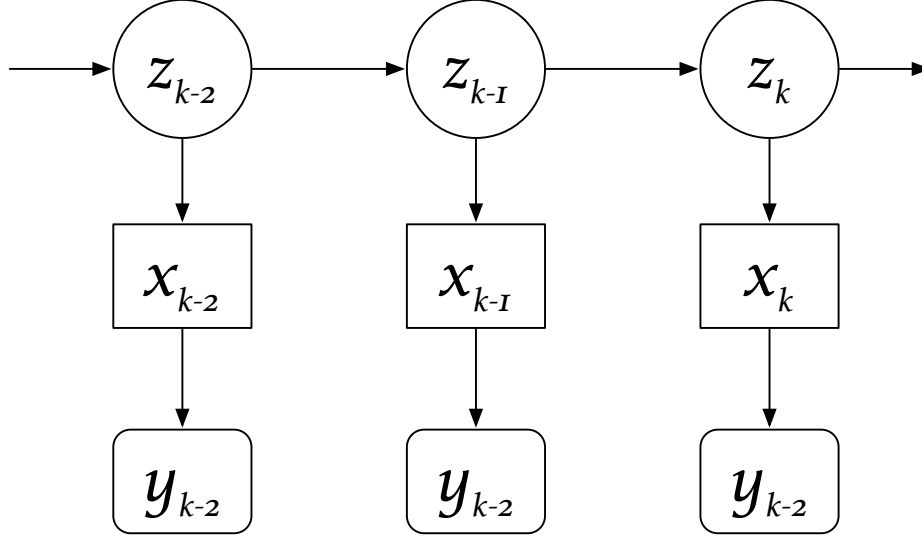


Figure 2.2: System Model

For clearer presentation, we first introduce our notations and terminologies. The workflow is represented as a directed graph with a set of states,  $\mathbf{s} = s_1, \dots, s_N$ , where  $N$  is the total number of states. Notice that the transition of the states is not directly visible to the sensor or the human beings. But the outputs of the states, namely, the objects are visible. Each state has certain probability distribution over the possible output objects. Therefore the sequence of the objects provides some hints about the state transition.

The above mentioned characteristic of the state transition fits well with the requirement of Hidden Markov Model (HMM) [19]. HMM has been widely used in modeling workflows and the emission procedures in machine-aided human translation [21], cryptanalysis [22], and time series [23]. Given the workflow, the transition probability distribution is  $t_{i,j} = p(z_{t+1} = s_j | z_t = s_i)$ , where  $\mathbf{z} = (z_1, \dots, z_T)$  is a state transition sequence for the time  $t = 1, \dots, T$ , as the circles in Fig. 2.2 shows.

However, despite the fact that the state transition could be modeled by Hidden Markov Model, the real challenging part of this work has not been tackled yet, i.e., the noisy sensing environment and the inherent unreliability of the sensors. Therefore we are going to present our solution to that in the following paragraphs.

Notice that the state sequence is hidden, thus it is not possible for us to directly observe state transitions. However, the good news for us is that, at each state, an observation of the object associated with that specific state

is acquired, denoted as  $y_i$ , where  $i = 1, 2, \dots, T$ , represented by the rounded rectangles in the figure. It is important to notice that, due to the environment noise and the unreliability of the sensor, those observations may not correctly represent the true object values. Therefore, we utilize a confusion matrix to capture the relationship between the true objects and the observations, and we will explain in detail about the generation and the utility of the confusion matrix in the next section.

The total number of possible true objects is limited because of the restriction of the workflow, denoted by  $\mathbf{x} = x_1, \dots, x_T$ , represented by the rectangles. It is easy to understand that, in the real world, the structured human interactions have to follow certain rules and the total number of possible states is finite.

The probability that the true object  $x_t$  is observed at state  $s_j$  is called emission probability, denoted as  $e_{j,t} = p(x_t | z_t = s_j)$ . And it quantifies the relationship between the objects and the states, which provides important hints on inferring the hidden state sequence.

We use classification probability, denoted as  $c_{i,j} = p(y_i | x_j)$ , to measure the correspondence between the actual observation in the sensed results and the true object set. This classification probability is obtained from the confusion matrix of the sensor—a sensor’s performance can be characterized by a confusion matrix, which contains the information about actual and predicted classification results done by the sensor’s classification system. It is a specific table layout that visualizes performance of a sensing system. As Table 2.1 shows, the column of the confusion matrix stands for the instances in a predicted class, while each row represents the instances in an actual true class. In this confusion matrix, of the ten *Object1*, the sensing system predicted that three were *Object2*, and two were *Object3*, and only five of the *Object1* were correctly predicted. Similarly of the ten *Object2*, six of the instances were correctly predicted while the remaining instances were wrong, with two *Object1* and two *Object3* respectively. As we can see from the confusion matrix, all the correct guesses reside in the main diagonal of the table.

With the confusion matrix defined above, we can further define the classification matrix. The classification probability  $c_{i,j}$ , which is the element in the classification matrix, is essentially the precision of the system. And it is defined as the ratio  $\frac{\# \text{ True Positive}}{\# \text{ True Positive} + \# \text{ False Positive}}$ . Let us take the previous confusion matrix as a concrete example. For the instances who



were predicted as *Object1* by the sensing system, five of them were actually *Object1*, and two of them are essentially *Object2*, and one of them is essentially *Object3*. Therefore the total number of *true positive* and *false negative* are 5 and 3, respectively. Taking the above information into the equation, we can get that the classification probability  $c_{i,j}$  is 5/8. In the similar manner, we can fill the remaining blanks of the classification matrix as Table 2.1(b) shows.

Table 2.1: An example of confusion matrix and classification matrix

(a) An example of confusion matrix

		Predicted Class		
		Obj 1	Obj 2	Obj 3
Actual Class	Obj 1	5	3	2
	Obj 2	2	6	2
	Obj 3	0	2	7

(b) An example of classification matrix

		Actual Class		
		Obj 1	Obj 2	Obj 3
Predicated Class	Obj 1	5/8	3/11	2/11
	Obj 2	2/8	6/11	2/11
	Obj 3	1/8	2/11	7/11

Last but not the least, as we assume the state transitions to follow the Markov model, we have:

$$p(z_{t+1}|z_t, z_{t-1}, \dots, z_1) = p(z_{t+1}|z_t) \quad (2.1)$$

, which essentially states the fact that the current state only depends on the most recent previous state, and it has nothing to do with other previous states. This assumption meets the requirement of the situations in the real world. For instance, in [24], the physicians make the decision and move to the next state based on the current the reaction of the patient to the current state operations. Moreover, the emission probability satisfies:

$$p(x_t|z_t, z_{t-1}, \dots, z_1) = p(x_t|z_t) \quad (2.2)$$

, which reveals that the observation is state-dependent. And the observations of different states are independent with each other. This is also valid in the

real case [24] since the physicians only speak keywords according to the current stage in the operation.

# CHAPTER 3

## MATHEMATICAL FORMULATION

In this chapter, we introduce the core algorithm we proposed to solve the problem described in the previous chapter. We first build mathematical foundation of the algorithm, and then we propose an addendum to the basic algorithm to meet the requirement of a special situation. And Last but not the least, we discuss some of the practical issues when implementing the algorithm in the real world.

### 3.1 Basic Algorithm

With the previously defined notations and terminologies, we formulate the target problem as an optimization problem. And the solution to this problem lies in two folds: on the one hand, we would like to find the most probable sequence of states (represented by the state vector  $\mathbf{z}$ ), and on the other hand, we would like to have correct sequence of classified objects (represented by the object set  $\mathbf{x}$ ). That being translated to the mathematical form, and is equivalent to maximizing the posterior probability  $p(\mathbf{zx}|\mathbf{y})$ , based on inaccurate measurement of  $\mathbf{y}$ . More rigorously, we write our goal as follows:

$$\widehat{\mathbf{zx}} = \arg \max_{\mathbf{zx}} p(\mathbf{zx}|\mathbf{y}) \quad (3.1)$$

In general, solving this equation would involve an exhaustive search for all possible state sequences and the actual object set. It is not hard to see that the general method would rapidly become impossible due to the fact that the operations grow exponentially. The reasoning is as follows: it will first search through all the possible true objects, and find the possibility of misclassification according to the classification matrix, and then it searches through all the different states in the workflow to get the possibility of observing that particular object at that state based on the emission probability. And lastly,

it has to combine with the previous round of computation by taking the transition probability into consideration. Suppose the total number of states in the workflow is  $N_s$ , the total number of different objects is  $N_o$ , and the state transition has taken  $T$  steps. Then the exhaustive search would have the time complexity of  $(N_s \times N_o)^T$ . Thus, we propose the following method which can bring a significant reduction in terms of computational efforts.

Based on Bayes' theorem, the posterior probability could be expressed as below:

$$p(\mathbf{zx}|\mathbf{y}) = \frac{p(\mathbf{zxy})}{p(\mathbf{y})} \quad (3.2)$$

$$= \frac{p(\mathbf{y}|\mathbf{zx})p(\mathbf{zx})}{p(\mathbf{y})} \quad (3.3)$$

$$= \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{y})} \quad (3.4)$$

The emission probability, by using the independence characteristics of the true object set Eqn. 2.1 and Eqn. 2.2, can be written as follows:

$$p(\mathbf{x}|\mathbf{z}) = p(x_1, \dots, x_T | z_1, \dots, z_T) \quad (3.5)$$

$$= p(x_1 | z_1) p(x_2 | z_2) \dots p(x_T | z_T) \quad (3.6)$$

$$= \prod_{i=1}^T p(x_i | z_i) \quad (3.7)$$

In the similar manner, the probability of sequence of state transition  $\mathbf{z}$  is given by considering the Markov property, shown as below:

$$p(\mathbf{z}) = p(z_1, z_2, \dots, z_T) \quad (3.8)$$

$$= p(z_T | z_{T-1}, \dots, z_1) \dots p(z_1 | z_0) p(z_0) \quad (3.9)$$

$$= \prod_{i=1}^T p(z_i | z_{i-1}) p(z_0) \quad (3.10)$$

where  $p(z_0)$  is the initial probability given as prior knowledge.

Furthermore, based on the classification probability discussed in Chapter 2, the conditional probability for the objects that are measured is:

$$p(\mathbf{y}|\mathbf{x}) = p(y_1, y_2, \dots, y_T | x_1, x_2, \dots, x_T) \quad (3.11)$$

$$= p(y_1 | x_1, \dots, x_T) \dots p(y_T | x_1, \dots, x_T) \quad (3.12)$$

$$= \prod_{i=1}^T p(y_i | x_1, \dots, x_T) \quad (3.13)$$

With the above analysis in mind, and the fact that the posterior probability is proportional to its numerator,  $\widehat{\mathbf{z}}\mathbf{x}$  in Eqn. (3.1) could be written as follows:

$$\arg \max_{\mathbf{z}\mathbf{x}} \left[ \prod_{i=1}^T p(x_i | z_i) \prod_{i=1}^T p(z_i | z_{i-1}) \prod_{i=1}^T p(y_i | x_1, \dots, x_T) p(z_0) \right]$$

We denote the final result of the above equation as  $\mu_T(z_T, x_T)$ ., thus we can rewrite it as:

$$\mu_T(z_T, x_T) = \arg \max_{z_{1:T}, x_{1:T}} \left[ \prod_{i=1}^T p(x_i | z_i) p(z_i | z_{i-1}) p(y_i | \mathbf{x}) p(z_0) \right] \quad (3.14)$$

$$= \arg \max_{z_{1:T}, x_{1:T}} \left[ \prod_{i=1}^{T-1} p(x_i | z_i) p(z_i | z_{i-1}) p(y_i | \mathbf{x}) p(z_0) \right] \quad (3.15)$$

$$p(x_T | z_T) p(z_T | z_{T-1}) p(y_T | \mathbf{x}) \quad (3.16)$$

$$= \arg \max_{z_{1:T}, x_{1:T}} [\mu_{T-1}(z_{T-1}, x_{T-1}) p(x_T | z_T)] \quad (3.17)$$

$$p(z_T | z_{T-1}) p(y_T | \mathbf{x}) \quad (3.18)$$

It is easy to see from the above equation that  $\mu_T(z_T, x_T)$  can be expressed by the its previous term  $\mu_{T-1}(z_{T-1}, x_{T-1})$ , which motivates the following algorithm that can find the optimal state transition sequence as well as the true object sequence. It is easy to show that the algorithm we proposed can maximize the posterior probability of  $p(\mathbf{z}\mathbf{x}|\mathbf{y})$ .

The basic idea of the proposed algorithm stem from 3.18, which reveals the essence of dynamic programming of the algorithm. The input to the algorithm are: the initial probability  $p_i$ , the observation sequence  $\mathbf{y}$ , and the prior knowledge of the workflow, which is represented by the transition probability  $t_{i,j}$ , the emission probability  $e_{i,j}$ , as well as the classification matrix. The output of the algorithm is the sequence of the most possible state

---

**Algorithm 1** find optimal  $\mathbf{z}$  and  $\mathbf{x}$ 

---

```
1: for  $i = 0; i < N_s; i++$  do
2:   initialize  $V[0][i]$ ,  $B[0][i]$ ,  $X[0][i]$ ;
3: end for
4: for  $t = 1; t \leq T; t++$  do
5:   for  $i = 0; i < N_s; i++$  do
6:      $p_{max} = 0$ ;
7:     for  $m = 0; m < N_v; m++$  do
8:       if  $t == 1$  then
9:          $p \leftarrow p_{ie_{x_m,i}}p(y_t|x_m)$ 
10:      else
11:        for  $j = 0; j < N_s; j++$  do
12:           $p \leftarrow V[t-1][j]t_{j,i}e_{x_m,i}p(y_t|x_m)$ 
13:          if  $p > p_{max}$  then
14:             $p_{max} \leftarrow p$ 
15:             $s_{max} \leftarrow j$ 
16:             $v_{word} \leftarrow x_m$ 
17:          end if
18:        end for
19:      end if
20:    end for
21:     $V[t][i] \leftarrow p_{max}$ 
22:     $B[t][i] \leftarrow s_{max}$ 
23:     $X[t][i] \leftarrow v_{word}$ 
24:  end for
25: end for
```

---

transition sequence and the true object sets. The pseudo code is shown as follows.

In this algorithm,  $V[i][j]$  stands for the maximum overall probability of arriving at *Statej* at time  $i$ . And  $B[i][j]$  stands for the previous state that can transit to *Statej* and generate the overall possibility of  $V[i][j]$ .  $X[i][j]$  stores the corresponding true object corresponding to  $V[i][j]$ . Those three matrices are initialized at the beginning of the algorithm, and updated in the later iterations. The key to reducing the computational efforts lie in the fact that, at each time step, we keep only the path of states coming from the previous state that has the largest probability. And since the probability stores at  $V[i][j]$  is the maximum cumulative probability achieved so far in the transitioning to *Statej*, it also ensures the optimality of the algorithm.

At any time step  $t$ , we first explore the probability of generating the actual

observation based on the true object, and it is accomplished by referring to the sensors' classification matrix. We then explore all the possible transitions that lead to the state that we are currently sitting at. And we store a list of most probable transitions and the true objects in  $B$  and  $X$  respectively (for the backtracking purpose) before we move on to the next step. We put the above-mentioned two steps of exploration together, and build a series of sequence from the beginning to the end of the states. When we get to the end, we select the final sequence that generates the highest likelihood and move backwards, following the transitions with the highest probability until we arrive at the beginning. Corresponding, we also trace back the true objects that are observed along the path, and therefore getting the final results of optimal state transition sequence and object sequence. Notice that when  $t$  is equal to 1, meaning that we just get started from the very beginning, the update of the overall probability needs special treatment. Specifically, the transition probability  $t_{j,i}$ , was replaced by the prior knowledge of initial probability  $p_i$ , since there is no previous state, and thus the transition probability does not apply in here.

Also it is worthy of noticing that the time complexity of this algorithm is  $O(N_s^2 T N_v)$ , where  $N_s$  stands for the overall state space,  $T$  stands for state sequence length, and  $N_v$  represents the vocabulary space. When we implement the algorithm, we made several adjustment though. For instance, we use the log likelihood to avoid the underflow problem.

## 3.2 Enhancement to Missing Measurements

The above section presents the basic algorithm that can meet the requirement of correcting inaccurate sensing results of standalone sensors as well as keeping track of the state transition of the workflow. However, in the real world, more challenging situations might take place. For example, in a figure recognition sensing task, the sensor may suddenly stop working due to mechanical malfunction. Another example might be the speech recognition task in the emergency room where the doctors may forgot to speak out about their operations occasionally due to the heavy workload of their work. It could also be possible that the doctor speaks so lightly that the automatic speech recognizer (ASR) does not capture his/her voice. All the above men-

tioned situations will lead to one consequence, namely, the measurement, based on which our basic algorithm keeps track of the workflow and enables the correction of the sensing data is missing. In this section, we propose an improvement addendum, which could be combined with our basic algorithm to solve this problem.

The consequence of missing of measurements could be reflected on the change of transition probability between different states of the workflow. Take Fig. 2.1 as an example, in which *state1* transits to *state3* with the probability of 66.67%, and *state1* cannot transit to *state4* directly. For the sake of simplicity, we assume that the sensors are perfect. Suppose the previous state is *state1*, with the *Object1* generated and correctly classified. The current operation is at *state3*, and the object that has been generated, e.g. *Object5*, is missing due to one of the causes mentioned above. And then the workflow reaches *state4*, with one of the objects 6 observed and classified by the sensor. Therefore, the overall output from sandalone sensor is: *Object1* and *Object6*; implying that *state1* transits to *state4* directly, which is contradictory to the predefined workflow. If we use the basic algorithm solely, it will try to match *Obejct6* to the objects in *State2* and *State3* and thus give erroneous classification result. Motivated by the above scenario, we propose our solution that tackles this problem by preprocessing the transition probability before running the basic algorithm.

Before we describe our detailed method, we first present some terminologies and our assumption. Suppose the probability that an object is missing at each state of the workflow is  $p$ . We also assume that at each state, at most one object will be missing, i.e., continuous missing of objects at each state does not happen. We make this assumption based on the following two reasons. First, in the normal sensing environment, where critical sensing tasks take place, the sensors in the system should have basic reliability guarantee, the missing of the objects should not happen too frequently. Moreover, our approach focuses on the most fundamental case, and it can provide insight for other cases where multiple objects are continuously missing at each state.

Based on the above analysis and assumptions, we adapt the transition probability according to the following equation:

$$t'_{i,j} = t_{i,j} - t_{i,j}p + \sum_{m \in M} t_{i,m}pt_{m,j} \quad (3.19)$$



where  $t'_{i,j}$  represents the updated transition probability, and  $M$  stands for the set of source node of all the incoming edges of node  $j$ . The first term on the right hand side of the equation indicates the traditional transition probability from state  $i$  to  $j$  based on the domain protocol, which is prior knowledge. The second term indicates the probability that the workflow transits from  $i$  to  $j$  without observing an object associated with  $j$ . And the last term indicates the probability that  $i$  transits to  $m$  but the word associated  $m$  is missing at the previous round, and then  $m$  transits to  $j$  at the current round, where  $m$  has direct transition link to  $j$ . After updating the transition probability matrix as Eqn. (3.21), we can combine it with our basic algorithm to cater the situations where missing measurements happen. In theory, the combined algorithm can tackle the scenario where measure missing happens every other object, which is rare enough in practical scenarios.

Moreover, it is worthy of proving that the previous equation from a more rigorous perspective. From the definition of the transition probability matrix, we know that the  $i$ th row of the matrix represent the probability that *State* $i$  transits to any other *State* $j$ , where  $j = 1, 2, \dots, N$ , and  $N$  is the total number of states. It is easy to see that the sum of the elements in one row adds up to 1 since that row concludes all the possibilities of the outgoing edges. We can summarize this into the formulation as follows:

$$\sum_{j=1}^N t_{i,j} = \sum_{j \in M} t_{i,j} = 1 \quad (3.20)$$

We can then argue that the updated transition probability matrix remains correct if for each row, the sum of the elements of that row adds up to 1, which is shown as follows:

$$t'_{i,j} = t_{i,j} - t_{i,j}p + \sum_{m \in M} t_{i,m}pt_{m,j} \quad (3.21)$$

$$\sum_{j=1}^N t'_{i,j} = \sum_{j=1}^N t_{i,j} - \sum_{j=1}^N t_{i,j}p + \sum_{j=1}^N \left( \sum_{m \in M} t_{i,m} p t_{m,j} \right) \quad (3.22)$$

$$= 1 - p + \sum_{j=1}^N \left( \sum_{m \in M} t_{i,m} p t_{m,j} \right) \quad (3.23)$$

$$= 1 - p + \sum_{m=1}^N \left( \sum_{j=1}^N t_{i,m} t_{m,j} p \right) \quad (3.24)$$

$$= 1 - p + \sum_{m=1}^N t_{i,m} p \left( \sum_{j=1}^N t_{m,j} \right) \quad (3.25)$$

$$= 1 - p + p \quad (3.26)$$

$$= 1 \quad (3.27)$$

One might still argue that what if multiple continuous missing happens. The solution to that problem is described in the next section.

### 3.3 Practical Issues

In this section, we present some of the practical issues when using our algorithm in real sensing environment to do the workflow tracking and sensing result correction.

One real scenario that could happen is that multiple continuous missing of measurements. Based on the analysis in Section 3.2, we can see that the missing of the measurements essentially contradicts with the workflow information, because it breaks the transition relationship between different states in the workflow. In other words, the previous states that has been recognized and recorded has become a negative factor, trying to match the object to a wrong state. Since when the continuous missing makes the workflow hinder rather than help us to achieve our goal, we should abandon it when that happens. Therefore, in practice, we can set up a threshold of time for the detection of missing measurements. If the sensor does not provide output longer than this threshold while the operations are on, it indicates that continuous missing happens. And then, the history of state tracking is refreshed, and the algorithm runs all over again when a newly classified object comes out of sensor.

# CHAPTER 4

## SIMULATION

In this section, we evaluate the performance of the schemes proposed in the thesis. We first introduce our experiment settings, and then present the results.

### 4.1 Simulation Settings

We consider two types of workflow in our simulation, namely, random directed graph and directed tree, as they represent two distinct types of workflow in the real world. A random graph represents a workflow where loops could be possible. For instance, the workflow for intubation and airway management algorithm [25], and the workflow for adult cardiac arrest algorithm [20]. Another type of workflow could be abstracted as a directed tree where circles are not present, such as the workflow for ventricular failure with cardiogenic shock [26], and the workflow for bradycardia algorithm [26]. In our simulation, we simulated the workflow topology by generating two typical random topologies as stated above. Each state of the workflow, which represented by a node in the topology, is associated with some objects, whose values are sensed by sensors. For the simulation purpose, without particularly mentioned, we chose a path and some associated objects randomly from the topology as the ground truth. The comparison metric is the *fidelity*, which indicates the percentage of right classified results among the ground truth.

Notice that there are four different legends in the following figures. *sensor object* and *modified object* represent the object fidelity of the sensor and our scheme, respectively. *sensor state* and *modified state* represent the state fidelity of the sensor and our scheme, respectively. As mentioned in the above chapter, the classification matrix between the observation and true objects is

very important when doing object recovery and state tracking. For the simulation, the classification matrix is generated in the following way. Suppose the recall, which is defined as the ratio,  $\frac{\# \text{ True Positive}}{\# \text{ True Positive} + \# \text{ False Negative}}$  of each object is  $r$ . For simplicity, we assume that for a single simulation setting, the recall is the same. Therefore, the diagonal of the confusion matrix is identical, which is  $r$ , and the remaining entries of the same row of the matrix will be evenly populated with  $\frac{1-r}{N-1}$ , where  $N$  is the total number of objects. The detailed simulation results is presented in the following section.

## 4.2 Simulation Results

**Random Graph based Workflows:** We first study the sensing fidelity result on random graphs. We study the system performance impact from four different aspects: raw sensor classification reliability, path length of the ground truth task workflow, average node degree of the graph, and the number of nodes in the graph.

The baseline sensor in the simulation does not use workflow information at all, it is merely based on the classification matrix to do classification of the sensing objects. Even the standalone sensor does not have the functionality of state tracking, we add the state tracking fidelity to the sensor to compare that with our scheme. In order to make it a fair comparison, we assume that the sensor already knows the workflow and makes the judgement based on a naive way that the current state is just the the state that object belongs to.

First, we take a look at how the reliability of raw sensors affects system performance, where we use the classification recall as a measure of raw sensor reliability. The experiment setting in this set of experiment is as follows: the total number of nodes is 30, the average in/out degree of a node is 3, and the path length, which is defined as the number of node in the path, is 8. The average object per node is 5. The results are shown in Fig 4.1. First and foremost, more reliable sensors lead to better system performance, as expected. We also observe that our proposed approach of taking advantage of workflow information brings consistent improvement across all sensor reliability settings, up to about 20% as shown in the results. Evidently, our approach would have a hard time improving system performance when the raw sensor reliability is too low (e.g., with only about 10% recall) or too high

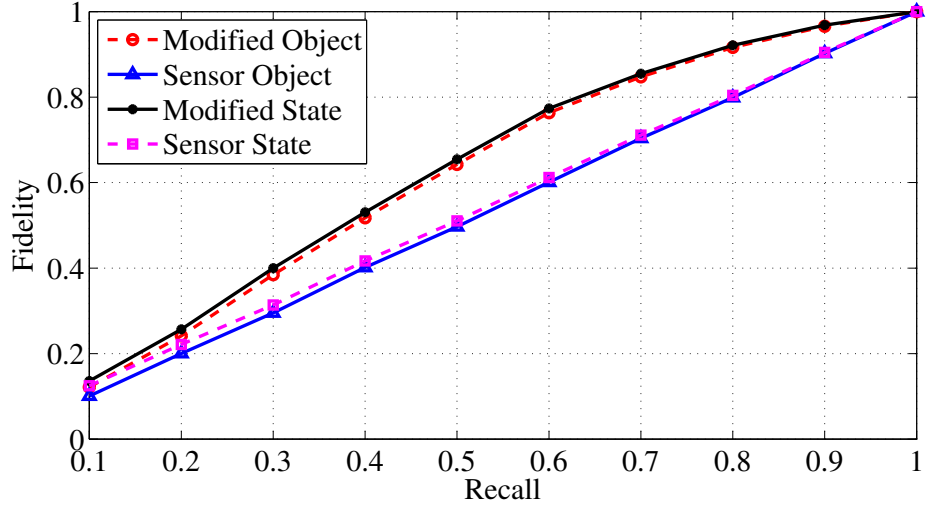


Figure 4.1: Impact of sensor classification reliability on system performance under graph based workflows

(e.g. sensors are perfectly reliable, saturating the system performance and leaving no room for improvements). We do, however, argue that in reality, “perfect” sensors rarely exist, and that system practitioners would normally utilize sensors that are reasonably reliable when building systems. Therefore, our proposed method will bring meaningful benefits to sensing systems in practice.

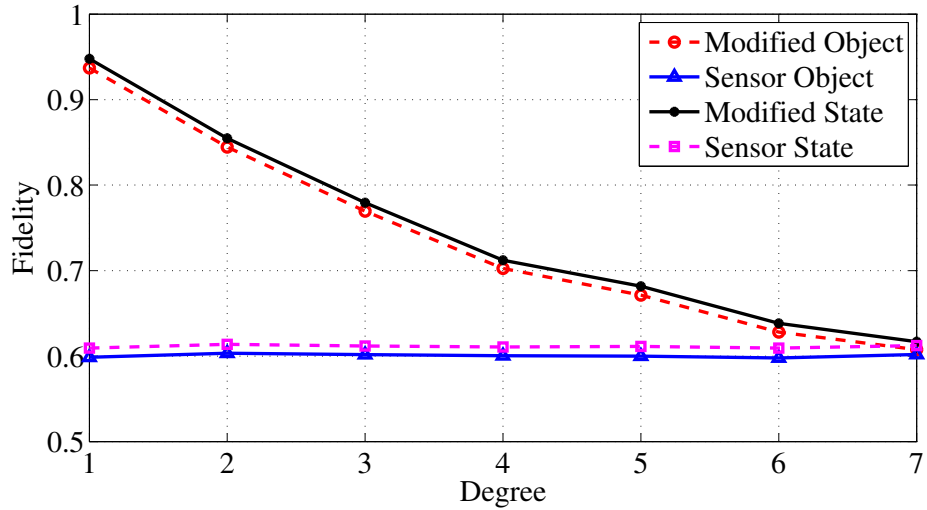


Figure 4.2: Impact of graph node degree on system performance under graph based workflows

Next, we study how system behaves when we vary the node degree in the

graph. The experiment setting in this set of experiment is similar to the previous one, which has 30 number of nodes, and the path length of 8. And we set the recall of the confusion matrix of the sensor to 0.6. The average object per node is 5. The results are shown in Fig 4.2. As can be seen, as it doesn't utilize the workflow information, the system performance in the baseline scheme remains at around 60%, unaffected by how the workflow structure changes. On the other hand, our proposed method achieves quite a advantage margin over the baseline in terms of system performance. This is because workflow topologies can provide extra information on how tasks are carried out, which our method takes into consideration and computes the constraints of where task progressions can move towards during execution. We do see that as the node degree increases, the effectiveness of our method shrinks. This is understandable because high node degree means higher connectivity in the graph, which in turn leads to less task progression constraints. An extreme case would be a fully connected graph, which would effectively provide no meaningful workflow information.

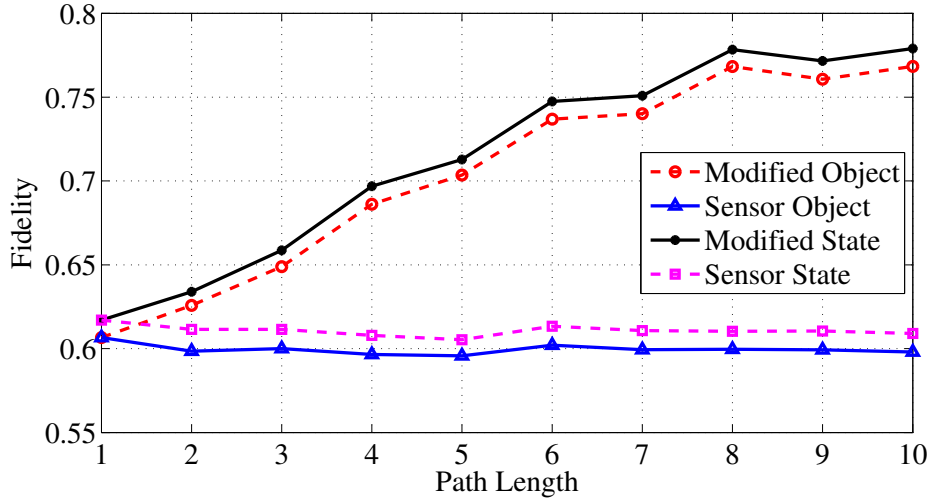
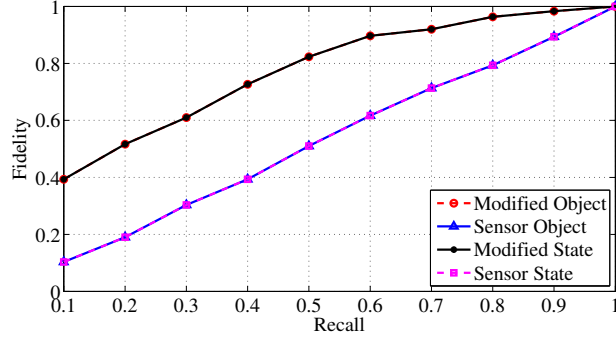
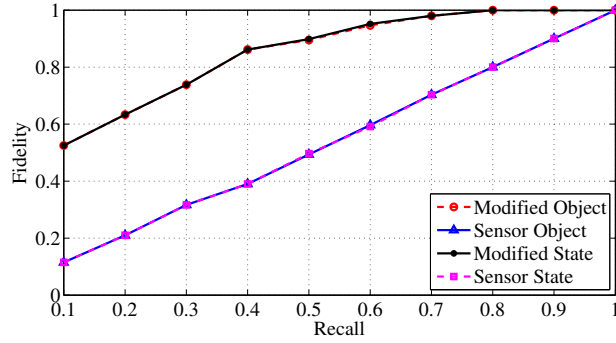


Figure 4.3: Impact of task path length on system performance under graph based workflows

We also look at how our method performs with varying task path lengths under the same workflow topology. The setting now becomes as follows: still the number of nodes in the graph is 30, the recall of the confusion matrix is 0.6, the degree is 3. The average object per node is 5. The results are shown in Fig 4.3. Similar to the previous experiment, the system performance under the baseline scheme does not change much as it does not take into considera-



(a) Task has random starting point.



(b) Task has known starting point (i.e., the root of the tree).

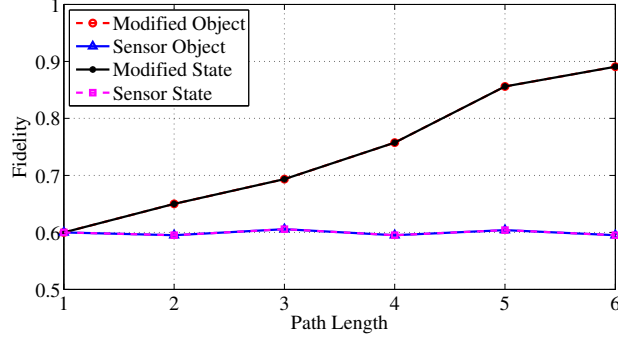
Figure 4.4: Impact of sensor classification reliability on system performance under tree based workflows

tion the workflow information in the first place. Our method, however, does benefit from taking advantage of the workflow information, clearly reflected in the result shown. We also observe that tasks corresponding to longer paths in the workflow topology tend to benefit more from our approach. This is because more matching constraints would be incurred when tasks last longer under a workflow, leading to higher error correction power in our method.

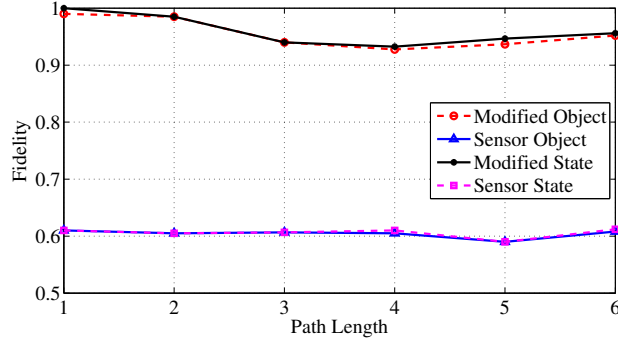
**Tree based Workflows:** We also study the sensing fidelity result on workflows whose topologies have tree shapes. We investigate the system performance impact from sensor classification reliability and path lengths. We also study the effect of whether or not the starting states of the tasks within workflows are known beforehand.

We first look at how the reliability of raw sensors affects system performance. The simulation setting for this experiment is as follows: The directed tree has the order of 3, and the height of 5. The number of objects per tree node is 5. The path length is 6.





(a) Task has random starting point.

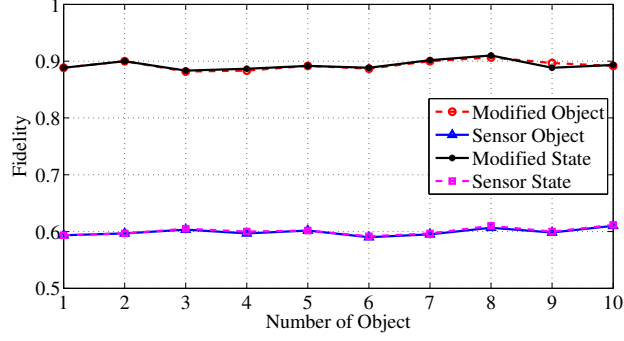


(b) Task has known starting point (i.e., the root of the tree).

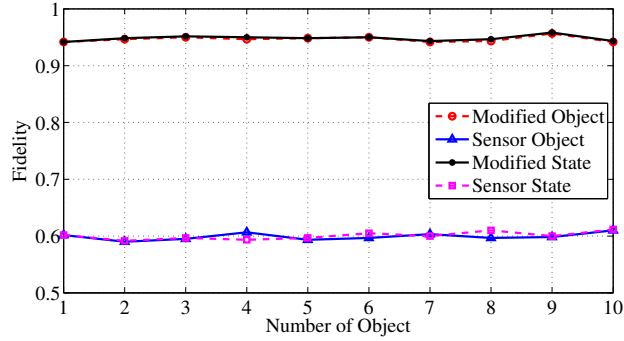
Figure 4.5: Impact of task path length on system performance under tree based workflows

The results are shown in Fig 4.4. We see the general trends of system behavior are similar to that of the graph based experiment result previously shown in Fig reffig:treeRecall. It can also be observed that, our scheme delivers better performance (compared to Fig. 4.5(a), Fig. 4.5(b) starts with higher fidelity score at low sensor reliabilities, and saturates sooner to perfect fidelity at high sensor reliabilities) when the starting point of the task is known (at the root of the tree), which is as expected.

Fig. 4.5 shows the experiment results of how task path lengths affect system performance. The simulation setting for this experiment is as follows: The directed tree has the order of 3, and the height of 5. The number of objects per tree node is 5. The recall of confusion is 0.6. We see that Fig. 4.5(a) is quite similar to the previously discussed graph based experiment (Fig. 4.3). However, if the starting point of tasks are known beforehand (fixed at the root of the trees in our experiment), we observe a slight V-shape in terms of the system performance under our proposed method; When the path length



(a) Task has random starting point.



(b) Task has known starting point (i.e., the root of the tree).

Figure 4.6: Impact of task object per node on system performance under tree based workflows

is extremely small, the prior knowledge of the starting point of the task path offers great help in terms limiting the search space in the workflow topology (i.e., only the tree-root vicinity will yield meaningful matching probabilities). The dominating effect of this prior knowledge decreases as the task path lengthens. But as path length keeps growing, the general workflow topology information will have an increasing constraining power that helps with path matching. Therefore, we observe the V-shape in the result figure.

Fig. 4.6 shows the experiment results of how number of object per tree node affect system performance. The simulation setting for this experiment is as follows: The directed tree has the order of 3, and the height of 5. The path length of the tree is 6. The recall of confusion is 0.6.

We see that Fig. 4.6(a) is quite different from the previously discussed graph based experiment (Fig. 4.3). One important conclusion we can draw from the simulation result is that the number of object per tree node does not have an impact on the sensing performance. However, if the starting point of

tasks are known beforehand (fixed at the root of the trees in our experiment), we observe a higher fidelity in terms of the system performance under our proposed method. Moreover, it is quite apparent that our scheme constantly outperforms the standalone sensors, therefore there are always a difference between the 'modified object fidelity' and the 'sensor object fidelity'.

# CHAPTER 5

## CASE STUDY EVALUATION

In this section, we apply our proposed scheme in the field of voice recognition under medical realm and develop a situation awareness tool for teams of first responders and emergency personnels. Experiment results using recorded audio data are presented and discussed.

### 5.1 Experimental Settings

**Workflow Information:** We chose adult cardiac arrest [24] as our case of study, as it strictly follows the emergency reaction algorithm, as shown in the Fig. 5.1. In realistic settings, when a patient is subject to cardiac arrest, multiple physicians and nurses are operating around the patient at the same time, and medical orders are vocally communicated, making the entire environment quite noisy, for which general-purpose standalone voice recognition softwares tend to perform poorly due to their error-prone nature.

**System component:** Our system consists of two major components. The first component is a standalone automatic speech recognizer (ASR), which we implemented using Google speech API [28]. And the web interface is available at the [29]. Its main functionality is that it acts as an audio interface for doing the initial recognizing of the medical team’s audio communications, as indicated by *R1* in Fig. 5.2. We also added additional functionality of making pauses and sending the recognition results to the server which is running the emergency transcriber component. Since the ASR does not have the workflow information, it tries to use the general language model and acoustic model embedded in the recognizer itself to match the signal it hears, which may lead to error recognition results, especially under noisy environment. One of the common mistake it makes is when it hears ‘epinephrine’, which is quite common under emergency situation when

it comes to resuscitate a cardiac patient, but very rare in general conversation. Since the ASR tries to make sense out of common language model, the recognizing result would usually be 'a friend of mine', which consists of common words and phonetically similar to the original 'epinephrine'. This phenomenon motivates our following solution.

The initial recognizing results  $R1$  are then fed into the second and most important component of our system – emergency transcriber as input. The emergency transcriber consists of two modules, i.e., the *keyword matching* module and the *word recovery and state tracking* module. And we will introduce their functionality in detail in the next section.

Since  $R1$  consists of sentences recognized by the standalone ASR, which does not take the workflow or the keyword information into consideration, the keyword matching module first applies keyword matching scheme in our emergency transcriber. The basic idea for this step is to find the most probable keyword that might occur in each sentence, which is equivalent to finding which keyword has the maximum number of overlapping phoneme characters with the heard sentence. We adopt the name of *convolution* to represent the maximum overlapping character. And the rationale behind is that the convolution measures the area overlap between two functions, which is similar to what we are trying to do with the phonemes. Since both  $R1$  and the keywords are in the form of English text, we need to convert them into phoneme representations before we do the matching since pure text cannot reveal the acoustic characteristics. We converted the  $R1$  and all the keywords into their phoneme representations using [30], a text synthesis software, and then calculate convolution according to the Algorithm 2:

In Algorithm 2,  $lenSrc$  and  $lenDst$  represent the length of the phoneme characters of the source word  $x$  and the destination word  $y$ , respectively.  $retval$  is the final return value of the overlapping phonetic characteristic. The basic idea of this algorithm is that we keep the destination word  $y$  unchanged, and we shift the source word and compare the shifted word with the destination word.  $step$  is the total maximum value of offset that the source word has. The algorithm loops through different values of offsets and sets add up the overlapping characters. It finally returns the possible maximum value.

Having been processed by the above algorithm,  $R2$ , which contains the keyword matching result of most probable keyword sequence, is then fed

---

**Algorithm 2** find maximum number of overlapping phonetic character between sentence  $\mathbf{x}$  and keyword  $\mathbf{y}$

---

```

1:  $lenSrc = x.length$ 
2:  $lenDst = y.length$ 
3:  $step = lenSrc + lenDst - 2$ 
4:  $retval = 0$ 
5: for  $i = 0; i < step; i++$  do
6:    $convIndex = 0$ 
7:    $startSrc = \max(0, lenSrc - 1)$ 
8:    $startDst = \max(0, i - lenSrc - 1)$ 
9:   while  $startSrc < lenSrc$  and  $startDst < lenDst$  do
10:    if  $x.substring(startSrc, startSrc + 1).equals(y.substring(startDst, startDst + 1))$  then
11:       $convIndex++$ 
12:    end if
13:     $startSrc++$ 
14:     $startDst++$ 
15:  end while
16:  if  $convIndex > retval$  then
17:     $retval = convIndex$ 
18:  end if
19: end for

```

---

into the second module in the emergency transcriber, namely, the *word recovery and state tracking* modules. We applied the general method described in Chapter 2 to this module. However, the speech recognition task has its own character that needs some special attention. Specifically, what makes the case study different from the general sensing scenario is the construction of the classification matrix. In the general sensing scenario, the classification matrix reveals the classification accuracy of a sensor, and it is obtained by training the sensor using large amount of training data, which is not available by the time we conducted our experiment. However, having been inspired by the above mentioned example of 'epinephrine', the probability that the ASR makes mistakes could essentially be measured by the phonetic similarity between different word phonemes. Therefore, instead of training the ASR to get the classification matrix, we can apply the following Eqn 5.1 to calculate the each element in the classification matrix.

$$sim(y_i|x_j) = (1 - \frac{LD_{i,j}}{\max(length(y_i), length(x_j))})conv(y_i, x_j) \quad (5.1)$$

where  $sim(y_i|x_j)$  represents the similarity between the observation word  $y_j$  and the true keyword  $x_i$ . And it is also the element in the  $i$ th row and  $j$ th column of classification matrix. Note that  $LD_{i,j}$  represents the Levensthein distance [31] between the phoneme representation of  $y_i$  and  $x_j$ . The  $length(y_i)$  and  $length(x_j)$  represent their phoneme length, respectively. The  $conv(y_i, x_j)$  represents the convolution of their phoneme representation, which captures the sub-phoneme overlapping between  $y_i$  and  $x_j$ . And it is calculated according to Algorithm 2.

## 5.2 Experimental Results

We recorded an episode of audio signals, which follows the workflow as presented in the Fig. 5.1. It is available for download from the source [32]. The sequence of states represented by the audio signals contains 21 sentences, and each sentence in the recording contains certain keyword according to the workflow. We fed the audio signals directly into the first component of our system, and the screenshot of the result of the standalone ASR,  $R1$ , is shown in the Fig. 5.3, where red circles means the words that are wrongly recognized. We then added noise with different amplitude to the original audio file, and sent them through the same pipeline as before. The final comparison result is shown in the figure below:

As we can see from the comparison results, when no noise is added to the signal, the accuracy of the standalone speech recognizer is 76.69%. And the first phase of keyword match can increase this word accuracy since the standalone ASR does not have the keyword information, and the phoneme similarity helps the system to enhance the its recognition accuracy to 81.2%. Moreover, with the workflow information taken into consideration, the word accuracy increased to 95.24%, and the state accuracy increase to 100%, which bolster the idea that workflow can help enhancing the sensing accuracy. When we add Guassian white noise to the original voice signals with a signal to noise ratio of 40dB, it is understandable that we the accuracy of

the word of  $R1$ ,  $R2$ , and  $R3$  are decreasing. But it is also clear that word matching can enhance the  $R1$  results and word recovery can enhance the  $R2$  even more. The situation is similar when the signal to noise ratio goes to  $33dB$ . The only difference, which is expected, is that the system overall accuracy decreases due to the environment noise, but still, our system can increase both the state-tracking and the word recovery accuracy.



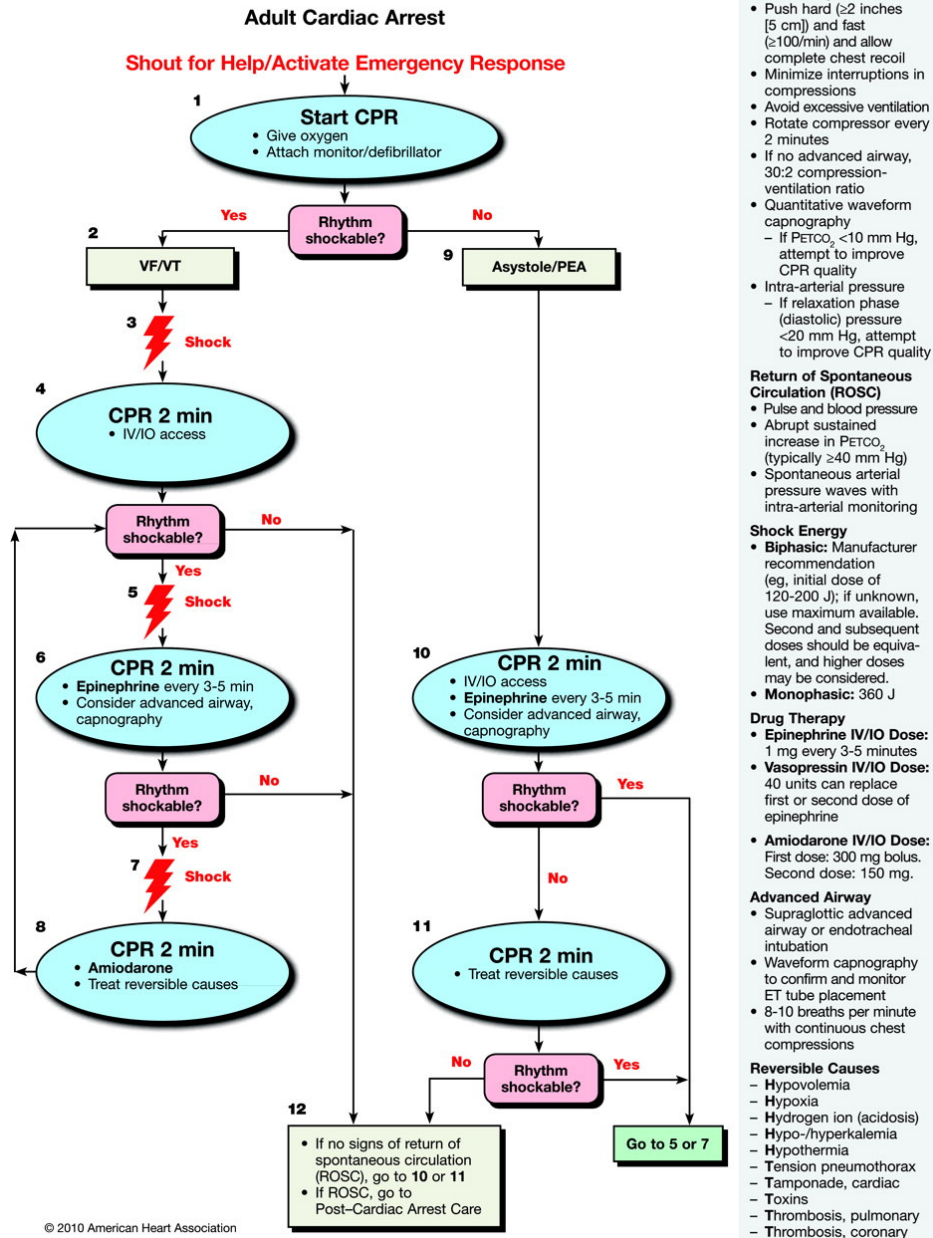


Figure 5.1: This illustration, from [27], demonstrates adult cardiac arrest algorithm used for resuscitation.

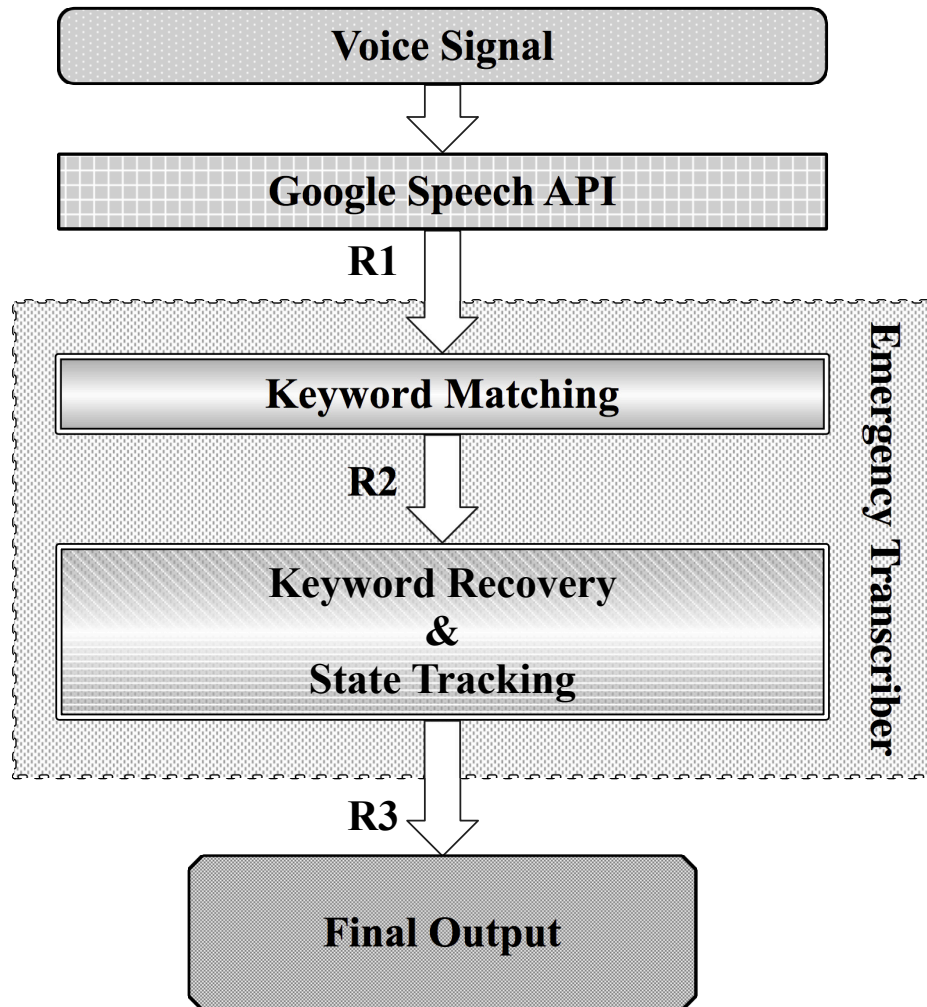


Figure 5.2: This illustration demonstrates the system component consisting of three major modules.

## Web Speech API Demonstration

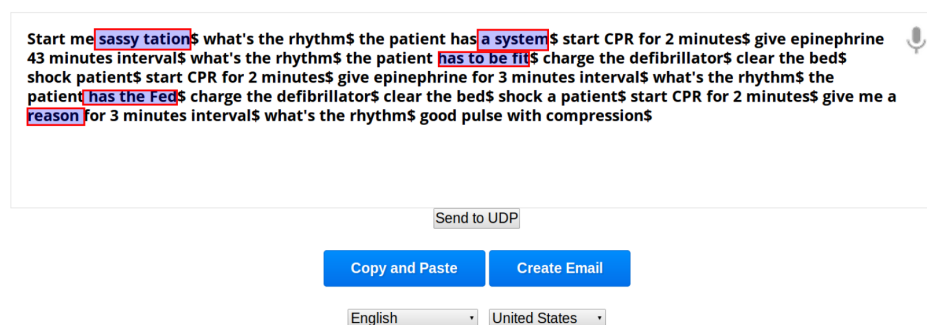


Figure 5.3: This graph shows system result for initial recognition.

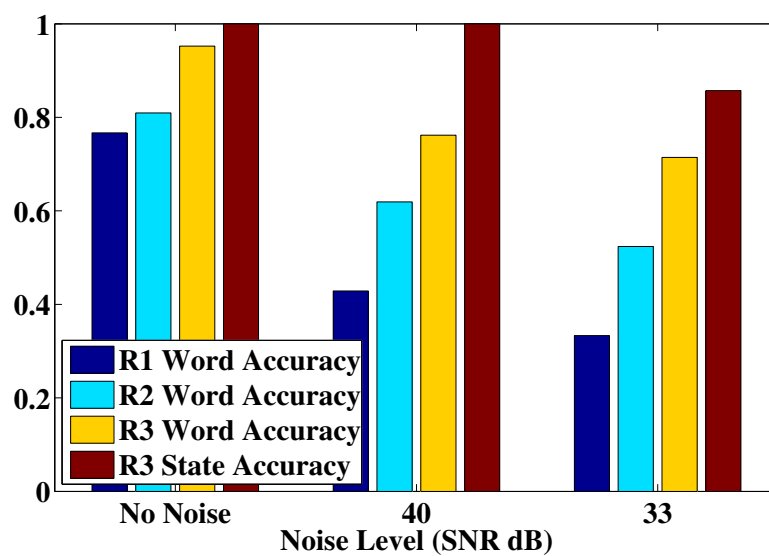


Figure 5.4: This graph shows the result for system comparison result.

# CHAPTER 6

## RELATED WORK

### 6.1 Sensing Classification

Classification techniques in sensing area have been widely studied. For instance, [33] studies data classification problem in wireless sensor networks. It proposed a classification approach in combining local classifier to form a global classifier in order to achieve high accuracy. [34] proposed a hierarchical aggregate classification method to achieve high accuracy in lack of energy and label information, and the author tested their scheme in the wild to classify bird species. In [35], the authors proposed a novel tiered heterogeneous wireless image sensor network for real-time unobtrusive species detection and video cataloging. Their scheme enhanced the detection accuracy via a new hierarchically scalarized-character oriented detection (HIS-CODE) algorithm and architecture.

[36] [37] [38] [39] and [40] have also accomplished some achievements in this area. Our work differs from the existing work in the sense that it takes the workflow information into consideration to increase the sensing accuracy in the face of unreliable sensors and environmental noise. What counts for more is that it also keeps track of the states that have been traversed in the workflow.

### 6.2 Hidden Markov Model

We utilized the Hidden Markov Model (HMM) [19] [41] to model the state transition and the relationship between each sensing object and its associated state. However, our scheme is different from traditional HMM because the observations that the sensing system acquired is not accurate, and merely

based on the inaccurate sensing result to predict the state transition sequence will lead to erroneous prediction results. In order to tackle that problem, we combine the classification matrix of the sensor with the general HMM and form a tiered solution to find the optimal sensing object as well as the hidden states as a whole.

Hidden Markov Model has been applied in different realms in engineering, such as time series analysis [42] [43], part of speech tagging [44], gene prediction [45], and metamorphic virus detection [46]. However, it is best known that the most successful field that Hidden Markov Model has been exploited is the area of speech recognition. Various works have utilized Hidden Markov Model to enable and enhance the performance of automatic speech recognizer, such as [47] [48] [49]. However, traditional HMM models the state transition between different phonemes as Hidden Markov Process. It is different than ours where we treat the state transition behind the sensing procedure as Hidden Markov Process. For the case study specifically, the hidden states refer to the stage where physicians have been working on.

### 6.3 Speech Recognition

We apply our scheme in the area of speech recognition [50] [51] under medical environment. There are several commercialized speech recognition software available for medical practice purposes. For example, Nuance Dragon Medical [52] is a speech recognition package specifically for physician practices. [53] is another industrial grade speech recognition used in healthcare sector. In fact, our scheme is complementary to the above-mentioned automatic speech recognizers (ASRs) because it includes the effect of workflow when doing the speech recognition. And the workflow information is free from the errors of the sensor and the noise of the environment, which makes our scheme perform better. It can act as a light-weight wrapper outside the ASRs for any specific use case, thus our scheme has the advantage of good compatibility and portability.

# CHAPTER 7

## CONCLUSION

In this thesis, we consider the problem of exploiting context information to enhancing the accuracy of sensing under noisy environment and unreliable sensors. We proposed a scheme which can find the optimal combination of sensing measurements and state transition relationship based on the workflow information and the confusion property of the sensors. Simulation results show that our scheme can bring performance improvement compared with standalone sensors. A user study in emergency environment shows that our scheme can increase the speech recognition accuracy and state tracking accuracy.

## REFERENCES

- [1] W. Wang, H. Luo, and H. Deng, “Research on data and workflow security of electronic military systems,” in *Intelligent Control and Information Processing (ICICIP), 2013 Fourth International Conference on*. IEEE, 2013, pp. 705–709.
- [2] InterGraph, “Plant operations,” <http://www.intergraph.com/learnmore/ppm/operations-and-maintenance/plant-operations.aspx>.
- [3] M. Darbari, S. Medhavi, and A. K. Srivastava, “Development of effective urban road traffic management using workflow wechniques for upcoming metro cities like lucknow (india).” *Development*, vol. 2, no. 2, 2008.
- [4] S. Chatty, H. Uninski, and S. Sire, “Supporting collaboration in air traffic control through flight contracts,” 2002.
- [5] C. Sell and I. Braun, “Using a workflow management system to manage emergency plans,” in *Proceedings of the 6th International ISCRAM Conference*, vol. 41. Citeseer, 2009.
- [6] Stanley, “healthcare workflow,” <http://www.stanleyhealthcare.com/solutions/clinical-operations-workflow>.
- [7] M. Kastner, M. W. Saleh, S. Wagner, M. Affenzeller, and W. Jacak, “Heuristic methods for searching and clustering hierarchical workflows,” in *Computer Aided Systems Theory-EUROCAST 2009*. Springer, 2009, pp. 737–744.
- [8] M. A. Tijunelis, E. Fitzsullivan, and S. O. Henderson, “Noise in the ed,” *The American journal of emergency medicine*, vol. 23, no. 3, pp. 332–335, 2005.
- [9] D. Orellana, I. J. Busch-Vishniac, and J. E. West, “Noise in the adult emergency department of johns hopkins hospital,” *The Journal of the Acoustical Society of America*, vol. 121, no. 4, pp. 1996–1999, 2007.
- [10] L. S. Zun and L. Downey, “The effect of noise in the emergency department,” *Academic emergency medicine*, vol. 12, no. 7, pp. 663–666, 2005.

- [11] T. W. Brown, M. L. McCarthy, G. D. Kelen, and F. Levy, "An epidemiologic study of closed emergency department malpractice claims in a national database of physician malpractice insurers," *Academic Emergency Medicine*, vol. 17, no. 5, pp. 553–560, 2010.
- [12] H. Nurse, <http://monster.salary.com/salarywizard/Head-Nurse-Emergency-Room-Job-Description.aspx>.
- [13] M. Anusuya and S. K. Katti, "Speech recognition by machine, a review," *arXiv preprint arXiv:1001.2267*, 2010.
- [14] M. Gales and S. Young, "The application of hidden markov models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.
- [15] S. E. Levinson, "Speech recognition technology: a critique," *Proceedings of the National Academy of Sciences*, vol. 92, no. 22, pp. 9953–9955, 1995.
- [16] J. Picone, "Fundamentals of speech recognition: A short course," *Institute for Signal and Information Processing, Mississippi State University*, 1996.
- [17] P. J. Moreno, "Speech recognition in noisy environments," Ph.D. dissertation, Carnegie Mellon University, 1996.
- [18] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech communication*, vol. 16, no. 3, pp. 261–291, 1995.
- [19] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [20] ACLS, "Cardiac arrest algorithm," <https://www.acls.net/basic-life-support-primary-survey.htm>.
- [21] Y. Deng and W. Byrne, "Hmm word and phrase alignment for statistical machine translation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 3, pp. 494–507, 2008.
- [22] C. Karlof and D. Wagner, *Hidden Markov model cryptanalysis*. Springer, 2003.
- [23] I. Visser, M. E. J. Raijmakers, and H. L. J. van der Maas, *Hidden Markov Models for Individual Time Series*. Springer US, 2009. [Online]. Available: [http://dx.doi.org/10.1007/978-0-387-95922-1\\_13](http://dx.doi.org/10.1007/978-0-387-95922-1_13)



- [24] M. A. Peberdy, C. W. Callaway, R. W. Neumar, R. G. Geocadin, J. L. Zimmerman, M. Donnino, A. Gabrielli, S. M. Silvers, A. L. Zaritsky, R. Merchant et al., “Part 9: Post-cardiac arrest care 2010 american heart association guidelines for cardiopulmonary resuscitation and emergency cardiovascular care,” *Circulation*, vol. 122, no. 18 suppl 3, pp. S768–S786, 2010.
- [25] R. Amathieu, X. Combes, W. Abdi, L. El Housseini, A. Rezzoug, A. Dinca, V. Slavov, S. Bloc, and G. Dhonneur, “An algorithm for difficult airway management, modified for modern optical devices (airtraq laryngoscope; lma ctrach64): a 2-year prospective validation in patients for elective abdominal, gynecologic, and thyroid surgery,” *Anesthesiology*, vol. 114, no. 1, pp. 25–33, 2011.
- [26] A. for Advanced Cardiac Life Support 2014, <https://www.acls.net/aclsalg.html/>.
- [27] uwhealth.org, “Acls algorithm,” [http://www.uwhealth.org/files/uwhealth/docs/pdf4/EEC/acls\\_algorithms.pdf](http://www.uwhealth.org/files/uwhealth/docs/pdf4/EEC/acls_algorithms.pdf).
- [28] G. S. API, <https://www.google.com/intl/en/chrome/demos/speech.htm>.
- [29] “initial speech recognition web interface,” <http://monster.salary.com/salarywizard/Head-Nurse-Emergency-Room-Job-Description.aspx>.
- [30] Linux, “espeak,” <http://espeak.sourceforge.net/>.
- [31] wikiBooks, “Levenshtein distance,” [http://en.wikibooks.org/wiki/Algorithm\\_Implementation/Strings/Levenshtein\\_distance](http://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance).
- [32] “Emergency voice recording,” <https://www.dropbox.com/s/s7nhcpc69yi5fty/path1version2take3.m4a>.
- [33] X. Cheng, J. Xu, J. Pei, and J. Liu, “Hierarchical distributed data classification in wireless sensor networks,” *Computer Communications*, vol. 33, no. 12, pp. 1404–1413, 2010.
- [34] L. Su, J. Gao, Y. Yang, T. F. Abdelzaher, B. Ding, and J. Han, “Hierarchical aggregate classification with limited supervision for data reduction in wireless sensor networks,” in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2011, pp. 40–53.
- [35] D. Duran, D. Peng, H. Sharif, B. Chen, and D. Armstrong, “Hierarchical character oriented wildlife species recognition through heterogeneous wireless sensor networks,” in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*. IEEE, 2007, pp. 1–5.

- [36] F. Huang, Z. Jiang, S. Zhang, and S. Gao, "Reliability evaluation of wireless sensor networks using logistic regression," in *Communications and Mobile Computing (CMC), 2010 International Conference on*, vol. 3. IEEE, 2010, pp. 334–338.
- [37] X. Yang, A. Dinh, and L. Chen, "Implementation of a wearable real-time system for physical activity recognition based on naive bayes classifier," in *Bioinformatics and Biomedical Technology (ICBBT), 2010 International Conference on*. IEEE, 2010, pp. 101–105.
- [38] L. Su, Y. Gao, Y. Yang, and G. Cao, "Towards optimal rate allocation for data aggregation in wireless sensor networks," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2011, p. 19.
- [39] Z. Zeng, S. Yu, W. Shin, and J. C. Hou, "Pas: A wireless-enabled, cell-phone-incorporated personal assistant system for independent and assisted living," in *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on*. IEEE, 2008, pp. 233–242.
- [40] L. Su, S. Hu, S. Li, F. Liang, J. Gao, T. F. Abdelzaher, and J. Han, "Quality of information based data selection and transmission in wireless sensor networks," in *Real-Time Systems Symposium (RTSS), 2012 IEEE 33rd*. IEEE, 2012, pp. 327–338.
- [41] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden markov model: Analysis and applications," *Machine learning*, vol. 32, no. 1, pp. 41–62, 1998.
- [42] S. Shi and A. S. Weigend, "Taking time seriously: Hidden markov experts applied to financial engineering," in *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*. IEEE, 1997, pp. 244–252.
- [43] D. Roman, G. Mitra, and N. Spagnolo, "Hidden markov models for financial optimization problems," *IMA Journal of Management Mathematics*, vol. 21, no. 2, pp. 111–129, 2010.
- [44] B. Highfill, "Part of speech tagging with discriminatively re-ranked hidden markov models."
- [45] M. Stanke and S. Waack, "Gene prediction with a hidden markov model and a new intron submodel," *Bioinformatics*, vol. 19, no. suppl 2, pp. ii215–ii225, 2003.
- [46] S. Attaluri, S. McGhee, and M. Stamp, "Profile hidden markov models and metamorphic virus detection," *Journal in computer virology*, vol. 5, no. 2, pp. 151–169, 2009.

- [47] M. Gales and S. Young, “The application of hidden markov models in speech recognition,” *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.
- [48] S. Young, “Hmms and related speech recognition technologies,” in *Springer Handbook of Speech Processing*. Springer, 2008, pp. 539–558.
- [49] B. H. Juang and L. R. Rabiner, “Hidden markov models for speech recognition,” *Technometrics*, vol. 33, no. 3, pp. 251–272, 1991.
- [50] R. P. Lippmann, “Speech recognition by machines and humans,” *Speech communication*, vol. 22, no. 1, pp. 1–15, 1997.
- [51] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, “Robust automatic speech recognition with missing and unreliable acoustic data,” *Speech communication*, vol. 34, no. 3, pp. 267–285, 2001.
- [52] D. Medical, <http://www.nuance.com/for-healthcare/by-solutions/speech-recognition/dragon-medical/index.htm>.
- [53] SpeechMagic, <http://www.nuance.com/products/speechmagic/index.htm>.