A Socially Aware Guide Robot with Illumination Invariant Face Recognition and Self Learning of Guiding Behavior

(照明不変の顔認識と案内行動の自己学習に基づく社会的認知機能をもつ案内ロボット**)**

August, 2016

Doctor of Engineering

Bima Sena Bayu Dewantara

ビマ セナ バユ デワンタラ

Toyohashi University of Technology

DATE : 2016/07/11

Dept. of Computer Science and Engineering		ID	D129303
Name	BIMA SENA BAYU DEWANTARA		

Supervisor	Prof. JUN MIURA

Abstract

Title	A Socially Aware Guide Robot with Illumination Invariant Face Recognition and Self Learning of Guiding Behavior

In recent decades, robots have entered into daily human life. The robot has penetrated wider areas of human life, with the main purpose of assisting and facilitating human works, ranged from home appliances until industrial areas. Today, one of the most rapidly growing areas is service robots. The functionality of the robot is expanded for many purposes which emulate human behaviors and activities. Along with the development of social needs, a robot is expected to be used as a companion or assisting or serving peoples' activities in a wider range. However, the implications of the presence of a robot as a new entity in the human social environment are still continuously studied.

To be able to further interact with humans, a robot that runs a particular function with social awareness capabilities and is able to behave by obeying social norms will be highly accepted rather than a robot which has a goal-oriented only. In regard to emulate what human can do with a robot, we focus our work on a socially aware guide robot behavior by arguing that this kind of robot has a unique task and can be used to show the usefulness of social awareness and how the robot should behave when interacting with a human.

The functions of guide robots have been studied in many previous works, ranged from museum, mall, office, airport, and helping a disabled person. In their study, the guide robots were equipped with many features which may useful for interacting with humans. However, the responsibility on the human side is still greater than the robot. There is a kind of obligation for the partner to always be aware of the robot states. In some cases, the partner is required to understand the rules or commands if he/she wants to interact with the robot, i.e., using hand gestures to call or stop the robot, voices, and other tools. This approach may be acceptable, but we think that it needs participants who are set in advance.

In this thesis, we design our socially aware guide robot from two basic tasks, namely coordination and navigation. The coordination task is a task or process of monitoring human partner awareness and acting a necessary response to ensure the human partner completing an ongoing guiding task, while the navigation task is a task or process of monitoring and controlling the movements of the robot from starting position to the goal position. By actively

monitoring and appropriately responding human partner awareness, the robot is expected to be more responsible than the partner. Exchanging this responsibility is very important because there are many things that can affect the focus and attention of the human partner. We want our robot more responsive to the possibilities that could happen so that preventive actions can be done.

To do its job, our robot is equipped with an ability to always recognize and track the human partner in any conditions. We utilize a variety of information obtained from the human partner, such as the face and the color of clothes. We complete our guide robot with an illumination invariant face recognition system that is robust for identifying a person even under harsh illumination conditions. With this addition, the visibility of the robot can be extended for backlighting, normal, strong lighting, and less lighting conditions. The color of clothes is also very helpful when the face cannot be detected. Once the human partner can be recognized and tracked, the robot measures a degree of the human partner awareness by combining information from the human partner's relative distance to the robot, walking orientation and head orientation. The combination of those three factors will generate a coordination control signal between the human partner and the robot.

We consider the robot's behavior that can meet all situations to deal with smoothness, safety, and comfort. We use a Q-Learning based Social Force Guiding Model (QL-SFGM), our motion planning and control method that is specifically designed and optimized for guiding purpose using a differential drive mechanism mobile robot under social force framework. To ascertain the robot movements to always convergence with its goal direction, we apply a Proportional-Integral-Derivative (PID) controller. This PID controller is used to reduce bouncing effects that sometimes exhibited by our QL-SFGM and to speed up focusing our robot heading to the goal direction. We insert the PID controller in the QL-SFGM scheme and optimize the PID coefficients also using Q-learning.

We validate our proposed method by performing experiments using a realistic 3D simulator and a real robot implementation. The experimental results show that our system is very promising and robust enough to perform guiding tasks in any situations. This work can be a stepping stone towards a socially aware guide robot that is fully responsible to the user.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Backgrounds

Nowadays, the utilization of robots is rapidly growing in many applications. We can find and interact with robot in our daily life since it is widely utilized ranged from home appliances until industrial robots [1]. At the beginning, the development of robots was inspired by the idea of helping the human in a heavy or repetitive work by artificial means. Nevertheless, today robots are expanded for many purposes which emulate human behaviors and activities. When a robot is demanded to replace or to support the human in a certain task, it implies that the robot needs to emulate, fully or partially, what the human do. This emulation can be in the form of actions, procedures, or even the human thinking perspective to finish the job.

Along with the development of social needs, a mobile robot is expected to be used as a companion or assisting or serving peoples' activities in a wider range. However, the implications of the presence of robots as a new entity in the human social environment are still continuously studied. In a mobile robot application with goal-oriented, completing the objective is the main goal. The robot system usually assumes a human as a moving obstacle with a forbidden area to be intruded with a center point and a predetermined radius. By this assumption, the robot must avoid that area when it navigates "near" the detected human. The term of "near" is relatives. It depends on how the robot system determines the safety in robot navigation. However, what has been determined sometimes does not ensure comfort for humans.

Involving a robot to work in a shared space with humans is necessary to always obey human social interaction norms. The norms usually implicitly set up how human relationships. These norms have classified human relations, for example, family, friends, and strangers. Each classification requires a different way of interacting. With a family, we might be able to interact as closely as possible. Normatively, interactions with friends may not be as close to the family. So it is with strangers. Respecting such kind of norms must be carried out during the interaction

By looking at the development of robotic applications that are increasingly exploring the social relationship with humans, we believe that the development of social robots will increase rapidly in the future. Therefore, a robot that runs a particular function with a social interaction and social awareness capabilities and is able to behave by obeying social norms will be highly accepted rather than a conventional robot which has a goal-oriented only. In regard to emulate what human do with a robot, we focus our work on a social guide robot by arguing that this kind of robot has a unique task and can be used to show the usefulness of a social awareness in a social interaction.

To further emphasize our focus, we will briefly give an example to demonstrate the usefulness of the social guide robot. Let us imagine a guest who comes to an unknown place in advance wants to go to a specific goal, e.g.: director room in an office, the room of the head of school or university, toys shop in a shopping mall, etc. Usually, the place authority will provide two solutions, i.e., informative sketch rooms or information desks. Let us consider the situation that the guest comes to the information desk to ask the location of his goal. There are two possibilities which may be obtained; First, the receptionist remains to stay in his/her position and will guide him using a map by explaining the location of the target place. It means that the guest must go to his goal by himself based on the direction given in advanced. Second, the receptionist will call another staff to accompany the guest to the goal. This option will certainly increase the guest satisfaction to the place, but it needs more human efforts. Therefore, introducing a robot that can tackle this task will be useful.

Now our mission is to replace the human guide with the robotic platform. Fully substituting a human guide by a robot is not an easy task. However, only emulating a behavior of the human guide into a robot is making sense. Nevertheless, several issues need to be considered:

1. The robot must have a social awareness.
2. The robot should be able to navigate safely by minimizing unnecessary movements to deal with the human partner comfort.
3. A robust identification of the targeted human partner is required.
4. Ensuring the targeted human partner to follow the robot is required.

Considering the above issues, we attempt to build a partner-oriented based social guide robot services and sort the service priority as follows: (1) partner comfort based join task coordination, and (2) safely navigation towards the destination. Since the partner comfort is closely related to the absence of stress or annoying feelings when interacting with the robot, then how the robot should behave is crucial. Safe and smooth navigation that is closely related to how the robot controls its motion and responds the obstacles also frequently affects the human comfort. Therefore, we propose a novel motion control and behavior learning strategy for a guide robot to imitate the job of the human guide that represents minimal robot functions as follows:

Definition 1. *(Socially Aware Guide robot). A guide robot takes a human partner to the destination, while behaving as socially acceptable as possible.*

The term "behaving as socially acceptable as possible" is raised, basically for tackling the problems specified by point (1) and (2). The robot behavior can be associated with the execution result of the motion control of the robot. The motion control responds the environment situation

perceived by sensors and output an action that should be done by the robot. Executing a specific action will produce a specific behavior, for example, when the robot orthogonally faces a moving obstacle at a very close distance, let say 1 meter, then it will react to avoid the obstacle to the right or to the left or turning back, where each reaction will produce different behavior. We expect our robot can smoothly avoid the obstacle, while still prioritize the safety. This smooth and safe behavior is important because it can be socially acceptable and can also keep the human partner comfort at once.

Another useful factor to support our guide robot is an ability to relocate and reidentify the human partner in a variety conditions to address an issue pointed by point (3). A stable information of the human partner is important to guarantee the robot guides the correct person. Guiding a wrong person in the middle of the guiding duration is prohibited. The robot is obliged to complete its duty to guide this specific person to reach his/her goal. We equip our robot with a combination of illumination invariant face recognition and a histogram of clothing color. These features are then combined with a position tracking to achieve a better result.

The last useful factor to realize the partner-oriented based guide robot is an ability to monitor the human partner intention to follow the direction given by the robot. The robot should monitor all possible aspects which can be obtained from the human partner in order to identify and to estimate the possibility of the human partner will be lost or left behind. In other words, we expect our robot can measure how committed the human partner to complete the task to address an issue pointed by point (4). The human partner is a subject to the guiding task by showing a specific behavior which symbolizes his/her obedience for the process that is being followed. The behaviors in guiding context usually involve the intentional awareness and attentional awareness. An example of the intentional awareness based behavior which subjects to the guiding process is walking in the same direction with the guide, while an example of the attentional awareness based behavior is frequent to look at the guide to ensure he/she does not leave behind. By doing so, we expect our robot can maintain a visible distance from the human partner by always be close to the human partner to prevent unexpected events may occur. Although a human guide usually goes together or side-by-side with his/her partner, but at a certain condition sometimes the human guide walking in front, i.e., to guide a group of peoples or walk on a narrow road. In this work, we argue a robot which has no communication ability is better to run in front of the human partner than side-by-side.

1.2 Contributions

We build a minimal guide robot system that is composed of our own three contributions where each contribution contributes equally as follows.

1. To strengthen the personal identity recognition, we have proposed a novel illumination invariant face recognition system using an Optimized Fuzzy Inference System (OptiFuzz). A

Fuzzy Inference System (FIS) is used to analyze the raw input image and adaptively adjust the image contrast. To make the system can be used in more general face appearance cases, the Fuzzy's rule is optimized using Genetic Algorithm.

2. To measure the person's attention robustly, we have proposed a robust optimized descriptor by combining several cues. This descriptor is then analyzed to classify a human heading direction.

3. To realize our partner-oriented based social guide robot, we have proposed a novel approach for handling a socially aware guide robot motion planning and control, and behavior learning by actively observing the human partner awareness. A Markov decision-based self-learning strategy of guiding behavior under Social Force framework is introduced.

Each contribution will be explained further in the following chapters related to each problem.

1.3 Outline of Dissertation

The outline of this dissertation is as follows.

- **Chapter 1** describes our motivation for doing research in the field of guide robot. This chapter also briefly explains our contributions.
- **Chapter 2** will present the state of the art and identify research challenges.
- **Chapter 3** will present the general framework of our research for the robot's behavior and motion planning.
- **Chapter 4** will present our illumination invariant face recognition method to control the face's contrast which is detected from video sequences to get invariant face appearances even though the system is used indoor and outdoor.
- **Chapter 5** will present a new method to classify human head facing direction.
- **Chapter 6** will present a guide robot system that integrates all ideas and its experimental validation.
- **Chapter 7** will conclude the dissertation with a summary of the concepts and frameworks introduced, followed by the potential future work and applications.

Chapter 2

State of the Art and Research Challenges

2.1 Historical Background of a Guide Robot

In the last two decades, guide robot has been a raised research topic in the service robot's field. History of the guide robot started with the development of Rhino [2] in 1999, as a tour-guide robot in a museum (Fig. 2.1(a) [1]). The robot was equipped with a modular and distributed software architecture, which integrates localization, mapping, collision avoidance, planning, and various modules concerned with user interaction and web-based telepresence. This robot was capable of demonstrating a reliable operation in public environments.

The use of museum as a suitable public space is then followed by Robox [3], Minerva [4] (Fig. 2.1(b) [2]), Rackham [5] (Fig. 2.1(c) [3]), Mobot [6] and Cice [7]. All of those robots had been embedded with navigation modules that efficiently prevented collisions with humans or other obstacles using robust navigation strategies such as a minimal required obstacle distance or a stop-and-wait behavior [8]. Some of them also equipped with abilities to make simple interaction with the user using gestures [5].

Recently, another application is an office guide robot [9] and a shopping mall guide robot that is proposed by Kanda *et.al*. [10]. The office guide robot uses a PeopleBot platform and they focused the study on the person passing and collision avoidance strategies. While the shopping mall guide robot uses a Robovie platform and they focused the study on an interactive guidance by displaying shopping information and answering simple questions from the users. URBANO [11] was used as a tour-guide with emphasized on presentation skills.

More recently, the development of mobile guide robots tends to enrich the functions such as engaging and disengaging the audience [12] and how to maintaining effective states of the users by offering outdoor location-based services using augmented reality [13]. Besides that, the use of the guide robots is not only to help normal people but also to help people who have disabilities. Feng *et.al*. [14] built a guide robot system to navigate a blind person to an indoor destination in an effective and socially-acceptable way.

[1] http://iai.uni-bonn.de
[2] http://www.ri.cmu.edu
[3] http://www.laas.fr

Figure 2.1: Examples of the museum guide robots; (a) Rhino, (b) Minerva, and (c) Rackham.

2.2 The State of the Art Technical Approaches

2.2.1 Robot Navigation in Social Environment

Navigation is a basic skill for autonomous mobile robots [15]. This skill needs a motion planning which controls the robot to move from its initial position towards a goal. However, navigating among human environment needs to understand social norms between each other that must guarantee safety [16]. Unnatural movements of the robot would cause conflicts between each other because such kind of movements will make the other humans feel disturbed and uncomfortable. The traditional approach in robotics has usually treated a human as a moving obstacle with a forbidden area to be intruded with a center point and a predetermined radius and utilized algorithms that assure collision-free motion in the presence of such moving obstacles. However, this approach frequently experienced some difficulties when applied to real world environments.

An overview of autonomous robots that have been utilized in human environments was given by Jensen in 2005 [8]. Among the presented system's building blocks, a navigation module that prevents collisions with humans or other objects using robust navigation strategies such as a minimal required obstacle distance or a stop-and-wait behavior in situations of conflict were commonly used. In regards to the human-aware capability needed by the robot while navigating, Kruse *et al*. [15] defines the challenges of human-aware navigation that are defined as follows.

- **Comfort** is the absence of annoyance and stress for humans in interaction with robots.
- **Naturalness** is the similarity between robots and humans in low-level behavior patterns.
- **Sociability** is the adherence to explicit high-level cultural conventions.

Comfort is different from safety; it is a psychological condition that a human feels safe. This must be supported by the absence of a sense of being intimidated or threatened by something. Related to this work, even when a robot is moving in its correct path, a human may still feel that the motion is not safe, lacking trust in the technology. It needs to identify robot behaviors that are clearly uncomfortable to humans and to modify the behavior.

Naturalness means that a robot strives to emulate nature, such as human motion, as the target behavior to be recreated by the robot. By this definition, a robot's behavior can be studied at two levels; at low-level, parameters of behavior, like dynamics, shapes and velocities can be approached using a continues distance measurement between behaviors. However, due to the different abilities of the robot compared to humans, not all aspects of human can be emulated. At a high-level, the way a robot makes a decision is accommodated. This level is needed to provide a more socially decision that should be taken by a robot from a social situation.

2.2.1.1 Proxemic Interpersonal Distances

Recent studies of robot navigation among human environments refer to socially-acceptable behavior during navigation [15, 16, 17, 18] by planning the robot's path to maintain a "social distance" from humans and respect their personal spaces. By respecting the human's personal space, a robot can avoid them from a considerable distance and hopefully they work as usual while a robot navigating in a shared space and time among them. A segmentation of space around a human or a robot is the Proxemic Interpersonal distances found by Hall [19] which is shown in Table. 2.1. Such interpersonal distances were then further examined by Zanlungo [20] by presenting several model specifications, i.e., circular and elliptical, shown in Fig. 2.2.

2.2.1.2 Motion Planning and Control for Robot Navigation

There are so many methods proposed for planning motion and control of a mobile robot to navigate among social environments. Hart *et. al.* [21] developed the first version of A* algorithm, a computer algorithm that is more efficient than Dijkstra algorithm and is widely used in finding the shortest path in graphs. A* solves the problems by searching among all possible paths towards the goal by considering the smallest cost (short distance, short time, etc.), and among these paths

Table 2.1: Proxemic interpersonal distances as found by Hall [19]. How those categories and the values are to be adopted for the human-robot context is an open research question.

Designation	Specification	Reserved for ...
Intimate distance	0 - 45 cm	Embracing, touching, whispering
Personal distance	45 - 120 cm	Friends
Social distance	120 - 360 cm	Acquintances and strangers
Public distance	> 360 cm	Public speaking

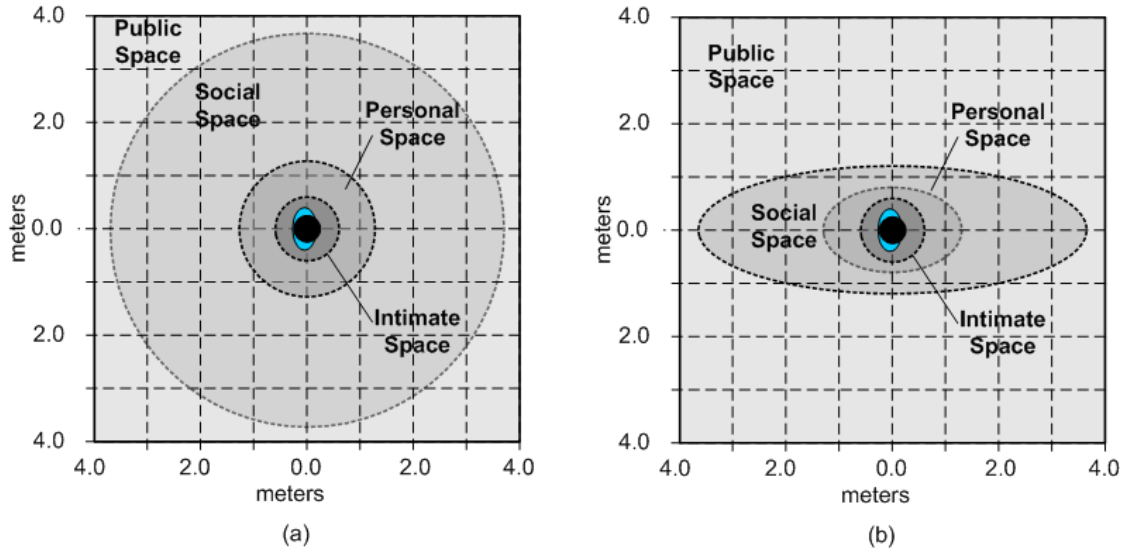


Figure 2.2: Proxemic interpersonal distance of a human; (a) circular model and (b) elliptical model.

it first considers the ones that appear to lead most quickly to the solution. The A* is then usually combined with other control methods to move the robot by following the resulted path.

Another motion planning and control algorithm that are widely used in robotics is Potential Field [22]. Principally, Potential Field method uses a two-dimensional Cartesian grid for obstacle representation. Each cell (i, j) in the grid holds a certainty value (CV) $c_{i,j}$ that represents the confidence of the algorithm in the existence of an obstacle at that location. Potential Field method adopts the concept of virtual forces: repulsive force and attractive force. A repulsive force that is proportional to the CV occurs when the robot closes to an obstacle. An attractive force pulls the robot to the goal. However, this algorithm has some significant problems: (1) trap situations due to local minima, (2) no passage between closely spaced obstacles, (3) oscillations in the presence of obstacles, and (4) oscillations in narrow passages.

Recently, the most phenomenal path planning method is Rapidly Exploring Random Tree (RRT) that was introduced by La Valle *et. al*. [23]. RRT is an algorithm designed to efficiently search nonconvex, high-dimensional spaces by randomly building a space-filling tree. The tree is constructed incrementally from samples drawn randomly from the search space and is inherently biased to grow towards large unsearched areas of the problem. This algorithm can handle problems with obstacles and differential constraints (nonholonomic and kinodynamic) and have been widely used in autonomous robotic path planning.

With the increasing number of applications of service robot to help people, the robot is more expected to understand human social aspects. It means that the behavior of the robot also becomes a crucial factor instead of its path planning and motion control. More recently, Social Force Model (SFM) by Helbing in [24] is widely accepted and implemented in robotic applications. SFM is a behavioral modeling of a pedestrian when interacting with the others. It responses an internal

and external stimulus based on physical and socio-psychological aspects. Helbing described that pedestrian movements are influenced by the perception of the situation and environment, and personal aims and interest factors. These factors are then described using forces, where the forces affected by the environment are usually categorized as repulsive, and the force affected by the personal interests is categorized as attractive. The works of [16, 17, 18, 20, 25, 26] are the examples of proposed methods which adopted SFM under specific interpersonal model specification.

2.2.2 Awareness in Human-Robot Interaction

The interaction between human and robot is not limited to communication only, but also includes how both human and robot can maintain their own situation to join in a mutual interaction task. Such an interaction usually requires awareness of both. In guide robot system, the activities of a human partner are necessary observed simultaneously by the robot to exhibit an appropriate guiding behavior that meets comfort and safety. There are many definitions of the awareness which is summarized in [27] regarding the interaction between human and robot. However, due to guiding process, we pointed out three categories of awareness that should be fulfilled by the robot and the human partner, namely situational-awareness, social-awareness, and task-awareness.

Situational awareness can be explained as "the perception of elements in an environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future". While social-awareness can be explained as "the understanding that participants have enough information about the presence and activities of people in a shared environment". Both kinds of awareness imply that the robot must consider both environment structures and moving objects in the same shared workspace and time while executing its task. The third kind of awareness, task-awareness can be explained as "the participant understanding of how their tasks will be completed". It implies that a human participant should have the intention to finish his work without canceling the process unilaterally.

When a human partner and a guide robot work together in the same task, a commitment between them is required. The guide robot should aware to the objects (i.e., static and dynamic) around it related to safety navigation and commit to finishing its job by performing a good service to the human partner. On the other hand, the human partner should manage his/her commitment to do and finish this joint task. In the history of guide robot above, all of the robots are designed to a function-oriented work. They consider the safety navigation with a little interaction between the robot and guided person(s). However, the level of human responsibility is still higher than the robot. Switching the level of responsibility from human to the robot is an inevitability.

By switching the level of responsibility, a more gentle guiding behavior is achievable and human's role can be minimized. By designing a guide robot that always understands what the human partner is doing based on his/her behavior, an appropriate selection of the robot behaviors can be obtained. This better behavior selection may answer at least two challenges in a human-aware robot

navigation as follows.

- Comfort - the human partner feels comfortable since the robot is always near him/her. The other humans feel comfortable with the robot's navigation during guiding process. No intimidation or threats from the robot.
- Sociability - Robot can decide an appropriate behavior by respecting human social norms while interacting each other.

Here we highlight two basic functions which respect to the target-awareness based interaction, namely face recognition and head orientation estimation. These two functions are briefly described as follows.

2.2.2.1 Face Recognition

Vision-based human face recognition is widely used in human-robot interactions, e.g. [28, 29]. It has been shown to be effective under normal illumination conditions. When it is used under severe illumination conditions, however, the recognition rate drops rapidly. We cannot always expect to have a good illumination. When we move through various environments, include both indoor and outdoor, for example, the appearance of facial features largely change, thereby making it difficult to recognize faces.

Several illumination invariant methods have been proposed to solve these problems. Most illumination invariant methods (e.g., [30, 31, 32, 33, 34, 35, 36, 37, 38, 39]) produced invariant face appearances. However, some problems still remain, e.g. the face look flat, unclear, still experiencing the boundary effect, different final color constancy, etc. In this part, we donate one contribution in [40] by proposing a novel illumination invariant face recognition method using Fuzzy Inference System (FIS) to adaptively adjust the pixel's contrast. To produce better contrast adjustment, we trained the Fuzzy's rules using a Genetic Algorithm (GA).

2.2.2.2 Head Orientation Estimation

The human head and face are the most common parts of the human body used in computer vision applications such as detecting the presence of a person, identifying and verifying a person, and indicating one's attention. Estimating head orientation is a crucial factor to maintain a good communication or a good interaction with the others. Head orientation is usually used to represent a human attention when interacting with others.

Since the human attention usually involves a wide range of the head orientation angle (i.e., $\pm 90^o$) then appearance-based methods (e.g., [41, 42, 43]) are strongly considered. However, these methods usually used limited feature variations such as intensity and edge information only. We, therefore, proposed a novel descriptor [44] that combines more features which are optimized using

Genetic Algorithm. This optimized combination was proven to be more effective to characterize the differences of each head orientation than just using a single feature.

2.2.3 Robot Motion Control and Behavior Adaptation Learning Strategy

Navigating a mobile robot from start to goal means planning a motion control and executing it. An ability to plan a motion control is one of the main ability which must be owned by a mobile robot. A good planning of motion control cannot be separated from the robot's ability to perceive and analyze the situation. To perceive and sense the situation around, the robot is usually equipped with sensors. These sensors will map the encountered obstacles and transform the mapping result in numeric measurement data (usually as the function of distance) to be sent to the planning algorithm. As the result of the planning, a selected motion control is expected to be the best choice to respond the stimulation received by the robot. Executing the selected motion control is the next step after the planning. Execution of a motion control usually gives a direct impact to the robot pose. A sequence of the robot's poses while navigating forms the robot behaviors.

Executing the motion control without analyzing the effect of the resulting behavior is risky. To deal with producing a smooth and safe behavior, evaluation of the behavior itself is absolutely necessary. Evaluation of this behavior will give correction on motion control that has been generated. The evaluation results are used to adjust the parameters in the planner. Most of the previous work uses a machine learning to optimize those parameters. Offline learning and online learning strategies are commonly used to optimize them. Fig. 2.3 shows a motion control and behavior selection problem experienced by a guide robot. The robot has to choose the best

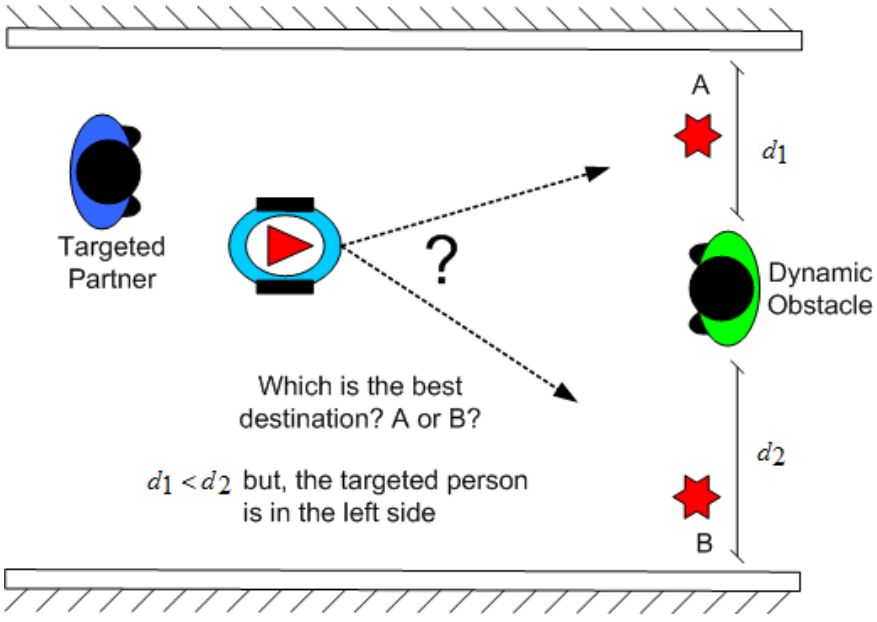


Figure 2.3: Behavior selection.

destination based on its own perception of the situation. When choosing the next destination, an appropriate behavior in a distinct social situation is crucial. The behavior selection must satisfy the definition of smoothness and safety.

2.2.3.1 Offline Learning

Under Social Force framework, all forces are formulated using a function of measurement data with several parameters. The measurement data are based on the real measurements of sensors while the parameters can be constant or variable. Some previous works [18, 20] dealt with the tuning of these parameters by utilizing evolutionary-based optimization approach, i.e., Genetic Algorithm (GA) [45]. Data samples of the path usually formed by the human actors are collected from real experiments. These data samples are then used for optimization using GA, and the optimized parameters apply to the robot. This approach may be practical; however, the optimized parameters are not always in accordance with various conditions, i.e., a robot which is trained for indoor may exhibit a strange behavior when operated in an outdoor environment and vice versa. As the consequence, the robot makes an unexpected behavior that is probably dangerous, threatening to the others, and potentially damaging itself.

2.2.3.2 Online Learning

Changing the parameters adaptively to fit with a particular situation is worth to be considered. Online learning schemes usually adopt Reinforcement Learning (RL) to find the optimal policy by directly interacting with the environments. Briefly, the Reinforcement Learning is the learning process where an agent automatically learns by itself what to do and how to map situation to actions so as to maximize a numerical reward value. The learner must discover which actions yield the most reward by trying them. These actions may affect not only the immediate reward, but also the next situation. Fig. 2.4 shows the basic process in Reinforcement Learning.

Since this approach can be defined as a learning problem, the most important aspects of

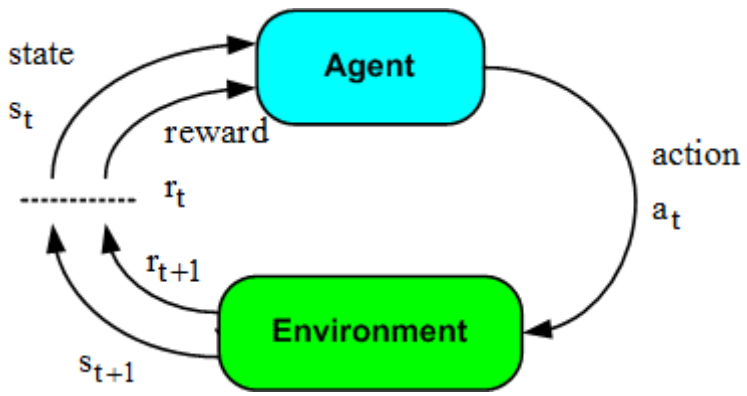


Figure 2.4: The agent-environment interaction in reinforcement learning.

the real problem faced by a learning agent during its interaction with its environment to achieve a certain goal can be captured. Interacting with the environment means that the agent senses the state of its environment and takes actions that affect the state. A unique issue existing in RL is balancing exploration and exploitation of the knowledge. Exploration is important to gain more information and knowledge. It is crucial to explore potential actions instead of using the obtained knowledge in the past in order to maximize a reward. However, conducting the exploration more than exploitation is not good due to policy convergence. The trade-off between exploration and exploitation has been one of the challenges of Reinforcement Learning.

From the book of Sutton *et.al*. [46], Reinforcement Learning mainly has four elements: a policy, a reward function, a value function, and optionally, a model of the environment. A policy defines the way the learning agent behaves at a given time. It could be defined as the mapping from perceived state of the environment to actions to be taken when in those states. A reward function defines the goal of a reinforcement learning problem. It maps each perceived state-action pair of the environment with a single number, a reward, indicating the inherent desirability of that state. A value function specifies what is good in the long run. The value of the state is the total amount of reward an agent can expect to accumulate over the future starting from that state. A model of the environment is something that allows us to mimic the behavior of the environment at any given time. Models are used for planning, hence their use is optional depending on the type of application the RL system is going to be used for.

In general, the RL problem can be formulated as a discrete time, finite state, finite action MDP [46]. The learning environment can be modeled by a 4-tuple $\{s_t, a_t, p_t, r_t\}$, where:

- $s_t \in \mathcal{S}$; $\mathcal{S}$ is a finite set of states.
- $a_t \in \mathcal{A}$; $\mathcal{A}$ is a set of actions that the agent can perform.
- $p_t \in \mathcal{P}$; $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Pi(\mathcal{S})$ is a state transition function, where $\Pi(\mathcal{S})$ is a probability distribution over $\mathcal{S}$. $p(s,a,s')$ represents the probability of moving from state s to s' by performing action a.
- $r_t \in \mathcal{R}$; $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to R$ is a scalar reward function.

The goal of the agent in a RL problem is to learn an optimal policy $\Pi^* : \mathcal{S} \to \mathcal{A}$. Fig. 2.4 diagrams the agent-environment interaction.

Q-learning (QL) is one of the most successful branches of reinforcement learning. In Q-learning, on each interaction step an agent senses the current state s of the environment, chooses an action a, executes this action, switches to the state s' of the environment, and receives a scalar reinforcement signal r (a reward or penalty). Q-learning learns the value of each action in each state. This Q-value is referred to as the state-action value and is updated by

$$Q(s,a) = Q(s,a) + \eta(r(s,a) + \gamma maxQ(s',a) - Q(s,a)), \tag{2.1}$$

where $Q(s,a)$ is a Q value of the state, s, and action, a, pair. η is a learning rate within range (0,1). $r(s,a)$ is a direct reward value for the state-action pair. γ is a discount factor. $maxQ(s',a)$ is the estimated maximum Q-value of the next state.

The goal of learning problem using reinforcement learning is to maximize the reward a learning agent receives in the long run. Formally, we seek to maximize the expected return, where R is defined as some specific function of the immediate reward sequence $r_{t+1}, r_{t+2}, r_{t+3}, ...$ It implies that the total reward value is the sum of the immediate rewards:

$$R = r_{t+1} + r_{t+2} + r_{t+3} + ... + r_T, \tag{2.2}$$

where T is a final time step. When the agent-environment interaction breaks naturally into subsequences, these are called *episodes*. Each episode ends in a special state called the *terminal state*, followed by a reset to a starting state. Tasks with episodes of this kind are called *episodic task*. In some cases, the agent-environment interaction does not break naturally into identifiable episodes. It may be prematurely stopped at $t+N < T$, or continuously without limit (called as *continuing task*). The total reward formulation in equation 2.2 is problematic for this continuing task because the final time step would be $T = \infty$, and the total reward may be infinite.

To address this, the concept of *discounted reward* is introduced. The agent now tries to select actions so that the sum of the discounted rewards it receives over the future is maximized. The discount reward is defined as:

$$R = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{2.3}$$

where γ is a parameter, $0 \leq \gamma \leq 1$, called as the discount rate.

Chapter 3

A Socially Aware Guide Robot Framework

3.1 Problem Definition and Research Objective

In this thesis, we focused our attention on two main issues: (1) how the behavior of a guide robot can be socially acceptable in a social environment and (2) how to establish guiding skills that grow independently by involving minimal human intervention. We raised this issue because to the best our knowledge, not many researchers are discussing how a robot should behave in a social environment, especially in the case of a guide robot. Another interesting thing is establishing independent guiding skills of a guide robot is relatively new. We believe that by letting the guide robot learns by itself will be easier for us to update its skill in the future.

A guide robot system is at least composed of two basic tasks: (1) maintaining a **coordination** with the human partner and (2) an autonomous **navigation** among social environments, as shown in Fig. 3.1. We assumed that the human partner states and a situation of the environment will greatly affect the guide robot behavior. Our main objective is how making a guide robot can run safely and smoothly while still considering the human partner comfort. We listed at least four open research questions that may arise in the case of our guide robot as follows.

- How to maintain a relationship with the human partner to deal with comfort?

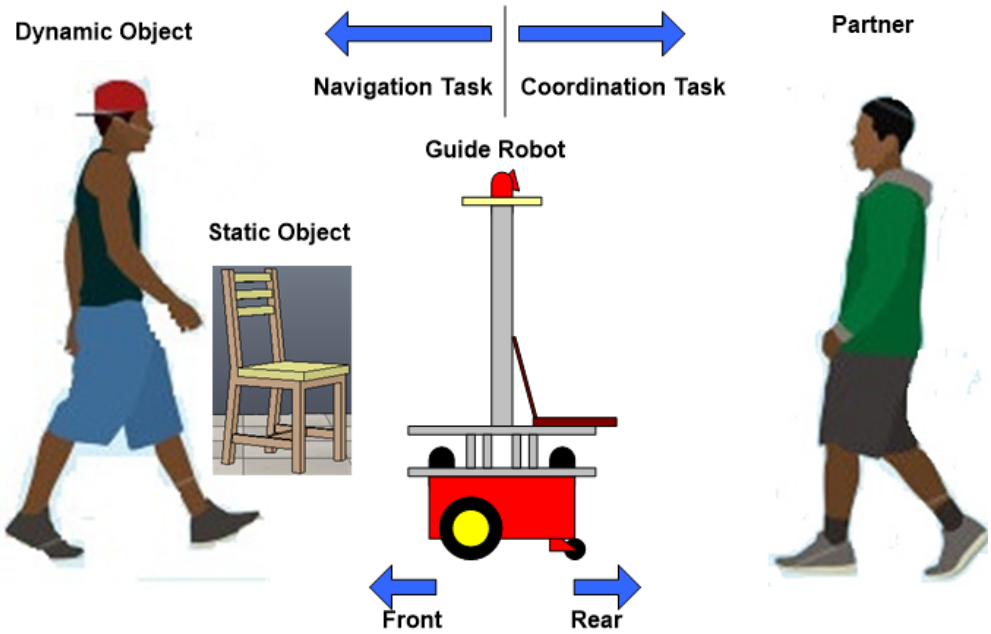


Figure 3.1: Basic tasks in guiding process.

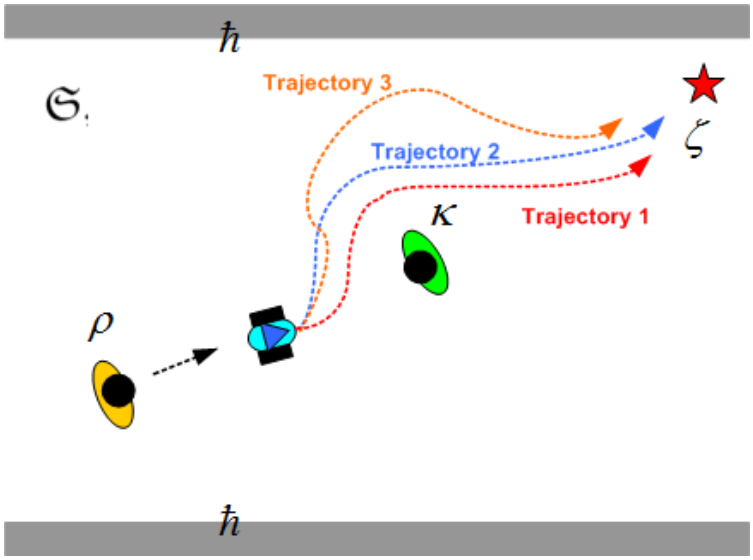


Figure 3.2: An illustration of how a guide robot to plan its motion control while a dynamic obstacle encounters its path. The behavior of a robot when planning and executing its motion control may exhibit different type of trajectory; (Trajectory 1) gentle, (Trajectory 2) mediocre and (Trajectory 3) sprightful.

- How to plan a motion and control of the robot for navigating in a social environment that deals with safety and smoothness?
- What is the most effective and efficient model which can tackle both basic tasks?
- How to teach the skills of guide robot independently?

From Fig. 3.2, let us consider that a human partner, ρ, is guided by a guide robot towards a specific goal, ζ. The guide robot guides the human partner through a social environment, $\mathfrak{S}$, which may be encountered by dynamic obstacles, κ, and static obstacles, $\hbar$. In a particular situation, such as a dynamic obstacle encounters the guide robot's path during guiding the human partner, the robot requires to plan its motion control by considering the human partner's relative position to meet with the human partner comfort, and the dynamic obstacle position to deal with safety and smoothness. To meet such requirement, answering the open research questions one by one is necessary.

3.2 Maintaining Coordination Task with the Human Partner

The most crucial part of a guide robot is how to maintain coordination with the human partner. We assumed that the human partner can be anyone who is not familiar with the guiding protocols and simply follow the guide robot. In this section, we focus our attention on how a coordination task between a guide robot with a human partner must be maintained as well as possible.

A coordination task in guiding processes requires a guide robot to do a specific function which can influence the human partner awareness for doing a particular task that makes them work for the same purpose. To ascertain whether the human partner has an awareness of the process being undertaken, the guide robot must do an evaluation. To perform the evaluation, it is necessary to define the parameters that can be measured from the human partner. The most potential parameters of the human partner are his/her intention and attention to the process. We assume that the intention can implicitly be expressed by the human partner effort to always keep a distance with the guide robot and where his/her direction is. Attention can be shown by the level of focus to the intended direction or the position of the guide robot.

Therefore, we designed our robot to be more responsible by including some abilities: detecting and identifying the human partner, measuring the human partner distance and orientation which respect to the partner's intention parameter, and estimating head orientation to measure the level of attention parameter. The combination of those three abilities is used to identify human partner's behaviors along the guiding process.

3.2.1 Human Partner Identification

In the previous chapter, we have discussed that the guided person should be served as well as possible. Leaving behind or losing the person is prohibited. An ability to relocate and to reidentify the correct person during the guiding process is, however, crucial. A combination of detection, recognition and tracking are usually implemented to recover such kind of problem in an online human-robot interaction system. By an assumption that the human partner always following the robot from behind, the human partner can be observed from his/her front side. This means that features such as face and clothing color are effective for identifying a person from the image.

3.2.1.1 Human Detection and Tracking

The presence of a human can be detected using image-based applications [47, 48] and a range data-based application [49, 50]. Each approach has its own advantages and disadvantages. An image-based human detection can provide a more detail information from the detected human, while the range data-based cannot. An image-based human detection is easily influenced by external disturbances such as lighting, while this does not occur in the range data-based. In addition, an image-based detection also requires more computing time than the range-based. However, since we need to obtain more information from the detected human, we chose to use image-based human detection rather than range data-based detection. The most suitable work which deals with our requirement is Haar-based human upper body detector [47].

Haar-based human upper body detector uses a set of Haar models to find a set of unique features of an object. Those features are then simplified using an integral image technique to reduce

the feature size and accelerate the computation time. Finally, an Adaptive Boosting (AdaBoost) is used to classify whether a region of interest contains human upper body parts or not. Based on our experiments, this detector consistently detects the existing of human in an image with a good illumination condition.

By using Haar-based human upper body detector, we can obtain a top-left position of the human upper body area and its width-height, which can be represented as the human upper body region of interest (ROI), $UB = \{x, y, w, h\}$. To get more stable detection, we adopt the work of [51, 52] for projecting the position of the human partner from the 2D image coordinate to the 3D world coordinate. Let us consider a pinhole camera model, with the following parameters: focal length f_c, camera height y_c, horizontal center point μ_c, and horizon position v_0. According to [52], the projection to the world coordinate is given by

$$W_d = \begin{bmatrix} \frac{y_c(\mu_d - \mu_c)}{v_d - v_0} \\ \frac{f_c y_c}{v_d - v_0} \\ \frac{h_d y_c}{v_d - v_0} \end{bmatrix}, \tag{3.1}$$

$$\begin{aligned} \mu_d &= x + 0.5w, \\ h_d &= 3.3h, \\ v_d &= y + h_d, \end{aligned} \tag{3.2}$$

where (μ_d, v_d) is the bottom center point of the extended bounding box (we extend the bounding box by approximating the human height is 3.3 times of the bounding box height) and h_d is the height of each detected person d in the 2D image. Vector $W_d = \left[x_d^{world}, y_d^{world}, z_d^{world}\right]^T$ denotes the position $(x_d^{world}, y_d^{world})$ in the real world relative to the camera position and the height h_d^{world} of the detected person d in the image.

The horizon position v_0 is obtained by collecting line segments in the image using Hough line detector and performing RANSAC to evaluate all segments and get the vanishing point (intersection point of all line segments). This horizon estimation is done offline and we use it as a pre-calibrated value for the online detection and tracking. For each frame, we send the position $(x_d^{world}, y_d^{world})$ to the tracker for getting the movement estimation in the real world.

We apply a Kalman filter to assist our detector with tracking the detection result. Our state model is composed of the detected body position in the world coordinate $(x_d^{world}, y_d^{world})$, its derivative $(\dot{x}_d^{world}, \dot{y}_d^{world})$, and the estimated person's real height (h_d^{world}). A constant velocity model is utilized to model the person's movements in the world coordinate by considering a time interval Δt:

$$
\begin{aligned}
x_d^{world}(t) &= x_d^{world}(t-1) + \Delta t \dot{x}_d^{world}(t-1) + \varepsilon_{x_d^{world}}, \\
y_d^{world}(t) &= y_d^{world}(t-1) + \Delta t \dot{y}_d^{world}(t-1) + \varepsilon_{y_d^{world}}, \\
\dot{x}_d^{world}(t) &= \dot{x}_d^{world}(t-1) + \varepsilon_{\dot{x}_d^{world}}, \\
\dot{y}_d^{world}(t) &= \dot{y}_d^{world}(t-1) + \varepsilon_{\dot{y}_d^{world}}, \\
h_d^{world}(t) &= h_d^{world}(t-1) + \varepsilon_{h_d^{world}}.
\end{aligned} \tag{3.3}
$$

The state can be expressed as a tuple $\mathcal{Z}_d = \{x_d^{world}, y_d^{world}, \dot{x}_d^{world}, \dot{y}_d^{world}, h_d^{world}\}$. Our Kalman filter model is expressed as $\mathcal{Z}_d(t) = \mathbf{F}(t)\mathcal{Z}_d(t-1) + \varepsilon(t)$, where $\mathbf{F}(t)$ is the state transition model and $\varepsilon(t) = N(0, Q(t))$ is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance $Q(t) = \{0.02, 0.02, 0.01, 0.01, 0.02\}$. The human position in the real world coordinate is then back projected to the image by following equations as follows.

$$
\begin{aligned}
\mu_d &= x_d^{world} \frac{(v_d - v_0)}{y_c} + \mu_c, \\
v_d &= \frac{f_c y_c}{y_d^{world}} + v_0, \\
h_d &= h_d^{world} \frac{v_d - v_0}{y_c}.
\end{aligned} \tag{3.4}
$$

3.2.1.2 Face Feature

In a friendly interaction among humans, recognizing and remembering each other is a common way to start a communication. A unique feature that can be used to distinguish each other is a face. Someone usually knows the other by recognizing his/her face first rather than other attributes, i.e., clothing, accessories, etc. This face recognition usually performs once in advanced and should be remembered for next interactions. When one is researching for a person who is known in advanced in the middle of a crowd of many others, the best approach is by recognizing each face in the crowd.

However, this feature usually fails when an external disturbance such as changes in illumination drastically happens. Therefore, another ability to recognize a person in any conditions should take a part to be considered. We use our illumination invariant face conversion method in [40] to robustly detect and convert the region of interest (ROI) of the face (see Fig. 3.3(a)).

Briefly, our illumination invariant face conversion method analyzes the characteristic of shadow or reflectivity occurs on the face. Our analysis based on Fuzzy Inference System (FIS) [53]. This FIS is used to provide an appropriate pixel by pixel based contrast ratio control. To provide a better control, the Fuzzy's rule is optimized using Genetic Algorithm (GA) [45]. The collaboration of FIS and GA is proven to be effective to improve two face related tasks: face detection and face recognition in a harsh illumination condition. The detail of this method can be found in chapter 4.

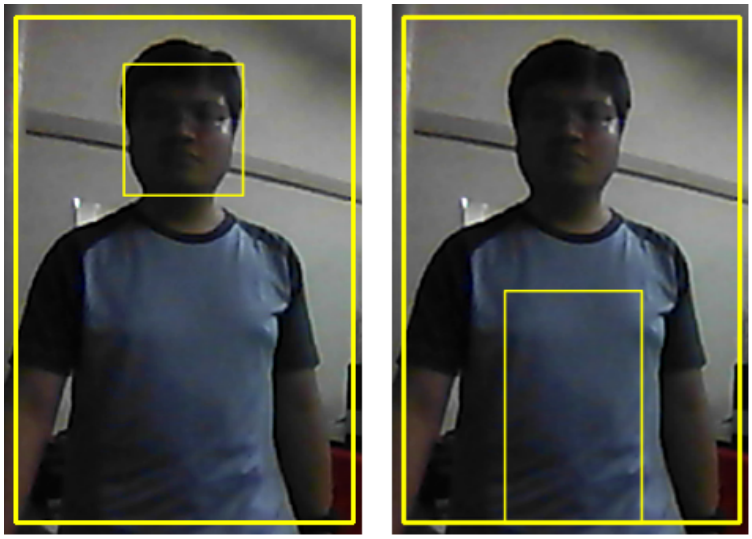


Figure 3.3: A region of interest (ROI) of the person's features: (a) face feature and (b) clothing color feature that are used for identifying person.

3.2.1.3 Clothing Color Feature

Another possible feature that can be used to distinguish person is clothing color. We can expect that clothes used by each person may vary. However, since the robot may guide the human partner to a particular destination through various environments, expecting a good illumination condition is difficult. A color histogram is one of the most popular representations of color modeling. We use a histogram of Hue-Saturation (HS-Histogram) to invariantly recognize the human partner's clothes due to lighting changes. To obtain a histogram, we set an ROI from the lower-half of the detected upper body, as shown in Fig. 3.3(b).

3.2.1.4 Updating Features

Before starting the guiding process, we collected the candidate's face and clothing color of the human partner and set them as the primary reference. When the guiding process begins, the system will retrieve a new face and clothing color as the secondary reference. Therefore, we have two sets of features for detecting the human partner. Due to the uncertain lighting situation may be experienced by the guide robot, updating features is crucial. We update the secondary reference of the face and the clothing color features by assigning constant thresholds, τ_{face} and τ_{cloth} for the face and the clothing color features, respectively. A new input of each feature that is higher than its threshold will be used to replace the previous secondary reference. The primary reference is not updated in order to keep the original information of the human partner. We argue that this update will be very useful for strengthening the detection and tracking of the correct human partner in the current situation.

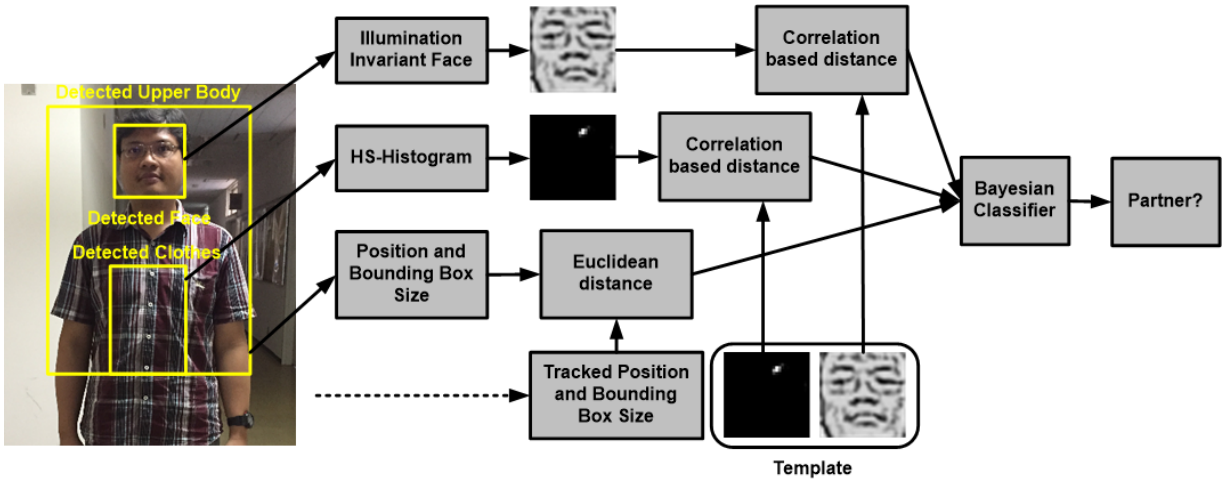


Figure 3.4: Combination of illumination invariant face, HS-Histogram of clothes, and people's position and bounding box size to identify the guided partner.

3.2.1.5 Distinguishing Human Partner

We evaluate the combination of the converted face, the HS-histogram of clothing color and body position within a detected upper body frame as the features as shown in Fig. 3.4. A naive Bayes classification is used due to its simplicity since it simply assumes that the value of a particular feature is independent of the value of any other feature, given the class variable. This kind of classifier also do not require training in advanced, only requires assumptions.

To distinguish the human partner from the other detected persons, let $p(a_i)$ be the probability that the i-th person is the human partner and $\mathbf{b}$ be the observation of the three features. Using Bayes' theorem, the posterior probability $p(a_i|\mathbf{b})$ is given by

$$p(a_i|\mathbf{b}) = \Omega p(\mathbf{b}|a_i)p(a_i), \tag{3.5}$$

$$p(\mathbf{b}|a_i) = p(b_{face}|a_i)p(b_{cloth}|a_i)p(b_{position}|a_i), \tag{3.6}$$

where $p(\mathbf{b}_*|a_i)$ is the likelihood obtained from each feature, and Ω is a normalization constant. The probability is updated every time a new observation is obtained, the probabilities are normalized to sum up to 1.0.

A correlation based nearest neighbor (C-NN) is used as a distance metric between the predefined face and clothes, and the new input face and clothes. The correlation coefficients of the converted face and HS-histogram of clothing color are respectively used to calculate the likelihood of face feature and clothing color feature for our naive Bayes classifier.

$$r_{ob,ref} = \sum_{x,y} \frac{(I^{ob}_{x,y} - \mu_{ob})(I^{ref}_{x,y} - \mu_{ref})}{\sigma_{ob}\sigma_{ref}}, \tag{3.7}$$

where $r_{ob,ref}$ is a correlation coefficient between the observed and the reference. $I^{ob}_{x,y}$ and $I^{ref}_{x,y}$ are the observed and the reference image pixels, respectively. μ_{ob} and μ_{ref} are the mean value of the observed and the reference images, respectively. σ_{ob} and σ_{ref} are the variance of the observed and reference images, respectively. The position of the detected person i-th are then compared with the tracker using a simple euclidean distance based association.

$$d_{ob,tr} = \frac{||(\mathbf{x}_{ob}, \mathbf{x}_{tr})||}{||\mathbf{x}_{ob}||||\mathbf{x}_{tr}||}, \tag{3.8}$$

where $d_{ob,tr}$ is a distance between observation and tracking vectors. $\mathbf{x}_{ob}$ and $\mathbf{x}_{tr}$ are the vector of observation and tracking, respectively.

3.2.1.6 The Human Partner Reidentification and Relocation Strategy

The image-based tracking method described above may sometimes fail to track the human partner. The robot has to be able to recover from such a failure situation. We, therefore, define two states: (1) tracking and (2) lost. In tracking state, the robot performs the usual tracking and identification. When the image-based tracking loses the human partner or probability of the human partner getting low, the robot transits to lost state. In this state, the robot looks for the human partner using the multiple features. If a person who has a high likelihood is found, the robot will set him/her as the human partner to track, and switch back to tracking state.

To reduce the computation time, we only treat one tracker that will be associated with the human partner. Even though in the detection phase the detector detects many persons in an image frame, but the tracker will only be granted to one candidate with the highest probability.

3.2.2 Human Partner Behaviors Observation

During a guiding process, a coordination between the robot and the human partner should always be maintained. Maintaining the robot's task to finish its job is easier than the human partner because the human partner may be attracted by other interesting things and postpone the guiding task. In order to prevent unexpected situations, i.e., the human partner left behind or lost, we equip our robot with an ability to always observe the human partner behaviors. In the guiding process, the human partner behaviors can be shown by his/her intention to always follow the guide robot, and his/her attention to frequently focus to the guide robot or to the guiding direction. We combine two parameters, namely the human position and walking direction in world-coordinate, and facing direction based human attention to describing the human partner's behavior in the context of guiding.

3.2.2.1 Estimating Human Partner Intention

The human partner intention can be indicated using his/her relative distance to the robot and the walking direction. When the distance between the human partner and the robot is constant or closer, it simply indicates that the human partner follows the robot in the same direction. The movement of the human partner can be derived from the change of his/her position. We have described how to obtain the targeted human partner position in the real world coordinate. In this section, we calculate the relative distance, D, and the relative angle, φ, as follows.

$$D = \sqrt{(x^{world})^2 + (y^{world})^2} \tag{3.9}$$

$$\varphi = \tan^{-1}\left(\frac{y^{world}}{x^{world}}\right) \tag{3.10}$$

The walking orientation, α, can be calculated from the change in position, $(\Delta x^{world}, \Delta y^{world})$, and is represented as follows.

$$\begin{aligned} \Delta x^{world} &= x^{world}(t) - x^{world}(t-1), \\ \Delta y^{world} &= y^{world}(t) - y^{world}(t-1), \\ \alpha &= \tan^{-1}\left(\frac{\Delta y^{world}}{\Delta x^{world}}\right) \end{aligned} \tag{3.11}$$

To keep the stability of the direction, we also apply a Kalman filter to assist our detection result. The state model is composed of the calculated walking orientation, α, and its derivative, ω. A constant angular velocity model is applied to model the change in orientation by considering a time interval Δt:

$$\begin{aligned} \alpha(t) &= \alpha(t-1) + \Delta t\,\omega(t-1) + \varepsilon_\alpha, \\ \omega(t) &= \omega(t-1) + \varepsilon_\omega. \end{aligned} \tag{3.12}$$

The state can be expressed as a tuple $\mathcal{W} = \{\alpha, \omega\}$. Our Kalman filter model is expressed as $\mathcal{W}(t) = \mathbf{G}(t)\mathcal{W}(t-1) + \varepsilon(t)$, where $\mathbf{G}(t)$ is the state transition model and $\varepsilon(t) = N(0, Q(t))$ is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance $Q(t) = \{1, 0.01\}$.

3.2.2.2 Estimating Human Partner Attention

Another important cue for human social interaction is the head orientation. One can say now paying attention to something when he/she looked into a certain direction. The change in orientation of the head can be used as a reference to evaluate person's attention. To estimate the head orientation, we use our method in [44] that will be described in detail in Chapter 5.

Briefly, this method is basically based on a 2D appearance of the head. We developed a new descriptor that is inspired by the Weber-based feature, which has been successfully implemented for robust texture analysis, and the gradient which performs well for shape analysis. To further enhance the orientation differences, we combine them with an analysis of the intensity deviation. The position of a pixel and its intrinsic intensity are also considered. All features are then composed as a feature vector of a pixel. The information carried by each pixel is combined using a covariance matrix to alleviate the influence caused by rotations and illumination. As the result, our descriptor is compact and works at high speed. We also apply a weighting scheme, called Block Importance Feature using Genetic Algorithm (BIF-GA), to improve the performance of our descriptor by selecting and accentuating the important blocks.

Experiments on three head pose databases demonstrated that our method outperforms the current state-of-the-art methods. A combination with a head detection and tracking made it can estimate the human head orientation in real applications.

3.2.3 Social Force Coordination Model

We have described several approaches for navigating a mobile robot in a social environment in Section 2.2.1.2. Due to our concern to study about the guide robot behaviors, we chose the Social Force Model (SFM), a microscopic approach for modeling agent behaviors. Our rationales for using the Social Force Model are it needs low computational speed, it is designed to deal with social interaction, and it is more flexible if implemented in real applications because it does not require the representation of space in the form of a grid.

In a guiding context, the main key of a successful coordination task with the human partner is by evaluating his/her action behavior along guiding duration. We have described some features with respect to the human partner behaviors, including the human partner's relative position in a world coordinate, his/her movement direction, and his/her head orientation. So far we only know two types of forces on the social force model that depends on the type of object, namely a repulsive force, and attractive force. A repulsive force occurs when an agent feels increasingly uncomfortable the closer he/she gets to a strange agent, who may react in an aggressive way. On the other hand, an attractive force occurs when an agent feels interesting to something or someone (e.g., window display, friends, street artist, etc.).

We are making an exception in the case of the human partner because he/she might be away from the guide robot either intentionally or unintentionally, or come closer to the robot because suspicion will be left of the guide robot. Since all of the human partner's behavior affects to the guide robot guiding behavior, we grant two types of forces at once to the human partner that depends on his/her behavior at that time. When the human partner is approaching the guide robot, the guide robot will assume that the human partner is within its observation range, and the guide robot can continue to perform the guiding task as a result of the repulsive force given by the human

partner. Conversely, when the human partner moves away from the guide robot, the guide robot will receive an attractive force with respect to the action behavior exhibited by the human partner.

In the social force model, there are two types of forces which compose a repulsive force, they are a social repulsive force and a physical repulsive force. Also, there are two types of forces which compose an attractive force, they are a social attractive force and a physical attractive force. A social repulsive force is a force associated with an abstract information that can be psychological conditions which greatly affects an individual's behavior and causes the individual to avoid it physically, while a physical repulsive force is a force associated with a more concrete information that can physically be sensed or measured and has a physical impact to the individual act which should be avoided physically. A social attractive force is a force associated with social interest against something and be able to influence the behavior of a person to approach it, while a physical attractive force is a force associated with physical interest against a concrete information and has a physical impact to the individual act which should be approached physically.

Considering the above description, an associated force with respect to the human partner relative position can be expressed as a combination of a social force and a physical force (can be repulsive or an attractive) as a function of the distance. Since the social force is more abstract, it can be represented using a specific function such as a bipolar sigmoid function or a logit function or a linear equation. In this work, we use a bipolar sigmoid function since it has an idle value after a particular input (in our case, the input is distance). The relative distance based force can be expressed as follows.

$$\mathbf{f}^{\rho}_{soc} = \left[2k^{\rho} \frac{1}{1+\exp\left(-\frac{d_{R\rho}-\mu_{\rho}}{\Psi^{\rho}}\right)} \right] - k^{\rho}\mathbf{e}_{R\rho}, \tag{3.13}$$

$$\mathbf{f}^{\rho}_{phy} = k^{\rho}(d_{R\rho}-\mu_{\rho})\mathbf{e}_{R\rho}, \tag{3.14}$$

$$\mathbf{F}^{\rho} = \mathbf{f}^{\rho}_{soc} + \mathbf{f}^{\rho}_{phy}, \tag{3.15}$$

where $\mathbf{F}^{\rho}$, $\mathbf{f}^{\rho}_{soc}$, and $\mathbf{f}^{\rho}_{phy}$ are the total force, the social force and the physical force given by the human partner to the guide robot with respect to his/her relative position, respectively. k^{ρ} is the magnitude of force with respect to the target partner relative position. $d_{R\rho}$ is a relative distance between the human partner and the guide robot. μ_{ρ} is a mean value which represents a comfortable distance when the human partner follows the guide robot. Currently, this mean value is set to 2.2 meters. We conducted a separate study to determine this value in Chapter 6. Ψ^{ρ} is an effective range of influence of the force. $\mathbf{e}_{R\rho}$ is a vector indicating the direction from the human partner to the guide robot, and vice versa.

We also formulate the human partner movement direction under social attractive force assumption as follows.

$$\mathbf{F}^{\alpha} = k^{\alpha}(1-\cos(\alpha+\varphi))\mathbf{e}_{R\rho}, \tag{3.16}$$

where $\mathbf{F}^{\alpha}$ is an attractive force due to human movement direction, k^{α} is the magnitude of the force with respect to human movement direction, α and φ are the human partner movement direction and the relative direction of the human partner from the guide robot, respectively. The last feature is the human partner's head direction that can be expressed as follows.

$$\mathbf{F}^{\beta} = k^{\beta}(1-\cos(\beta+\varphi))\mathbf{e}_{R\rho}, \tag{3.17}$$

where $\mathbf{F}^{\beta}$ is an attractive force due to human partner's head orientation, k^{β} is the magnitude of the force with respect to human partner's head orientation, β is the human partner's head orientation. Since the human partner movement direction and head orientation only affect to the social related context, we defined them as the social forces only.

Therefore, we can define the resulting force with respect to the coordination with the human partner, $\mathbf{F}^{coord}$, used for influencing the motion planning and control of the guide robot as follows.

$$\mathbf{F}^{coord} = \mathbf{F}^{\rho} + \mathbf{F}^{\alpha} + \mathbf{F}^{\beta}. \tag{3.18}$$

The parameters that are involved in $\mathbf{F}^{coord}$ are k^{ρ}, Ψ^{ρ}, k^{α}, and k^{β}.

3.3 Performing Robot Navigation in Social Environment

Besides our focus to maintain a coordination with the human partner, robot navigation task is one of the most important parts that should be considered. A successful execution of the guiding process is determined by how successfully the robot navigates the human partner towards a goal. The robot is usually equipped with sensors to perceive and sense the presence of other objects and knows its current state. To deal with this task, performing object detection is crucial. It is expected that the robot can analyze the situation and initiates an appropriate action to be executed. To deal with this task, a module of the motion planning and control model is important.

3.3.1 People Detection and Tracking

Detecting and tracking a person as a dynamic object in an environment is necessary for successful and safe robot navigation. This function uses range data from a single laser range finder (LRF) [49, 50]. A combination of LRFs and cameras for more robust tracking of the target person is evaluated in [54]. In our work, persons other than the human partner are considered as dynamic obstacles. Since knowing only their positions is enough, we use only a single LRF to detect and track them.

We use a method by Koide and Miura [54] to find a person's position from his/her legs. Leg candidates are detected at local minima in range data, and then features which represent the shape of the candidate are extracted. SVM [55] and Adaboost [56] are used to classify each detected candidate as a person or not from its extracted features. A Kalman filter with a constant velocity model and a Nearest Neighbor (NN) data association are adopted to track the detected person.

3.3.2 Static Object Detection

Besides detecting a person as a dynamic object, the guide robot should be able to map the physical environment at least in 2D. This map is important due to the guide robot is usually operated indoor or outdoor, where there are many static objects along the path. We use range data to map the local environment around the guide robot to determine the passable and the non passable area.

3.3.3 Social Force Navigation Model

Under Social Force perspective, a guide robot is a mobile robot with mass m which tries to reach a goal with a desired speed v^0 in the desired direction e^0, thereby the desired velocity $\mathbf{v}^0 = v^0 e^0$. The robot tends to adapt its actual velocity $\mathbf{v}$ to the desired velocity within a period of relaxation time τ. Hence, the basic equation of motion of the robot towards a goal is given by the social force term:

$$\mathbf{F}^g = m\frac{\mathbf{v}^0 - \mathbf{v}}{\tau} \tag{3.19}$$

During its movement from the start position to the goal, the robot will try to keep a distance both from the closest dynamic obstacle, κ, and the closest static obstacle, $\hbar$, by the interaction force $\mathbf{F}^\kappa$ and $\mathbf{F}^\hbar$ respectively, as shown in Fig. 3.5(a). This behavior is referred to in term of the repulsive effects. Both $\mathbf{F}^\kappa$ and $\mathbf{F}^\hbar$ are the result of a combination of social repulsive force, $\mathbf{f}_{soc}$, and physical repulsive force, $\mathbf{f}_{phy}$. The repulsive force with respect to the closest dynamic obstacle can be expressed as

$$\mathbf{F}^\kappa = \mathbf{f}^\kappa_{soc} + \mathbf{f}^\kappa_{phy}, \tag{3.20}$$

$$\mathbf{f}^\kappa_{soc} = k^\kappa \exp\left(\frac{r_{R\kappa} - d_{R\kappa}}{\Psi^\kappa}\right)\mathbf{e}_{R\kappa}\omega, \tag{3.21}$$

$$\mathbf{f}^\kappa_{phy} = k^\kappa (r_{R\kappa} - d_{R\kappa})\mathbf{e}_{R\kappa}, \tag{3.22}$$

where k^κ is the magnitude of force with respect to a dynamic obstacle. $r_{R\kappa} = r_R + r_\kappa$ is a sum of the robot's radius, r_R, and the dynamic obstacle's radius, r_κ, at an intersection point between

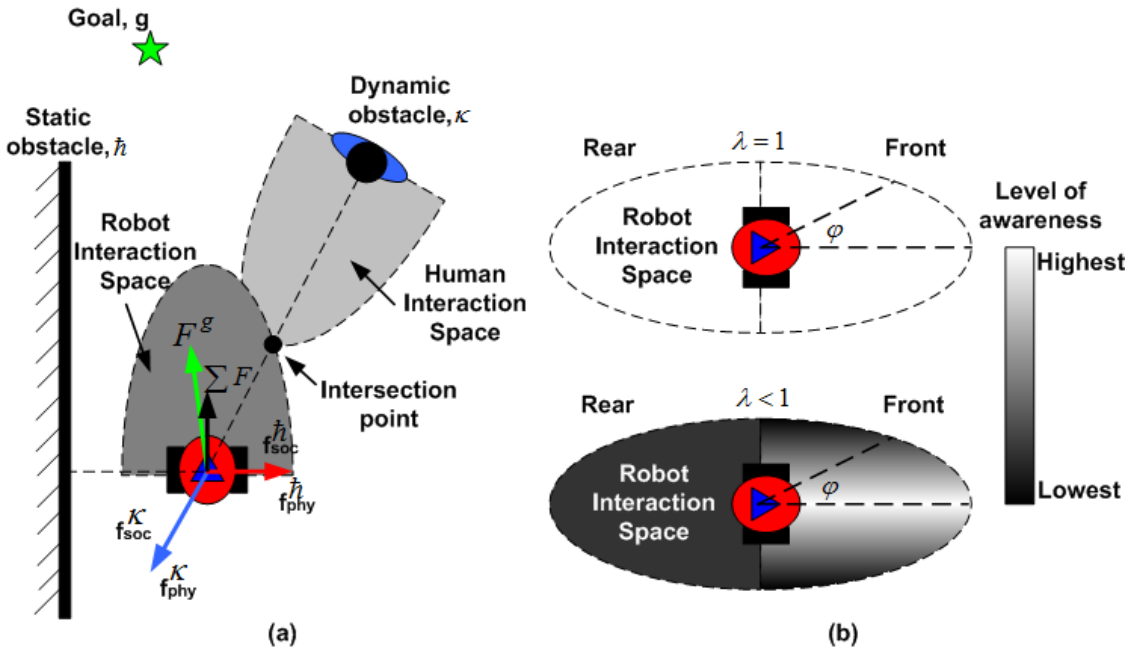


Figure 3.5: The social force for modeling robot navigation. (a) An example of the robot behavior when interacts with the environment, where there are three forces generated by three elements, they are the dynamic obstacle, the static obstacle and the desired goal. (b) An anisotropic factor to define a limited awareness of an agent; $\lambda = 1$ means the same level of awareness for all φ, and $\lambda < 1$ means the level of awareness is gradually decreasing depends on φ.

their interaction space. $d_{R\kappa}$ is a distance between the robot to the closest dynamic obstacle. Ψ^{κ} is an effective range of influence of the force with respect to a dynamic obstacle. $\mathbf{e}_{R\kappa}$ is a vector indicating the direction from the dynamic obstacle to the robot. When the distance between the robot and the dynamic obstacle, $d_{R\kappa}$, is larger than $r_{R\kappa}$, the force will be ignored.

Due to a limited field of view of human, an anisotropic factor expressed by $\omega = \lambda + 0.5(1-\lambda)(1+\cos(\varphi))$ is introduced into the equation. λ is a parameter which defines an anisotropic influence region that represents the fact that the obstacle in front of an agent are usually more relevant than that located behind it and φ is an angle between obstacle and robot as illustrated in Fig. 3.5(b). The repulsive force with respect to the closest static obstacle can be expressed as

$$\mathbf{F}^{\hbar} = \mathbf{f}^{\hbar}_{soc} + \mathbf{f}^{\hbar}_{phy}, \tag{3.23}$$

$$\mathbf{f}^{\hbar}_{soc} = k^{\hbar} \exp\left(\frac{r_R - d_{R\hbar}}{\Psi^{\hbar}}\right) \mathbf{e}_{R\hbar}\omega, \tag{3.24}$$

$$\mathbf{f}^{\hbar}_{phy} = k^{\hbar}(r_R - d_{R\hbar})\mathbf{e}_{R\hbar}, \tag{3.25}$$

where $k^{\hbar}$ is a magnitude of force with respect to a static obstacle. r_R is a robot's radius. $d_{R\hbar}$ is a distance between the robot to the closest static obstacle. $\Psi^{\hbar}$ is an effective range of influence of the

force with respect to a static obstacle. $\mathbf{e}_{R\hbar}$ is a vector indicating the direction from the dynamic obstacle to the robot. When the distance between the robot and the static obstacle, $d_{R\hbar}$, is larger than r_R, the force will be ignored. We can define a resulting force with respect to the navigation among social environment, $\mathbf{F}^{nav}$, as follows.

$$\mathbf{F}^{nav} = \mathbf{F}^{g} + \mathbf{F}^{\kappa} + \mathbf{F}^{\hbar}. \tag{3.26}$$

The parameters that are involved in $\mathbf{F}^{nav}$ are k^{κ}, Ψ^{κ}, $k^{\hbar}$, $\Psi^{\hbar}$, λ, v^0 and τ.

3.4 Integrating Tasks

3.4.1 Regulating the Guide Robot

We have defined two basic tasks in our guide robot system. All tasks have been formulated under the same model, that is the social force model. Therefore, integrating all tasks is easier. The total force at the time step t can be formulated as

$$\mathbf{F}_t^{guide} = \mathbf{F}_t^{coord} + \mathbf{F}_t^{nav}. \tag{3.27}$$

From the above expression, the planning of motion control of the guide robot can easily be done by solving the force formulation with respect to the velocity as follows.

$$\mathbf{F}_t^{guide} = m\frac{d\mathbf{v}_t}{dt}. \tag{3.28}$$

As the result, a velocity can be obtained from that expression to regulate the guide robot speed and heading direction, as follows.

$$\mathbf{v}_t = \frac{\mathbf{F}_t^{guide}}{m}t. \tag{3.29}$$

3.4.2 PID Controller

We realize that a robot which is equipped with the social force model can behave well when operated in an open space (e.g. outdoor with less static obstacles). However, its behavior drastically different when operated in an indoor space (e.g. corridor, a room which is surrounded by narrow walls). Bouncing effects frequently occur due to an interaction space shape used by the robot, i.e. elliptic or circle. To solve this problem, we use the Q-learning method to optimize the social force model parameters. We expect the over-reactive behavior can be reduced by adaptively adjusting those parameters.

However, in some cases, the bouncing effect still occurs even though we have applied the adaptive parameter adjustment. This problem occurs due to the original characteristic of the social

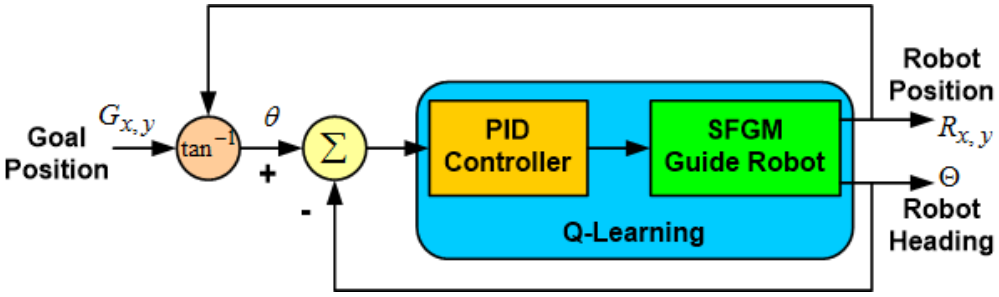


Figure 3.6: Integration between SFGM and PID controller.

force model cannot be completely eliminated. We found that the guiding force, F_t^{guide}, is too late to respond the first bounce, which causes the robot difficult to get back on track towards the goal. Therefore, we employ a Proportional-Integral-Derivative (PID) controller as shown in Fig. 3.6 to reduce bouncing effects exhibited by our Social Force Guiding model (SFGM) by accelerating its heading direction, Θ, towards the goal direction, θ. Employing this PID controller can greatly speed up the Q-learning convergence or reducing the number of learning episodes. We cascade the PID controller in front of our SFGM scheme and optimize the PID coefficients together with the SFGM parameters using Q-learning.

3.5 Self-Learning Strategy

3.5.1 Problem Identification

Under social force perspective, the motion of a guide robot is driven by a velocity as the result of the navigation force and the coordination force at that time step, $\mathbf{F}_t^{guide}$. Obtaining an appropriate $\mathbf{F}_t^{guide}$ for each circumstance is, however, crucial to exhibit a socially acceptable behavior, i.e. smooth and safe. With respect to smooth and safe navigation, a trade-off between keeping direction towards the goal, abilities to avoid collision with obstacles, and keeping coordination with the human partner is a crucial factor to produce a reliable $\mathbf{F}_t^{guide}$. Therefore, we argue that this trade-off problem can be solved using Reinforcement Learning (RL) technique.

3.5.2 Q-learning based Parameters Optimization

Under Reinforcement Learning framework, a problem can be formulated as a discrete time, finite state, finite action MDP [46]. The learning environment can be modeled by a 4-tuple $\{x, a, p, r\}$, where:

- $x \in \mathcal{X}$; $\mathcal{X}$ is a finite set of states.
- $a \in \mathcal{A}$; $\mathcal{A}$ is a set of actions that the agent can perform.

- $p \in \mathcal{P}$; $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \Pi(\mathcal{X})$ is a state transition function, where $\Pi(\mathcal{X})$ is a probability distribution over $\mathcal{X}$. $p(x,a,x')$ represents the probability of moving from state x to x' by performing action a.
- $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to R$ is a scalar reward function.

The goal of the agent in a RL problem is to learn an optimal policy $\Pi^* : \mathcal{X} \to \mathcal{A}$.

Q-learning (QL) is one of the most successful branches of Reinforcement Learning (RL). In Q-learning, on each interaction step an agent senses the current state x of the environment, chooses an action a, executes this action, switches to the state x' of the environment, and receives a scalar reinforcement signal r (a reward or penalty). Q-learning learns the value of each action in each state. This Q-value is referred to as the state-action value and is updated by

$$Q(x,a) = Q(x,a) + \eta\,(r(x,a) + \gamma max Q(x',a) - Q(x,a)), \tag{3.30}$$

where $Q(x,a)$ is a Q value of the state, x, and action, a, pair. η is a learning rate within range (0,1). $r(x,a)$ is a direct reward value for the state-action pair. γ is a discount factor. $maxQ(x',a)$ is the estimated maximum Q-value of the next state.

3.5.2.1 Configuring a State

A state describes a local situation faced by the robot during its travel. The local situation is composed of several features as shown in Fig. 3.7. We built and specify our own interaction zone by modifying the proxemic interpersonal distance that was introduced by Hall in [19]. In the proxemic interpersonal distance, each distance is specified within a specific shape, e.g., circular or elliptical. Due to simplify the computational problem and the generation of states, we discretized all features using binary segmentation and assigning a constant real value, w, as the identity of each segment as shown in Table. 3.1. The state with respect to each feature can be expressed as

$$x_1 = \sum_{i=0}^{3} p_i w_i. \tag{3.31}$$

The human partner's body orientation is roughly divided into three directions $\mathbf{B} = \{b_0, b_1, b_2\}$, to express a linguistic meaning, such as straight, oblique and turn. Again, we assume that different direction in the same division, b_i, has the same impact for generating the force. The human partner body orientation is also coded as binary digit "0" and "1". To simplify the generation of state, we

Table 3.1: The code of the presence of the human partner in the coordination zone

w_0	w_1	w_2	w_3
1	2	3	4

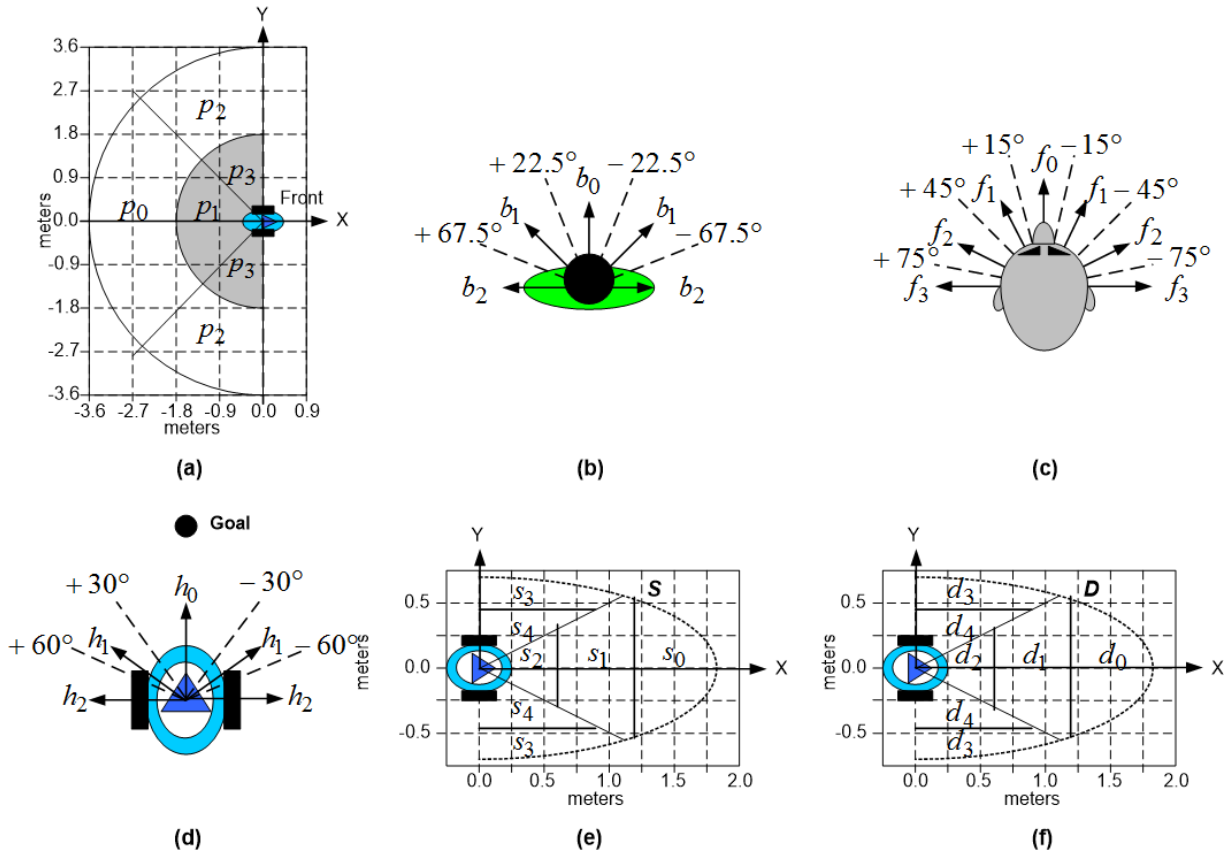


Figure 3.7: Features for composing the state: (a) A zone for observing the human partner position, (b) A division of the human partner's body orientation, (c) A division of the human partner's head orientation, (d) A division of the guide robot's heading direction, (e) A zone for observing static obstacles, and (f) A zone for observing dynamic obstacles.

Table 3.2: The code of the human partner body orientation

w_0	w_1	w_2
5	10	15

assign a constant real value, w, as the identity of each division as shown in Table. 3.2. The state with respect to the human partner body orientation, x_2, can be expressed as

$$x_2 = \sum_{i=0}^{2} b_i w_i. \tag{3.32}$$

The human partner's head orientation is roughly divided into four directions $\mathbf{F} = \{f_0, f_1, f_2, f_3\}$, to express a linguistic meaning, such as looking forward and looking to the side. Again, we assume that different direction in the same division, f_i, has the same impact for generating the force. The human partner head orientation is also coded as binary digit "0" and "1". To simplify the generation of state, we assign a constant real value, w, as the identity of each division as shown in Table. 3.3. The state with respect to the human partner head orientation, x_3, can be expressed as

Table 3.3: The code of the human partner head orientation

w_0	w_1	w_2	w_3
20	40	60	80

Table 3.4: The code of the robot heading orientation

w_0	w_1	w_2
0	100	200

$$x_3 = \sum_{i=0}^{3} f_i w_i. \tag{3.33}$$

The robot's heading orientation is roughly divided into three directions $\mathbf{H} = \{h_0, h_1, h_2\}$, to express a linguistic meaning, such as straight, oblique and turn. Again, we assume that different direction in the same division, h_i, has the same impact for generating the force. The human partner head orientation is also coded as binary digit "0" and "1". To simplify the generation of state, we assign a constant real value, w, as the identity of each division as shown in Table. 3.4. The state with respect to the robot heading orientation, x_4, can be expressed as

$$x_4 = \sum_{i=0}^{2} h_i w_i. \tag{3.34}$$

Our guide robot is designed to work with safety and smoothness as the constraints. It must be able to detect the presence of obstacles encountered its path. We divided the interaction zone of the static obstacles into five regions $\mathbf{S} = \{s_0, s_1, s_2, s_3, s_4\}$ by following the partition of the proxemic interpersonal distances, where the three regions in front represent the intimate distance s_2, the personal distance s_1, and the social distance s_0 in front of the person. The other two regions on the right side and the left side represent the intimate-personal distance s_4, and the social distance s_3. Two lines with angle of $\pm 30^o$ divide the region into "in front of the robot" which includes $\mathbf{S} = \{s_0, s_1, s_2\}$ and "side the robot" which includes $\mathbf{S} = \{s_3, s_4\}$. We assume that right side and the left side have the same impact for generating the force, even with a different direction. The presence of the static obstacles is coded as binary digit "0" and "1". To simplify the generation of state, we assign a constant real value, w, as the identity of each region as shown in Table. 3.5. The state with respect to the presence of the static obstacles in the interaction zone, x_5, can be expressed as

$$x_5 = \sum_{i=0}^{4} s_i w_i. \tag{3.35}$$

Lastly, we also divided the interaction zone of the dynamic obstacles into five regions $\mathbf{D} = \{d_0, d_1, d_2, d_3, d_4\}$ by the same reason as stated in the partition of static obstacle's interaction

Table 3.5: The code of the presence of the static obstacles in the interaction zone

w_0	w_1	w_2	w_3	w_4
300	600	900	1,200	2,400

Table 3.6: The code of the presence of the dynamic obstacles in the interaction zone

w_0	w_1	w_2	w_3	w_4
3,600	7,200	10,800	14,400	28,800

zone. Two lines with angle of $\pm 30^o$ divide the region into "in front of the robot" which includes $\mathbf{D} = \{d_0, d_1, d_2\}$ and "side the robot" which includes $\mathbf{D} = \{d_3, d_4\}$. We assume that right side and the left side have the same impact for generating the force, even with a different direction. The presence of the dynamic obstacles is coded as binary digit "0" and "1". To simplify the generation of state, we assign a constant real value, w, as the identity of each region as shown in Table. 3.6. The state with respect to the presence of the dynamic obstacles in the interaction zone, x_6, can be expressed as

$$x_6 = \sum_{i=0}^{4} d_i w_i. \tag{3.36}$$

Finally, we can obtain a state which can be calculated from the final configuration of all features as follows.

$$x = x_1 + x_2 + x_3 + x_4 + x_5 + x_6. \tag{3.37}$$

3.5.2.2 Defining and Selecting an Action

In the MDP framework, an action is a set of variables that can be chosen and executed to switch from current state to the other. In our case, an action is defined as a set of social force guiding model parameters, $a = \{k^\kappa, \Psi^\kappa, k^\hbar, \Psi^\hbar, \lambda, k^\rho, \Psi^\rho, k^\alpha, k^\beta, K_p, K_i, K_d\}$. Taking an action means adjusting those parameters. v^0 and τ are not included since adjusting them is very risky and greatly affect the robot stability. We constantly set v^0 below the maximum robot speed permitted in the specification and constantly set τ in approximation to our algorithm computation time.

For each step of the training episode, an action is selected by following the ε-greedy approach [46]. The ε is arbitrarily set in advance and gradually decreased until 0.01 to deal with the exploration-exploitation dilemma. Using the ε-greedy approach, only pairs of state-action with probabilities more than the $1-\varepsilon$ will be considered. We simply calculate the probability of a state-action pair, $p(x,a)$, from Q-value normalization as follows.

$$p(x,a) = \begin{cases} 1.0 & \text{if } Q(x,a) = \max Q(x) \\ \frac{Q(x,a) - \min Q(x)}{\max Q(x) - \min Q(x)} & \text{otherwise} \end{cases} \tag{3.38}$$

Enlarging the probability of exploration in advance is important to gain more information and knowledge. After exploring the information up to some limits, exploiting the learned knowledge is performed to further reinforce it.

3.5.2.3 Reward Value

A reward value is used to update the Q-value of a state-action pair. Referring to section 3.5.1, the main objective of using Q-learning is to solve the trade-off problem between those three objectives. We proportionally calculate this reward value using a linear equation for representing the reward function as follows.

$$r = mf(g,h) + c \tag{3.39}$$

where r is a reward value, m is a constant multiplier or a slope, $f(g,h)$ is a function derived from a specific objective, and c is a constant. A small positive reward value (e.g. +1) is required to keep the action to stay on the list of the best actions when a learner successfully does a job. A negative reward value is required to remove the action from the list when a learner failed. If we expect the result of $f(g,h) = [0,1]$, c is a negative reward value, and $m = 1-c$, then by following the formulation, we define three objectives by heuristically setting $c = -10$ as follows.

1. Completing an episode:
$$r_{success} = \begin{cases} +100 & \text{if robot can reach the goal} \\ c & \text{if robot fails} \end{cases}.$$

2. Minimizing the robot runs in different direction with the goal to deal with smooth navigation:
$r_\Theta = (1-c)\cos(\theta_t - \Theta_t) + c,$
where θ_t and Θ_t are the relative goal direction from the robot's position and the robot's heading at time t, respectively.

3. Controlling the magnitude of forces: the dynamic obstacle, ($r_{f_t^\kappa}$), the static obstacle, ($r_{f_t^\hbar}$), and the human partner force, ($r_{f_t^{\rho\alpha\beta}}$) to deal with safety and the human partner comfort:
$$r_{f^\kappa} = \begin{cases} (1-c)\exp^{(-(F_t^\kappa - F_{max}^g)^2/(2(F_{max}^g)^2))} + c & \text{if region } \mathbf{D} = d_1, d_2, d_4, d_1 \cup d_4 \text{ or } d_2 \cup d_4 \\ (1-c)\exp^{(-F_t^\kappa/F_{max}^g)} + c & \text{otherwise} \end{cases}$$
$$r_{f^\hbar} = \begin{cases} (1-c)\exp^{(-(F_t^\hbar - F_{max}^g)^2/(2(F_{max}^g)^2))} + c & \text{if region } \mathbf{S} = s_1, s_2, s_4, s_1 \cup s_4 \text{ or } s_2 \cup s_4 \\ (1-c)\exp^{(-F_t^\hbar/F_{max}^g)} + c & \text{otherwise} \end{cases}$$
$$r_{f_t^{\rho\alpha\beta}} = \begin{cases} (1-c)\exp^{(-(F_t^{\rho\alpha\beta}/F_{max}^g)} + c & \text{if region } \mathbf{P} = p_1 \text{ or } p_3 \\ 1 & \text{otherwise} \end{cases}$$
where F_{max}^g is the maximum F^g without repulsive forces, when $\mathbf{v} = 0$.

The direct reward value of each step, r_i, can be expressed as

$$r_i = r_\Theta + r_{f^\kappa} + r_{f^\hbar} + r_{f_t^{\rho\alpha\beta}}. \tag{3.40}$$

The total reward value, R, after completing one episode is expressed as

$$R = \sum_{i=1}^{N} \gamma^i r_i + r_{success}. \tag{3.41}$$

where $i = 1,2,3,...,N$, N is the number of steps, and γ is a discount rate applied to the expected maximum Q-value of the next state. We applied different reward values on four groups of Q-maps. Q_{k^κ} and Q_{Ψ^κ} are updated using r_{f^κ}, $Q_{k^\hbar}$ and $Q_{\Psi^\hbar}$ are updated using $r_{f^\hbar}$, Q_{k^ρ} and Q_{Ψ^ρ} are updated using $r_{f^{\rho\alpha\beta}}$, and Q_{f^α}, Q_{f^β} and Q_λ are updated using r_i. The main reason for this separation is each group has different impacts on the final control result. Using the same reward value can damage the updated-Q values of the other groups and takes more time to converge.

3.6 System Integration

As the last section of this chapter, we integrate the coordination task, the navigation task, and the self-learning strategy that can be shown in Fig. 3.8. In guiding context, we have two main blocks, they are the social environment and our socially aware guide robot. Our robot perceives and sense the environment situation using two main sensors, namely camera and laser range finder. From both sensors, we extract information surrounding the robot to get the human partner identity, position of dynamic obstacles, and static obstacles position, all relative to the robot position. From all information we obtain, several features can be extracted. Features such as human partner relative distance, intention and attention, dynamic object distance, dynamic object interaction space, static

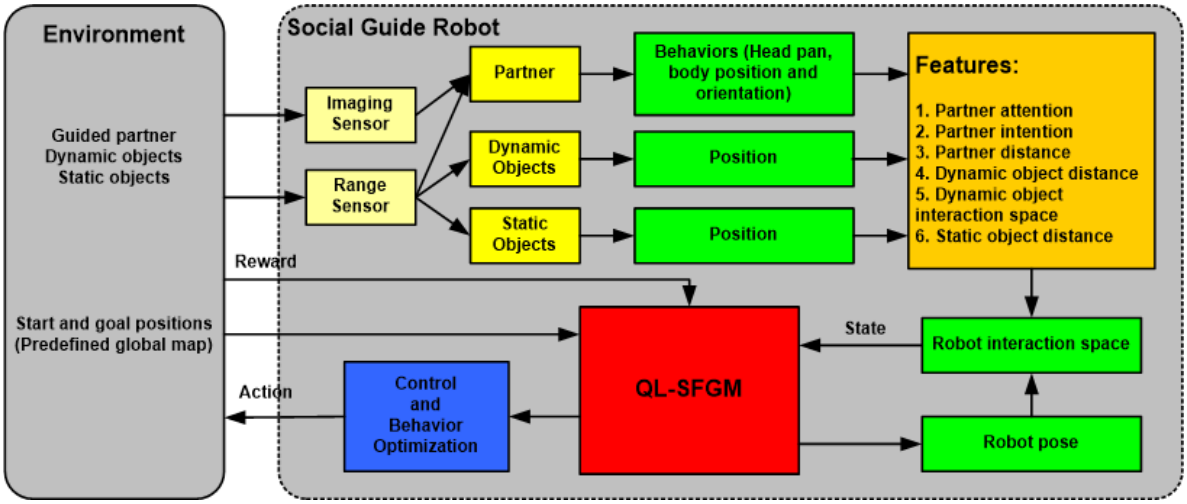


Figure 3.8: Integrating all designed systems of our socially aware guide robot.

object distance, and robot pose are then sent to our Q-learning based Social Force Guiding Model (QL-SFGM) to be analyzed and planned. As the result, our QL-SFGM outputs a control signal to be executed by the robot that meets with the most optimal behavior.

Chapter 4

Supporting Component: Illumination Invariant Face Recognition System

4.1 Introduction

Vision-based human face recognition has been shown to be effective under normal illumination conditions. When it is used under severe illumination conditions, however, the recognition rate drops rapidly. We cannot always expect to have good illumination and changes in illumination are hard to predict and control. When we move through various environments, including both indoor and outdoor, the appearance of facial features change, thereby making it difficult to recognize faces (see Fig. 4.1).

In face recognition problems, the most important factor is a constancy; for each human face's part, color constancy in appearance-based approaches and feature constancy in feature-based approaches will be very helpful for distinguishing each person. A region of interest (ROI) of the face should also be constant. An ROI that is shifted from a correct face region would cause inaccurate recognition results.

Several illumination invariant methods have been proposed to solve these problems. Both appearance-based and feature-based methods are widely used. The appearance-based methods convert an input image into the desired output image, and are categorized into three groups: Histogram transformation, photometric-based methods, and gradient-based methods. Feature-based methods transform an input image into a set of features or descriptors instead of using the intensity of each pixel directly.

Histogram transformation such as Histogram Equalization (HE) and Block Histogram Equalization (BHE) [57] apply a contrast enhancement to the image which works very fast. However, the equalized image frequently experiences washed-out and over-equalized effects in a narrow region with a very large distribution of intensity. The effects of light on a face is manifested by photometric-based methods. It uses human perception theory and illumination properties. Multi-Scale Retinex (MSR) [30], Fuzzy-Retinex (FR) [31], Self Quotient Image (SQI) [32], and Classified-Appearance Quotient Image (CAQI) [33] are able to produce a fairly invariant image conversion. However, these methods are less able to control the final color constancy and sometimes fail to eliminate the boundary region between different illumination effects. These problems can be minimized by specifying some parameters either adaptively or with optimization techniques performed by Adaptive Scale Retinex (ASR) [34] and Optimized-SQI (Opt-SQI) [35], respectively. However,

Figure 4.1: Different visual appearances caused by different illumination conditions.

both of these methods require very long computation periods.

Logarithmic Discrete Cosine Transform (LDCT) [36] discards several first DCT's coefficients which are related to shadow effects. This process runs very fast, but the output image becomes unclear. SVD-Face [37] uses the normalized coefficients of the singular value decomposition (SVD) which are insensitive to different illumination conditions for describing the underlying structures of faces. Mean Estimation (ME) [38] removes the illumination component by subtracting the mean estimation from the original image. The mean estimation is locally obtained from the exponential form of the illumination-reflection model instead of using smoothing techniques. However, both of these methods are difficult to control for the final color constancy of the two areas affected by different lighting effects.

Approaches using a gradient can be found both on appearance-based and feature-based methods. GradientFace (GF) [58] and Block Difference of Inverse Probabilities and Block Variation of Local Correlation Coefficient (BDIP-BVLC) [39] are examples of appearance-based gradient methods. The result of the GF is poor when used for facial images with low lighting. While the sketch-like converted image of the BDIP-BVLC often experience strong edge artifacts at boundaries and frequently fail to maintain their final color constancy. Weber Local Descriptor (WLD) [59], Local Binary Pattern Histogram (LBPH) [60], and Logarithm Gradient Histogram (LGH) [61] also use a gradient to build features. LGH successfully combines the magnitude and the orientation of the logarithmic gradient to form a robust histogram-feature. Its achievement outperforms WLD and LBPH.

We propose a novel appearance-based illumination invariant method that works very fast and can control the final color constancy while reducing the boundary effects to keep the high recognition rate. It controls the image contrast using an illumination invariant model that is supported by the Fuzzy Inference System (FIS) [53]. We optimize FIS rules using Genetic Algorithm (GA) [45], to find the most optimal illumination ratio which is a crucial component in the illumination invariant model. Our goal is the implementation of the method in real conditions. We consider that the appearance-based methods are easier to be used instead of using feature-based methods because of their flexibility when combined with other systems such as Viola-Jones face detector [47]. Here, we show that our illumination invariant method is effective in two face related tasks, face detection and face recognition.

4.2 Illumination Invariant Face Recognition

The human face has distinctive components, texture and contour. When the face is exposed to light coming from various directions, its appearance from the front side will vary considerably. Most illumination invariant methods [30, 31, 32, 33, 34, 35, 36, 37, 38, 39] are capable of producing invariant facial appearances, however, they make the face look flat, unclear, create a boundary effect, differing final color constancy, etc.

Our method takes the advantage of the illumination normalization model proposed in [62]. This model was originally designed for normalizing a partially shadowed area of landscapes using a pair of appearance based segmented regions (shadowed and shadow-free regions). The model considers not only environment based lighting effects but also a direct lighting effect. The model is derived from those two lighting effects and forms a kind of contrast ratio of an image.

4.2.1 Illumination Normalization Model

In general, the reflectance model [63] can be expressed as:

$$I_{x,y} = L_{x,y} R_{x,y}, \tag{4.1}$$

where $I_{x,y}$ is an image intensity at pixel (x,y) that is captured by the camera, $L_{x,y}$ is the illumination that comes to the surface of object, and $R_{x,y}$ is the reflectance property of the surface.

Referring to the model by Guo *et al*. [62], an extended version of the reflectance model can be expressed as:

$$I_{x,y} = \{d_{x,y} L^d_{x,y} \cos(\theta_{x,y}) + L^e_{x,y}\} R_{x,y}, \tag{4.2}$$

where $L^d_{x,y}$ and $L^e_{x,y}$ are representing the amount of the direct light and the environment light, respectively. $R_{x,y}$ is the surface reflectance. $\theta_{x,y}$ is an angle between the direct lighting direction and the surface normal. $d_{x,y}$ is a value between 0 and 1 indicating how much direct light gets to the surface (albedo). To simplify (2), they define $k_{x,y} = d_{x,y}\cos(\theta_{x,y})$ and get a simplified model :

$$I_{x,y} = \{k_{x,y} L^d_{x,y} + L^e_{x,y}\} R_{x,y}. \tag{4.3}$$

where $k_{x,y}$ is henceforth referred to as the shadow coefficient. Since [62] only focused on shadow and shadow-free pixels, they used $0 \leq k_{x,y} \leq 1$. From (3), a shadow-free pixel can be represented by setting $k_{x,y} = 1$ (i.e. direct lighting):

$$I_{x,y} = \{L^d_{x,y} + L^e_{x,y}\} R_{x,y}, \tag{4.4}$$

and a shadowed pixel when $k_{x,y} < 1$, as follows

$$I^*_{x,y} = \{k_{x,y}L^d_{x,y} + L^e_{x,y}\}R_{x,y}. \tag{4.5}$$

Based on (4) and (5), a new relationship between shadowed and shadow-free pixels can be obtained as:

$$\begin{aligned} I_{x,y} &= \{L^d_{x,y} + L^e_{x,y}\}R_{x,y}\frac{\{k_{x,y}L^d_{x,y}+L^e_{x,y}\}}{\{k_{x,y}L^d_{x,y}+L^e_{x,y}\}} \\ &= \frac{\{L^d_{x,y}+L^e_{x,y}\}}{\{k_{x,y}L^d_{x,y}+L^e_{x,y}\}}I^*_{x,y}. \end{aligned} \tag{4.6}$$

We can see that a shadow-free pixel, $I_{x,y}$, can be generated from a shadowed pixel, $I^*_{x,y}$, by a specific ratio. If we use $r_{x,y} = L^d_{x,y}/L^e_{x,y}$ to simplify (6), then we get:

$$I_{x,y} = \frac{\{r_{x,y}+1\}}{\{k_{x,y}r_{x,y}+1\}}I^*_{x,y}, \tag{4.7}$$

where $r_{x,y}$ is called as an illumination ratio of pixel (x,y).

4.2.2 Illumination Invariant Model for Human Face

The model in [62] can also be used in our case since the illumination problem has similar effects, i.e. shadow and reflection. However, estimating $k_{x,y}$ by a physics-based approach is very hard, because measuring the actual values of $d_{x,y}$ and $\theta_{x,y}$ are very difficult. Instead, we utilize information provided by the image. Firstly we attempt to classify the appearance of face regions into normal and abnormal conditions. Normal condition ($k_{x,y} = 1$) is an actual appearance of a human face in normal lighting. Abnormal conditions ($k_{x,y} \neq 1$) correspond to an appearance which is largely affected by the condition of direct lighting, i.e., shadow ($k_{x,y} < 1$) and reflection ($k_{x,y} > 1$).

Equation (7) is valid only for certain conditions such as the presence of shadow ($k_{x,y} < 1$) and shadow-free ($k_{x,y} = 1$); the presence of reflection effect ($k_{x,y} > 1$) is not taken into account. Although the shadow effect is a more crucial problem than the reflection effect, these effects also potentially produce miss-recognition. Therefore, we consider all potential effects by dividing the range of human face intensities and assigning their properties as shown in Fig. 4.2.

Considering that the most of human face is composed of skin, we segment human skin

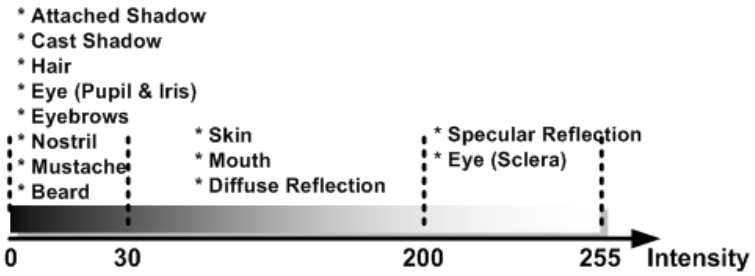


Figure 4.2: The human face color/intensity segmentation in a normal illumination.

intensity based on [64, 65, 66] where the human skin intensities are located between 30 and 200 (when maximum intensity is 255). To obtain the conversion of image that is invariant and to keep the final color constancy, we use the maximum intensity of human skin and make a rough approximation as follows:

$$\hat{k}_{x,y} \approx \frac{I^*_{x,y}}{200}, \tag{4.8}$$

where $\hat{k}_{x,y}$ is an approximated shadow coefficient at position (x,y) that shows a better normalized intensity of the human face. We then make a minor modification to the equation (7), that is,

$$I_{x,y} = \frac{\{r_{x,y}+1\}}{\{\hat{k}_{x,y}r_{x,y}+1\}} I^*_{x,y}, \tag{4.9}$$

where we use a new shadow coefficient $\hat{k}_{x,y}$ instead of $k_{x,y}$.

4.3 Fuzzy Inference System and Its Optimization

We now have an illumination invariant model for human faces. The method of obtaining an illumination ratio, $r_{x,y}$, used in [62] cannot be used in our case because their approach is used exclusively for a partially shadowed area. In our case, the lighting may cause a full shadow effect on the face. It is difficult to always get a pair of shadowed and shadow-free regions from a given image.

Usually, we can estimate and reconstruct an appearance by comparing the current appearance with the intrinsic appearance which is commonly known in advance. For example, in the case of the pupil (see Fig. 4.3(a)), the intensity of a pupil (marked by yellow cross) is usually very low (VL) in the image regardless of shadow effects. We can thus develop reasoning as follows: if the intrinsic appearance is very low (VL) and the current appearance is very low (VL), then $r_{x,y}$ should be very

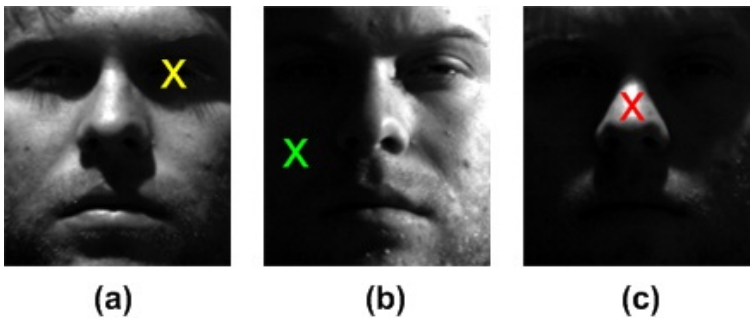


Figure 4.3: The differences of face's appearance due to direct lighting: (a) the pupil's appearance which is always dark even it is covered by shadow or not, (b) the cheek's appearance which is darker because of shadow effect, and (c) the nose's appearance which is brighter because of specular effect.

low (VL). Another example, in the case of the cheek (See Fig. 4.3(b)), its intensity (marked by green cross) is usually medium (M, in the range of normal skin intensity in Fig. 4.2). We thus make the following reasoning; if the intrinsic appearance is medium (M) and the current appearance is very low (VL), $r_{x,y}$ should be high (H).

To get the original intrinsic appearance is, however, difficult. Therefore, we apply a local contrast adjustment to the input image and make a rough approximation of the intrinsic appearance image. We use the Block Histogram Equalization (BHE) [57] to see if a pixel is inherently dark or is inherently bright but appears dark due to a shadow effect. An input image is divided into blocks. A masking block with size 12x12 pixels is applied to the image and histogram equalization is performed. The block is moved every 6 pixels to overlap with an adjacent block to avoid a blocking effect on the adjacent blocks. Then, we apply a weighted sum of neighboring adjacent blocks to smooth the boundaries. Fig. 4.4 shows the illustration of BHE process. The BHE'd result (I^{int}) is shown in Fig. 4.5. At the same time, we use the input image as the current appearance image (I^{cur}).

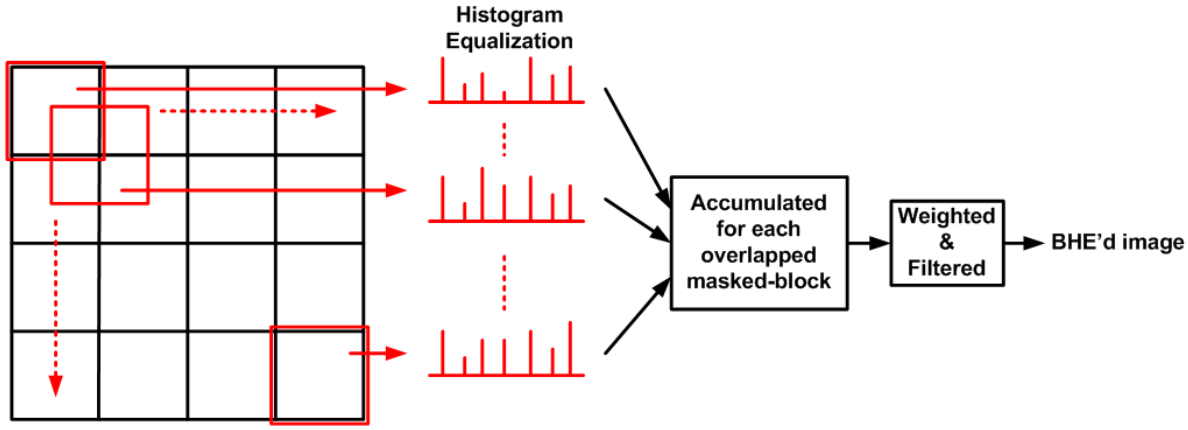


Figure 4.4: Illustration of Block Histogram Equalization.

Figure 4.5: Block Histogram Equalization; Original image (left), BHE'd image (right).

4.3.1 Fuzzy Inference System

For now, we have two pieces of information from the intrinsic appearance image, I^{int}, and the current appearance image, I^{cur}. We can make a simple reasoning from both images. We realize our idea by using a Fuzzy Inference System (FIS) [53] as proposed in [67] because FIS has the capability to combine two types of information based on a determined rule. We argue that the human face has a similar shape and contour; the same lighting direction may produce the same effect for every person. Therefore, it is reasonable to use a simple rule to enhance the affected face region. FIS can be decomposed into three main parts: Fuzzification and membership function, the inference engine and rule base, and defuzzification.

4.3.1.1 Fuzzification and Membership Function

The fuzzification converts a crisp input into a linguistic variable using the membership function. Linguistic variables are labels or concepts corresponding to partitions of a state space, such as low, medium, and high. During reasoning, the variables are referred to by the linguistic terms, and the fuzzy membership function determines the correspondence between linguistic variables with the numerical values. In the case of using five members, for example, we set five linguistic variables, very low (VL), low (L), medium (M), high (H), and very high (VH), for both intrinsic and current appearance, as shown in Fig. 4.6.

We set our Fuzzy membership using a triangular function. The reason for using the triangular function is it outperformed other functions (please refer to Section 5.1 for more detail). Based on the triangular shape, we can derive a degree of the membership function, that is,

$$\mu = \begin{cases} \frac{x-a}{b-a} & \text{if } a < x \leq b \\ \frac{c-x}{c-b} & \text{if } b < x \leq c \\ 0 & \text{if others} \end{cases} \tag{4.10}$$

where μ is a degree of membership function, x is the crisp input, a is the left boundary of membership function, b is the center of membership function and c is the right boundary of membership function. The parameters of the membership function of each input can be expressed as

$$b_m = (m-1)\frac{255}{M-1}, a_m = b_{m-1}, c_m = b_{m+1} \tag{4.11}$$

where $m = 1, 2, ..., M$ is the index of membership. M is the number of Fuzzy membership functions used in our system. M can be 3, 5, 7 or 9. Fig. 4.6 shows the shape of the membership function.

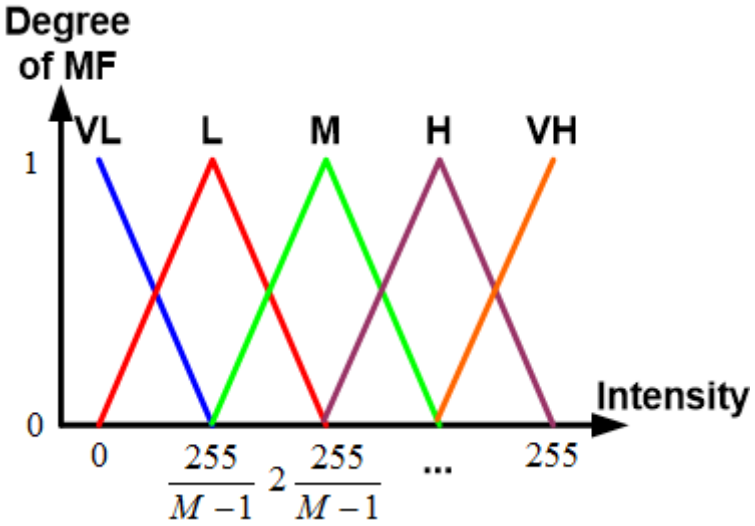


Figure 4.6: Fuzzy membership functions using triangular shape.

4.3.1.2 Inference Engine and Rule Base

We have two kinds of memberships, $\mu_{I_m^{cur}}$ and $\mu_{I_m^{int}}$ to represent the degree of the membership function of the current appearance and the intrinsic appearance images, respectively. We used the product of $\mu_{I_m^{cur}}$ and $\mu_{I_m^{int}}$ instead of the min-operator to form the rule as follows

$$\mathbf{B} = [\mu_{I_m^{cur}} \mu_{I_m^{int}}]_{m=1,2,...,M}, \tag{4.12}$$

where $\mathbf{B}$ is the T-norms between $\mu_{I_m^{cur}}$ and $\mu_{I_m^{int}}$. There are M-memberships for each of the current and the intrinsic appearances and, therefore, we have $F = M^2$ dimensions in total, for which a Fuzzy rule $\mathbf{R}$ needs to be defined.

4.3.1.3 Defuzzification

Finally, defuzzification is performed by converting the fuzzy output of the inference engine to crisp by adopting Center of Area (CoA) that is

$$r_{x,y} = \frac{\sum_{f=1}^{F} B_f R_f}{\sum_{f=1}^{F} B_f}, \tag{4.13}$$

where $r_{x,y}$ is the illumination ratio and $f = 1, 2, ..., F$ is the number of inference $\mathbf{B}$ as well as the number of $\mathbf{R}$.

4.3.2 Optimizing Rules of Fuzzy Inference System

The most important part of the Fuzzy Inference System is a rule base. Manual selection of the rule is very difficult due to the vast variation of inputs, consideration of the image size and range

Algorithm 1 Genetic Algorithm for optimizing Fuzzy's rule

Input: A set of training image, I_n, $n = \{1,2,...,N\}$
Number of iteration, $Iter$
Number of population, H
Number of Fuzzy's rule element, F
Output: Optimized-Fuzzy's rule
Initialization: Randomize H sets of Fuzzy's rule vector, $\mathbf{R}_h = \{R_1, R_2, ..., R_F\}$, $h = \{1,2,...,H\}$
Begin

1. **for** i:=1 **to** $Iter$ **step** 1 **do**
2. **for** h:=1 **to** H **step** 1 **do**
3. **for** n:=1 **to** N **step** 1 **do**
4. Convert I_n using equation (4.9) for $\mathbf{R}_h$
5. **end for**
6. Choose the conversion result of I_1 as a target, and calculate the distances between I_1 and $I_{2:N}$ using equation (4.14) to obtain the fitness value of $\mathbf{R}_h$, $f(\mathbf{R}_h)$
7. **end for**
8. Evaluation: sorting the fitness value, $f(\mathbf{R}_h)$ from the highest to the lowest
9. Reproduction: $\mathbf{R}_1 \leftarrow \mathbf{R}$ with the highest fitness value from the sorting
$\mathbf{R}_2 \leftarrow \mathbf{R}$ which is randomly chosen from the sorting
$\mathbf{R}_1$ and $\mathbf{R}_2$ are set as a new pair of parent
10. Crossover between $\mathbf{R}_1$ and $\mathbf{R}_2$
11. Mutation between $\mathbf{R}_1$ and $\mathbf{R}_2$
12. New composition of population $\mathbf{R}_h$ is formed. Two new individuals as the result of crossover and mutation operations replace the worst two individuals in the current population.
13. **end for**
14. The most optimized $\mathbf{R}$ is obtained after $Iter$ step

of intensities. Optimizing the FIS rule base is, therefore, crucial. We use the Genetic Algorithm (GA) to produce an optimal rule that satisfies all illumination conditions. The Fuzzy optimization using GA is shown in Algorithm 1.

We randomize H-sets of the F-dimensional FIS rule, $\mathbf{R}_h$ (h=1,2,..,H) to initialize the population. Each vector $\mathbf{R}_h$ represents a set of the gene to represent a solution to the optimization problem being addressed and called as an individual. A fitness function associated with each individual is used to evaluate the appropriateness of the solution. We use an averaged-normalized cross-correlation based distance to compute the fitness value, $f(\mathbf{R}_h)$, as follows

$$f(\mathbf{R}_h) = \frac{1}{N-1} \sum_{n=2}^{N} \sum_{x,y} \frac{(I^1_{x,y} - \mu_{I^1})(I^n_{x,y} - \mu_{I^n})}{\sigma_{I^1} \sigma_{I^n}}, \tag{4.14}$$

where $h = 1, 2, ..., H$ is the index of individual, $n = 2, 3, ..., N$ is the index of training image excluding the first image as the normal reference; (x,y) is a pixel position; $I^1_{x,y}$ and $I^n_{x,y}$ are the first and the n-th normalized image, respectively. μ_{I^1} and μ_{I^n} are the mean of the first and the n-th normalized image, respectively. σ_{I^1} and σ_{I^n} is the standard deviation of the first and the n-th normalized image, respectively.

The aim of applying the genetic operators is to transform an individual into a new individual with a higher fitness value. The reproduction operator performs a natural selection function to choose the individuals with the highest fitness values that have a greater probability to be selected as a pair of parents to produce better offspring. The crossover operator chooses a pair of genes by randomly selecting a point and exchanging their tails. The mutation operator randomly mutates the values of the same gene position to optimize each individual. The optimization process is iterated until required conditions are met (e.g., determined number of generation or fitness value threshold).

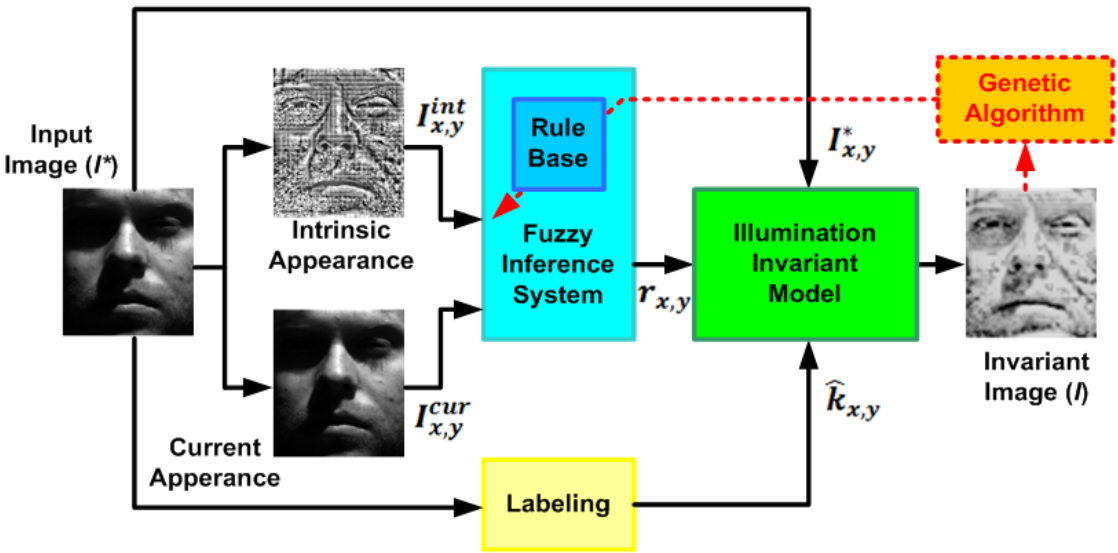


Figure 4.7: Our proposed system for the illumination invariant face recognition.

Fig. 4.7 shows the block diagram of our proposed method. The complete diagram shows the optimization process, where an input image is converted to an invariant image using our illumination invariant model. The GA-based optimized-rule ensures that the Fuzzy Inference System produces an appropriate illumination ratio for a better invariant image. When the optimization process has finished, the Genetic Algorithm (red dashed) is released. Our system is then ready to directly enhance the input image.

4.4 Online Face Recognition System

To deal with an online system, we designed our system as shown in Fig. 4.8. First, we convert the input image using our illumination invariant method. The converted image is then fed into the face detector process to search for and detect human faces in the image. The detected face is then aligned using the face alignment process to get a better face position and orientation. Finally, the aligned face is recognized using the mutual subspace method to determine the person's identity.

4.4.1 Face detection

The Viola-Jones face detector [47] is applied to the illumination invariant images in order to detect human faces online in real scenes. We trained the detector using 4,000 positive images and 7,000 negative images; where all images were converted to invariant images using the illumination invariant method before training.

4.4.2 Face alignment

A detected face from the face detector needs to be further processed. Face recognition is feeble against rotation and scaling in the image. Aligning the cropped face is, therefore, very

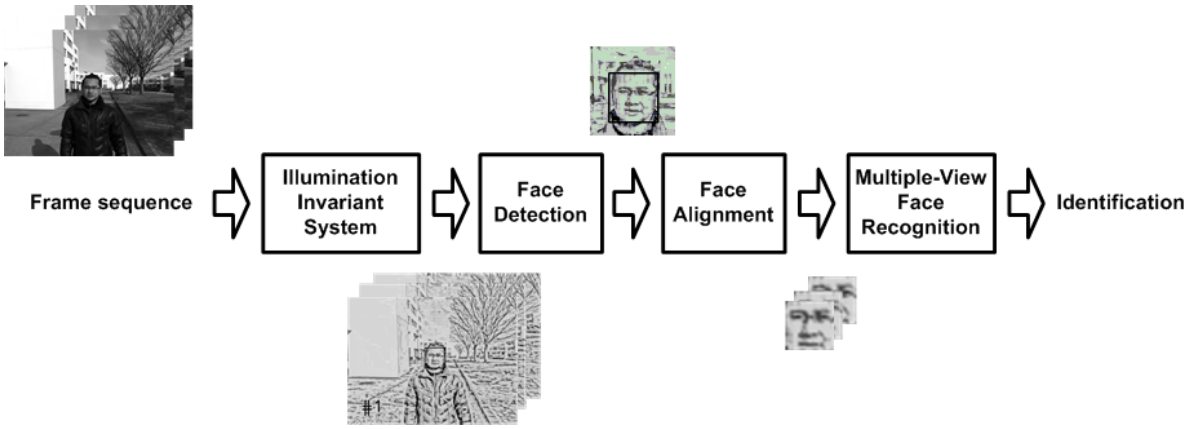


Figure 4.8: Our robust illumination invariant face recognition system.

important to minimize the miss-recognition. We adopt a multiple-scale and multiple-angle template matching based face feature detection [68] to detect eye position, measure the rotation angle, and correct the scale.

We first capture several templates of eyes. The angle of rotation of each template is limited up to ± 45 degrees. Each template is then enhanced using the illumination invariant method. From the detected eye positions, an angle of rotation can be calculated. The face image is then rotated using the calculated angle. Unnecessary parts such as hair, neck and background are then discarded. After the alignment process has completed, the aligned image is resized to 32×32 pixels.

4.4.3 Face recognition

Pattern matching using a single view of the face frequently causes miss-recognition. We employ the Mutual Subspace Method (MSM) [69], a powerful matching technique that tolerates variations in face patterns. It utilizes multiple canonical angles between the input subspace and the reference subspace. The MSM recognizes a temporal aligned face image sequence as shown in

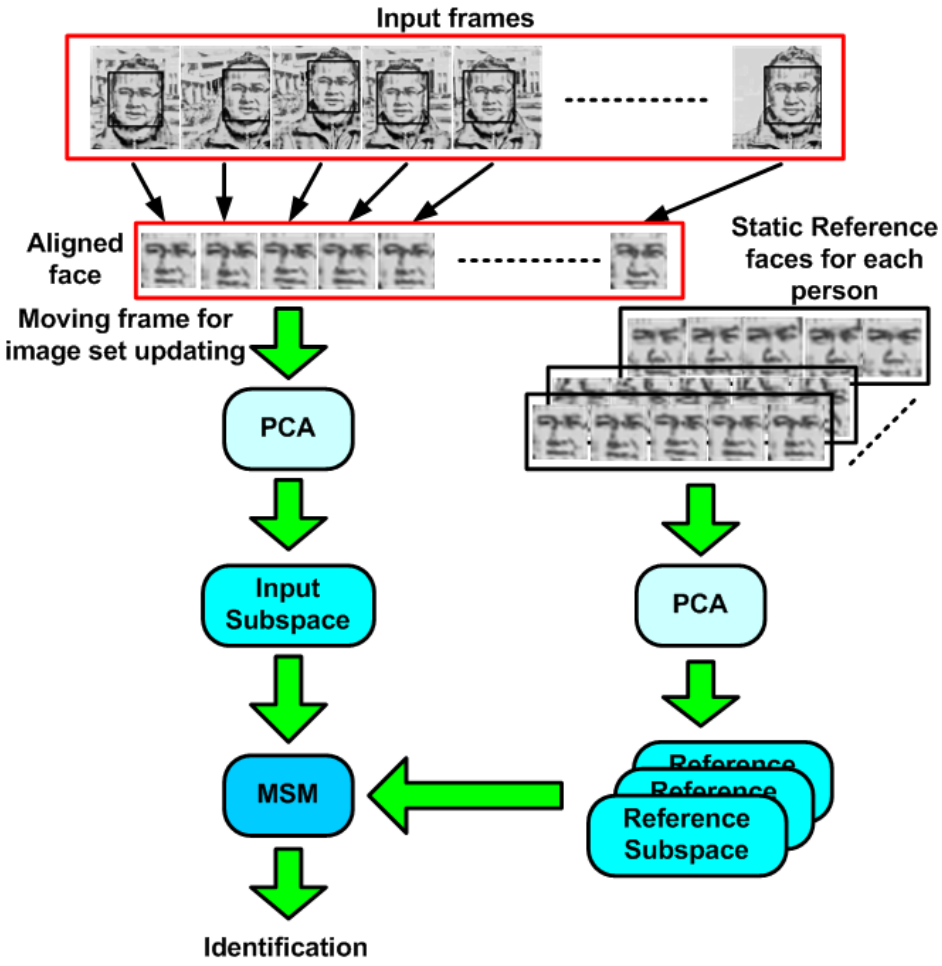


Figure 4.9: Flow of face recognition based on MSM.

Fig 4.9.

4.5 Experimental Results

4.5.1 Selecting the Best Function for Fuzzy Membership

There are two functions that are commonly used in Fuzzy systems, triangular function and Gaussian/Bell function. To ascertain which function is the best, we compared the performance of our method using both functions. We use a set of images of one subject in the Yale B extended face database [70] for optimization. The database is fairly representative because it provides a sufficient variety of illumination conditions. We have tested several sizes of image, namely 168×192 pixels, 84×96 pixels, 42×48 pixels, and 21×24 pixels. We chose a resized image with size of 42×48 pixels because it is the smallest size that can maintain recognition accuracy. Similar appearance images are the objective of the optimization.

We performed the GA training for a 5×5 Fuzzy rule. The reason of using the 5×5 Fuzzy rule is explained in Section 5.2. We iterate and record the fitness value up to 10,000 iterations to guarantee its convergence as shown in Fig. 4.10. Even though the gaussian function can reach higher fitness value faster than the triangular function, but its fitness stops around 0.85, while the triangular function can achieve higher value (around 0.88). These achievements affect to the

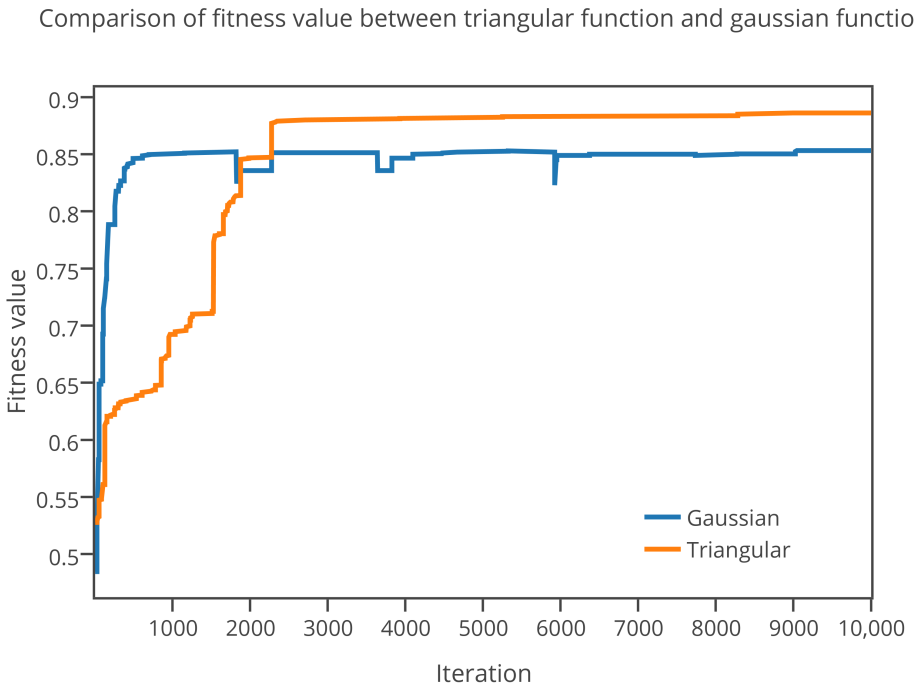


Figure 4.10: Comparison of fitness value evolution between the triangular function and the gaussian function.

Table 4.1: Comparison of face recognition performance using the triangular function and the gaussian function.

Function	Nearest Neighbor	Principal Component Analysis	Time
Triangular	94.24 %	92.85 %	8.41 ms
Gaussian	91.49 %	85.61 %	78.99 ms

recognition results (using 38 subjects of Yale B Extended face database) as shown in Table. 4.1. From this reason, we decided to use the triangular function.

4.5.2 Fuzzy Rule Optimization

After getting the best function for our Fuzzy membership, then we perform the optimization of the Fuzzy rule. We utilize the same set and size of images, iteration number, and objective as used in Section 5.1 for training using different sizes of the Fuzzy rule as shown in Table 4.2. The evolution of fitness value of each size of Fuzzy rule is shown in Fig. 4.11. Based on the training outcomes and referring to the test results using the 10 subjects from Yale B Extended face database, we used the 5×5 rule because it has a rapid transition without experiencing the local optimum problem. It also achieves a high fitness value and produces the best invariant image as shown by Fig. 4.12. The best achievements using NN and Eigenfaces (PCA) show the stability of the 5×5 rule. The 5×5 rule is shown in Fig. 4.13.

Fig. 4.13 shows a mapping of inputs (current appearance and intrinsic appearance) and an output of the fuzzy inference system based reasoning. For example, if the intensity of a particular pixel has "current appearance = VL" and "intrinsic appearance = VL", then "output = 10". It means that our illumination invariant model will receive a small illumination ratio, which gives only a small adjustment to the intensity of that pixel.

Table 4.2: Comparison results of different size of the Fuzzy rule.

Rule	Max Fitness	Evaluation Method		Time (msec)
		NN	PCA	
3×3	0.7956	97.66 %	84.38 %	8.02
5×5	0.8863	99.84 %	98.75 %	8.41
7×7	0.8912	93.28 %	74.69 %	10.04
9×9	0.8933	99.69 %	89.06 %	13.56

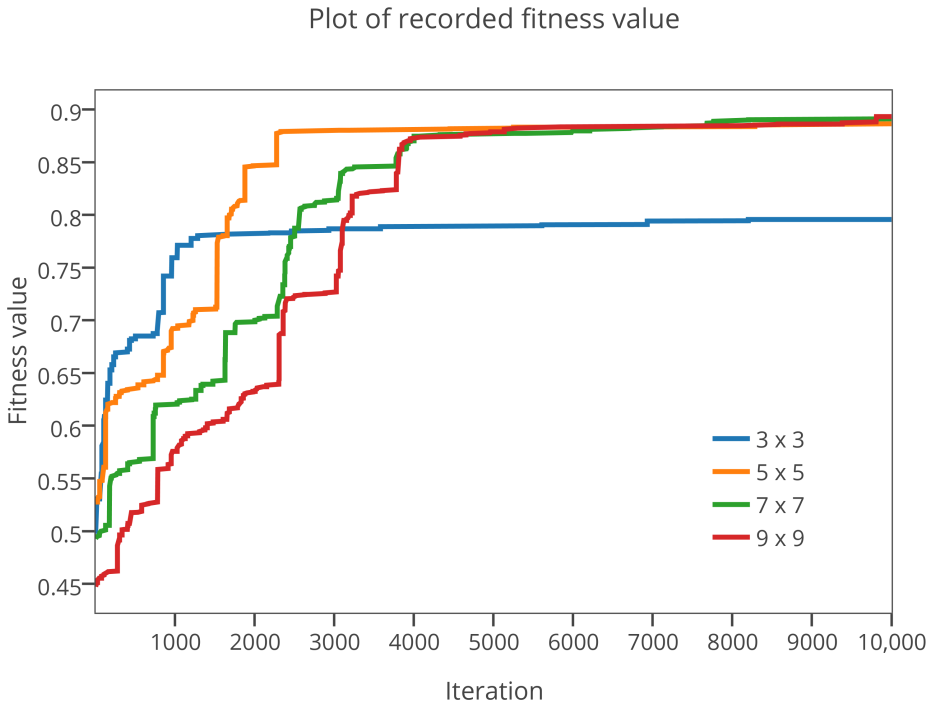


Figure 4.11: Evolution of GA's fitness value for different size of the Fuzzy rule for 10,000 iterations.

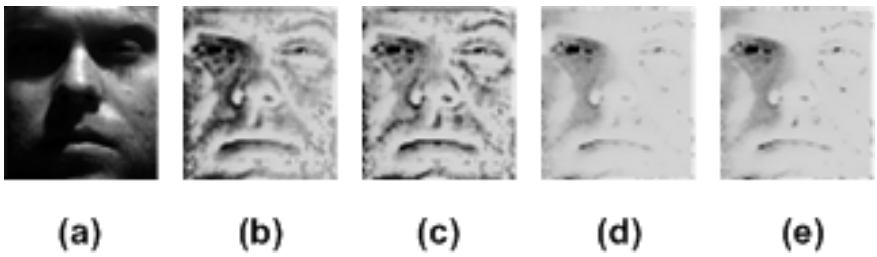


Figure 4.12: Comparison results of different size of the FIS rule; (a) The original image, (b) The converted image using a 3×3 rule, (c) The converted image using a 5×5 rule, (d) The converted image using a 7×7 rule, and (e) The converted image using a 9×9 rule.

4.5.3 Experiment Using Face Database

Our experiments use two public face databases to measure and compare the quality of all methods using our best-defined parameters as shown in Table 4.3. Yale B Extended [70] and CAS-PEAL [71] face databases are the representative face databases that deal with the illumination problem. We use Nearest Neighbor (NN), Eigenfaces (PCA) [72] and Support Vector Machines (SVMs) [55] for evaluation. The classifier settings are summarized in Table 4.4.

5 x 5		Current Appearance				
		VL	L	M	H	VH
Intrinsic Appearance	VL	10	1	0	0	20
	L	2	0	0	0	2
	M	57	7	2	0	14
	H	142	137	97	57	0
	VH	145	118	148	145	31

Figure 4.13: A 5×5 Fuzzy rule generated by GA.

Table 4.3: Parameters setting of proposed method and other methods in comparison

Method	Parameters	Value
None	-	-
HE	-	-
BHE [57]	block size	12
MSR [30]	filter size (Gaussian)	3, 5, 7
FR [31]	filter size (Gaussian)	3
	Fuzzy membership function	Triangular, 5×5
ASR [34]	Smoothing iteration, T	50
SQI [32]	filter size (Gaussian)	3, 5, 7
CAQI [33]	filter size (Gaussian)	3, 5, 7
	α	0.05
Opt-SQI [35]	filter size (Gaussian)	3, 5, 7, 9, 11, 13, 15, 17
	β_k	1, 1/15, 2/15, 7/15, 1, 1, 1, 3/15
	m_k	4/15, 1/15, 1/3, 0, 4/15, 14/15, 14/15, 3/5
	α	0.98
DCT [36]	Ddis (discarded DCT coefficient)	15
BDIP-BVLC [39]	block size	3
GF [58]	-	-
OptiFuzz(Ours)	block size (Appearance model)	12
	Fuzzy membership function	Triangular, 5×5
	Fuzzy rule	See Fig. 4.13

4.5.3.1 Experimental Results Using Yale B Extended Face Database

The Yale B extended face database consists of 38 subjects with 64 illumination variations. Each image is already cropped and aligned at size of 168×192 pixels. 38 frontally illuminated face images are chosen as a training set. While the remaining are used as a testing set.

We re-implement the existing appearance-based illumination invariant methods as listed in Table 4.3. The illumination invariant results of each method are shown in Fig. 4.14. We can see

Table 4.4: Parameters setting of the classifier methods

Classifier	Parameters	Method/Value
NN	Classification type	Pearson Product Moment Correlation Coefficient
PCA [72]	Number of principal components	38
SVMs [55]	SVM type	nu-SVC multi-class classification
	Kernel type	Linear

Table 4.5: Recognition rate of appearance based illumination invariant method for Yale B Extended face database (%)

		Recognition Rate (%)		
No	**Methods**	**NN**	**PCA**	**SVMs**
1	Raw	41.98	40.46	32.81
2	HE	46.83	53.95	47.12
3	BHE [57]	90.25	86.76	87.34
4	MSR [30]	83.80	66.82	12.21
5	FR [31]	65.17	59.42	12.91
6	ASR [34]	92.97	90.38	81.58
7	SQI [32]	87.54	86.72	88.90
8	CAQI [33]	53.50	60.57	7.77
9	Opt-SQI [35]	91.90	89.93	92.02
10	DCT [36]	91.94	86.76	30.88
11	BDIP+BVLC [39]	93.96	85.07	82.81
12	GF [58]	56.78	54.15	52.38
13	SVD Face [37]*	93.50	N/A	N/A
14	ME [38]*	N/A	92.00	N/A
15	OptiFuzz(Ours)	**94.24**	**92.85**	**94.78**

Table 4.6: Recognition rate of feature based illumination invariant method for Yale B Extended face database (%)

		Recognition Rate (%)		
No	**Methods**	**NN**	**PCA**	**SVMs**
1	LBPH [60]	48.91	49.06	48.44
2	WLD [59]	50.63	60.16	31.41
3	LGH [61]*	93.30	N/A	N/A

that our method and ASR are able to maintain the final color constancy and minimize the boundary effect. The quantitative results are shown in Table 4.5. We also include the results of the other new methods such as the SVD-Face and the Mean Estimation. The (*) mark indicates that the result are directly taken from the paper.

The existing appearance based illumination invariant methods such as HE, MSR [30], FR [31], CAQI [33], and GF [58] are not robust enough when tested using very harsh illumination

conditions. They fail to control the final color constancy of the converted images. The remaining methods such as BHE [57], ASR [34], SQI [32], Opt-SQI [35], DCT [36], and BDIP-BVLC [39] can maintain the final color constancy of most of the converted images, however, some of them fail to reduce boundary effects. Considerable differences in the performance of each classifier show that these methods do not have consistent stability.

SVD-Face [37] and ME [38] seem to perform well regarding the performance. However, we do not know the consistency of their stability since we do not have data for the other classifiers. However, one can note that our proposed method has the best performance for face recognition and the best stability in the experiments. It successfully maintains the final color constancy, reduce the boundary effects and provide useful facial textures that are very helpful to maintain a high recognition rate.

We also compare the performance of our proposed method with the feature-based methods as shown in Table 4.6. Local Binary Pattern Histogram (LBPH) [60], Weber Local Descriptor (WLD) [59] and Logarithm Gradient Histogram (LGH) [61] are the examples of feature based illumination invariant method. The (*) mark indicates that the result are directly taken from the paper. Our method also still shows better results for all the classifiers.

4.5.3.2 Experimental Results Using CAS-PEAL Face Database

The CAS-PEAL face database is a huge Chinese face database that consists of pose, expression, accessories and lighting datasets in one package. We use only the lighting dataset which consists of 1,042 subjects with 20 illumination variations. We chose the first 20 subjects and captured the whole area of face using the eye position ground truth information that is included in the distributed dataset. We rotate, align, crop and resize each detected face at 168×192 pixels.

We used four images (one image of frontally illuminated face, one image under fluorescent lighting, one image under incandescent lighting, and one image is randomly chosen under an elevation angle of $\pm 45^o$) of each subject as the training set. The remaining images are used as the testing set.

We use the same re-implementation as we have done for Yale B Extended face database when evaluating the CAS-PEAL face database. The illumination invariant results are shown in Fig. 4.15. The quantitative results are shown in Table 4.7. We also include the result of the Mean Estimation method. The (*) mark indicates that the result is directly taken from the paper and the (**) mark indicates that each method is evaluated using only one frontally illuminated face image of each subject as the training image and the remaining used for testing. Our proposed method successfully outperforms the other existing and the state-of-the-art appearance-based illumination invariant methods.

We also compare the performance of our proposed method with the feature-based methods as shown in Table 4.8. Local Binary Pattern Histogram (LBPH) [60] and Weber Local Descriptor

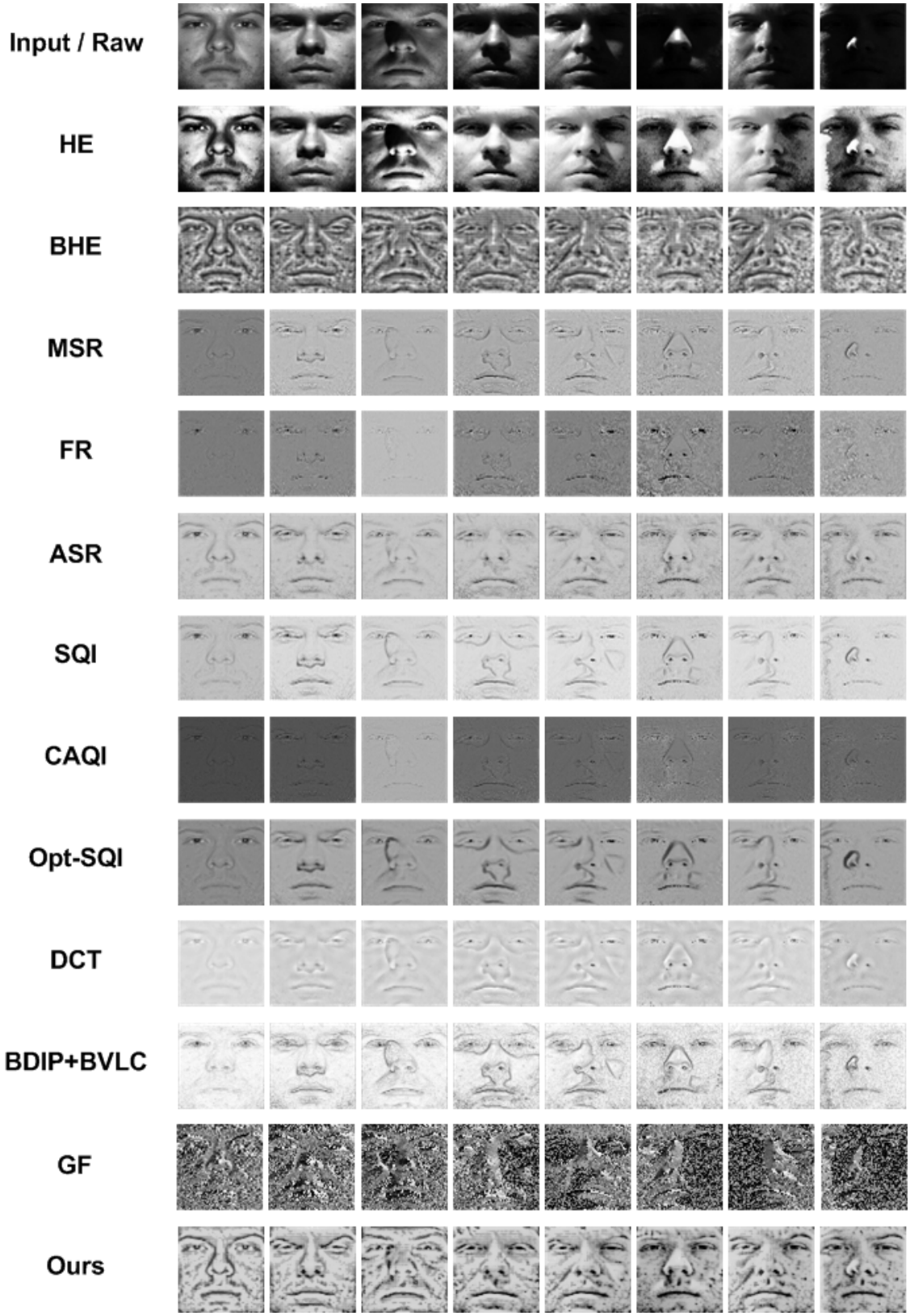


Figure 4.14: Comparison of illumination normalization methods for Yale B Extended face database.

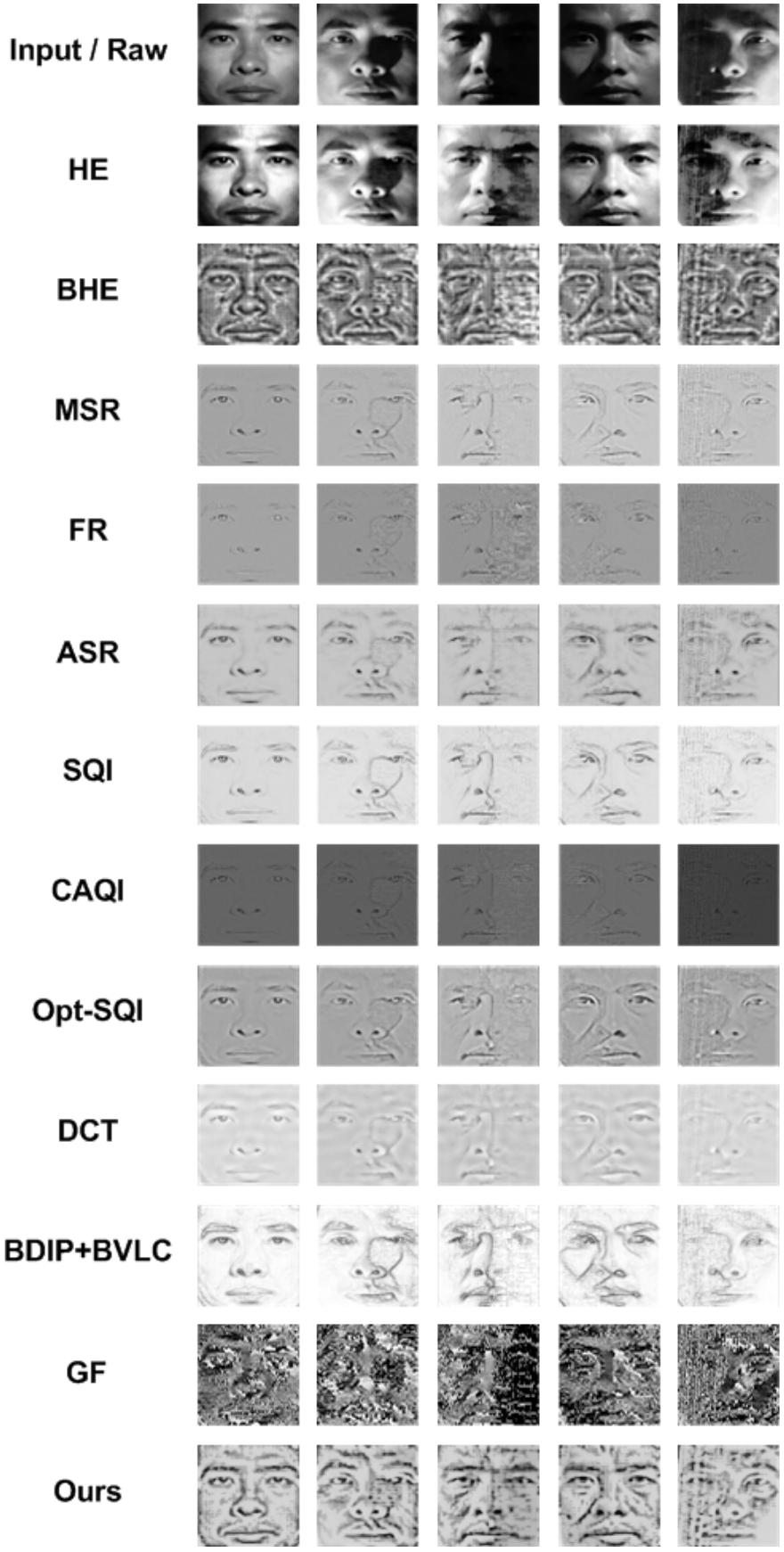


Figure 4.15: Comparison of illumination normalization methods for CAS-PEAL face database.

Table 4.7: Recognition rate of appearance based illumination invariant method for CAS-PEAL face database (%)

No	Methods	Recognition Rate (%)		
		NN	**PCA**	**SVMs**
1	Raw	30.00	44.38	50.63
2	HE	33.13	53.44	54.06
3	BHE [57]	**79.06**	76.25	75.63
4	MSR [30]	60.00	52.19	60.31
5	FR [31]	38.44	28.44	34.38
6	ASR [34]	71.25	65.94	76.88
7	SQI [32]	70.94	71.25	77.81
8	CAQI [33]	27.50	30.94	39.38
9	Opt-SQI [35]	73.44	75.63	72.50
10	DCT [36]	64.69	56.25	60.00
11	BDIP+BVLC [39]	70.63	62.81	74.38
12	GF [58]	53.75	46.25	47.81
13	ME [38]*	N/A	**23.21	N/A
14	OptiFuzz(Ours)	**79.06**	**79.06**	**80.94**
15	OptiFuzz(Ours)	N/A	****51.25**	N/A

Table 4.8: Recognition rate of feature based illumination invariant method for CAS-PEAL face database (%)

No	Methods	Recognition Rate (%)		
		NN	**PCA**	**SVMs**
1	LBPH [60]	38.13	37.81	40.31
2	WLD [59]	33.13	34.69	28.13

(WLD) [59] represent the feature based illumination invariant method. Our method still shows better results for all classifiers.

4.5.3.3 Computation Time

We evaluate the computation time of all illumination invariant methods for still image databases. It is obtained by averaging the computation time of 2,394 images in Yale B Extended face database and 320 images in CAS-PEAL face database. Evaluation is conducted using Microsoft Visual C++ running on a personal computer system equipped with 2.66 GHz Intel processor supported by 2 GB of RAM. The result is summarized in Table 4.9.

4.5.4 Experiment For Face Detection

In most online face recognition problems, the first step is finding the face itself. Finding the face in a normally illuminated image is not as difficult as finding it in a harshly illuminated

Table 4.9: Computation time of each method

No	Category	Methods	Time(msec)
1	Raw	Raw	0.08
2	Histogram	HE	0.16
3		BHE [57]	2.64
4	Photometric	MSR [30]	7.73
5		FR [31]	4.80
6		ASR [34]	69.21
7		SQI [32]	34.16
8		CAQI [33]	33.88
9		Opt-SQI [35]	214.81
10		DCT [36]	5.94
11		ME [38]	9.00
12		SVD-Face [37]	50.00
13		**OptiFuzz(Ours)**	**8.41**
14	Gradient	BDIP+BVLC [39]	38.84
15		GF [58]	1.52
16	Features	LBPH [60]	3.12
17		WLD [59]	11.63
18		LGH [61]	N/A

image. Use of illumination invariant images improves the detection in harshly illuminated images. We conducted three different experiments to prove the effectiveness of our illumination invariant images.

Figure 4.16: Comparison of the performance of face detection without and with the illumination invariant method. Upper images show the input images. Middle images show the face detection results without illumination invariant method. Lower images show the face detection results with illumination invariant method. The first and the second column, the third and the fourth column, and the fifth and the sixth column show the results using "Yale B Extended" face database, "Real1" video dataset, and "Real2" video dataset, respectively.

Table 4.10: Comparison of the performance of face detection without and with illumination invariant method.

Dataset	No of Person	No of Image	Original				Applying illumination invariant			
			TP+FP	% (TP+FP)	TP	% (TP)	TP+FP	% (TP+FP)	TP	% (TP)
Yale B Ext.	5	325	273	84.0 %	228	70.0 %	324	**99.7 %**	322	**99.0 %**
Real1	3	240	228	95.0 %	176	73.0 %	240	**100.0 %**	197	**82.0 %**
Real2	3	250	216	86.4 %	181	72.0 %	230	**92.0 %**	205	**82.0 %**

We apply Haar-based detector (Viola-Jones face detector) [47] for detecting face in both original and converted images. For detecting original images, the Haar-based detector was trained using Viola-Jones original database. To be used in our system, we trained the Haar-based detector using 4,000 positive images and 7,000 negative images; all images were converted to invariant images using the illumination invariant method before training. In the first experiment, we used the "Yale B Extended" face dataset (5 persons, 65 images each). In the second experiment, we used "Real1" dataset (day, indoor, 3 persons, 80 images in sequential each). In the third experiment, we used "Real2" dataset (night, outdoor, 3 persons in one frame, 250 images of sequential video capture). We set the detection setting to output only one biggest face region among the ones found in the image. The results of each experiment are shown in Fig. 4.16 and Table. 4.10.

True positive (TP) detection means that the detector can precisely detect face areas, while false positive (FP) detection means the detector fails to precisely detect face areas, i.e., shifted, detects other object and the detected face area is too small or too large. We validated the results by manually checking each image produced in each experiment. From the experimental results, we show that using illumination invariant images is effective in increasing the number of correct detected faces which are very helpful for supporting face recognition.

4.5.5 Experiment For Online Face Recognition

4.5.5.1 Experimental Setup

We have examined each illumination invariant method performances to the offline face recognition problems. In this experiment, we evaluate the performance of the best six of the appearance-based illumination invariant methods which were successfully re-implemented by us, i.e. BHE [57], DCT [36], BDIP-BVLC [39], ASR [34], Opt-SQI [35], and ours.

We captured 75 illumination invariant face images of three people to create the reference set in this experiment. 25 images were associated with each person. Each image was re-sized to 32×32 pixels. We then recorded three 640×480 pixels videos with each person for testing. Each person walked following the camera at a distance of about two meters and were conditioned to always look ahead. Each video was captured at our campus in both indoor and outdoor scenes.

The indoor video was taken in a corridor with various lighting conditions. The outdoor video was taken during the daytime. The duration of each video recording was one minute. Each video was captured at 30 frames per second by a web camera.

Since the performance of the MSM depends on the number of the reference images and the subspaces used, the experiment is divided into two parts: First, using only five reference images and five subspaces. Second, using 25 reference images and increasing the number of subspace steps by five. The performance of each illumination invariant method is measured by counting the number of correctly recognized people.

4.5.5.2 Experimental results

The comparison result of the selected illumination invariant methods is shown in Fig 4.17. Our method and ASR equally outperform the others for all combinations in the number of training images and subspace. The results of both methods increase from around 70% to 80% by increasing the number of training images, even while maintaining the same number of subspace. Overall, from the methods in comparison, Opt-SQI and DCT recorded the highest increases in performance. The recognition rate increases when we increase the number of subspace. In contrast, the recognition rate of BDIP-BVLC and BHE show constant and negative trends, respectively. Nevertheless, Opt-SQI and DCT frequently fail to control final appearances in some harsh illumination conditions whereas BHE and BDIP-BVLC are frequently too rough. As a consequence, important parts of the face are not clear and tend to be vague as shown in Fig 4.18.

Fig 4.19 shows the experimental results of three people using our illumination invariant

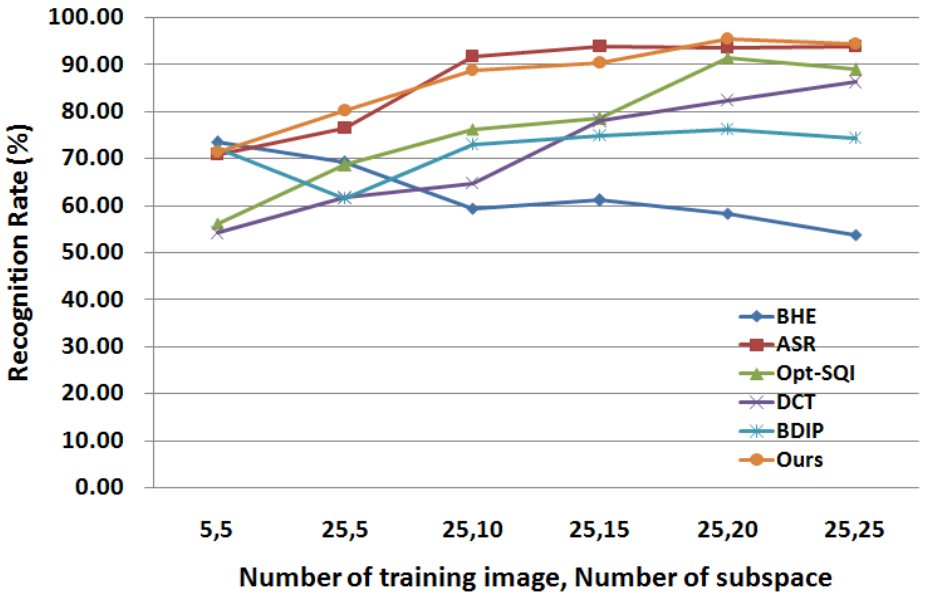


Figure 4.17: Recognition rate correspond to the number of training image and number of subspace.

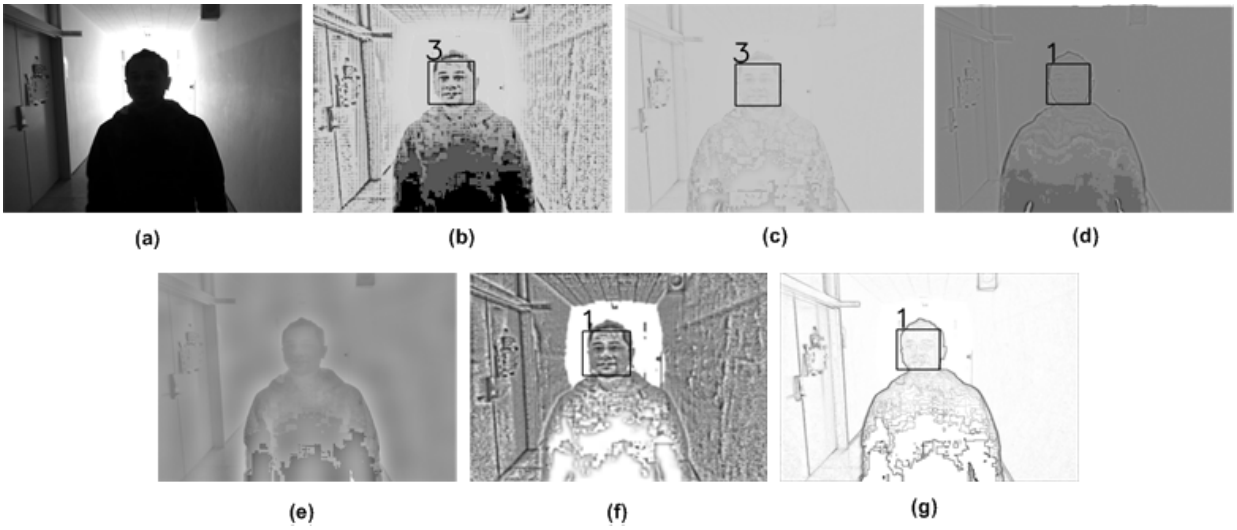


Figure 4.18: A comparison of online illumination invariant methods: (a) The original input image of person #3, (b) Ours, (c) ASR, (d) Opt-SQI, (e) DCT, (f) BHE and (g) BDIP-BVLC. A number on the top-left of the bounding box shows the recognition identity. (b) is the converted image which look quite clear and can be recognized correctly. The face region in (c), (d), (e), and (g) seem to be vague where the face properties can not be seen clearly. Viola-Jones face detector fails in (e), while MSM fails in (d), (f) and (g).

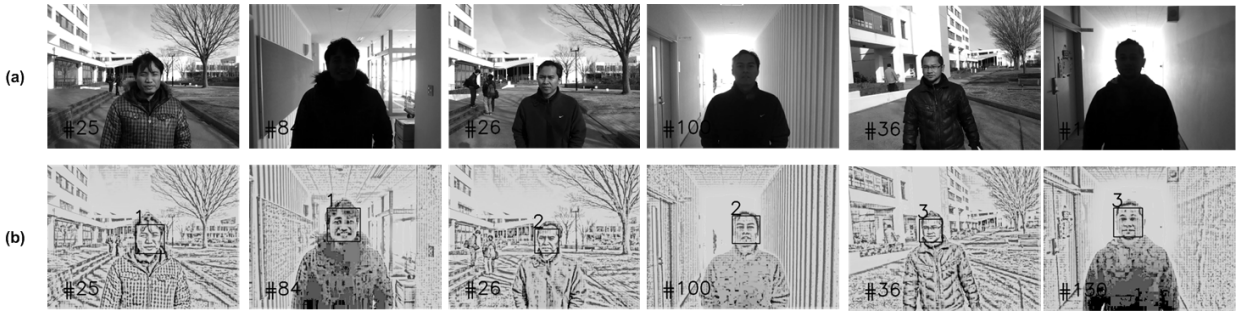


Figure 4.19: Experimental results for online system using our illumination invariant method; (a) the original input frames and (b) the converted frames and its recognition results that are informed on the top-left of the bounding box.

method. Our method can produce a stable invariant image while retaining the clarity of the parts of the face. With these capabilities, the classifier will be easier to recognize the identity of the person.

4.5.5.3 Computation time

The average computation time of the illumination invariant method, the face detection, the face alignment and the multiple-view face recognition processes of each illumination invariant method are presented in Fig 4.20. Although our illumination invariant method is slightly slower than BHE and DCT, it is much faster than BDIP-BVLC, ASR, and Opt-SQI. From Fig. 4.17, even though the performance of ASR is comparable to our method, based on Fig. 4.20, our method is

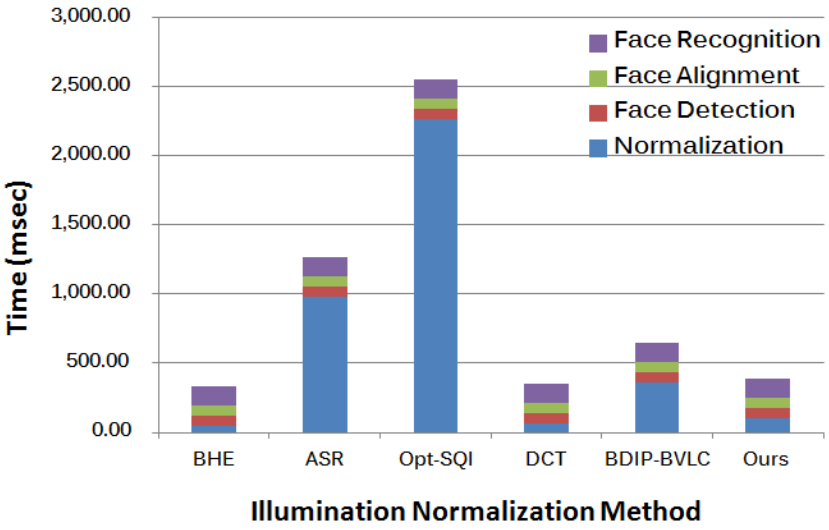


Figure 4.20: Comparison of computation time of each illumination invariant method using 25 training data and 10 subspaces.

3.57 times faster than ASR. This means that our method is more applicable for real implementation ($\approx$ 3 fps).

Chapter 5

Supporting Component: Human Head Orientation Estimation

5.1 Introduction

The human head and face are the most common parts of the human body used in computer vision applications such as detecting the presence of a person, identifying and verifying a person, and indicating one's attention. In order to maintain a good communication or a good interaction, the estimating of head orientation is important. Head orientation can be used to estimate an attentional awareness during an interaction process so that one may expect an appropriate response that is in-line with the degree of attention.

Many methods have been proposed to deal with head orientation estimation [73]. The methods can be categorized into three main groups: (1) facial features based method, (2) model-based method, and (3) appearance-based method. The facial features based methods detect facial components such as the eyes, mouth and nose and calculate the geometrical relationship between them for face orientation estimation. This approach normally requires a high precision facial component detector. However, the detector is usually sensitive to distance and changes in illumination levels. The model-based methods use a priori known 3-D models of a human's head or 2-D models of a face such as the face contour and the facial components that are matched with the unrecognized 3-D head or 2-D face models. This approach usually works well at limited angles only, e.g., pan angle is within $\pm 50^o$. The appearance-based methods assume there is a relationship between the head or face pose and changes in some properties in a 2-D facial image. This approach accommodates wider pan, tilt and roll angles in a plane. However, inaccuracy is the main drawback of this approach.

Some applications such as car driver awareness detection, human-robot interaction, and video surveillance involve a variety of view-angle ranges, from a tight range up to a full 360^o view. In human-robot interaction, the range of angle $\pm 90^o$ is commonly used. Therefore, we focus our work on the appearance-based approach by proposing a novel descriptor that combines more features such as Weber, gradients and intensity deviation. This descriptor is named the Covariance of Weber-Gradient-Deviation Descriptor (CWGDD). We show that integrating more features with different capabilities will make our descriptor more robust in order to discriminate various head orientations.

5.1.1 Related works

Many prior works on head pose estimation have been surveyed in [73]. In general, most of them used limited feature variations such as intensity and edge information.

Han *et al*. [41] proposed the Image Abstraction and Local Directional Quaternary Pattern (IA-LDQP). An edge-like image is extracted from a precisely segmented-image using a Difference of Gaussian (DoG) filter. A set of edges is then extracted into a histogram by following the LDQP technique. However, distribution of the histograms are frequently similar among different poses due to binary intensity. This issue potentially reduces accuracy.

The Gabor-Filter is frequently used for edge detection. The multiple-scale and multiple-orientation edge features that are extracted from an image using the Gabor-Filter have been proposed in the Covariance of Gabor Filter (CovGa) [42]. This feature is then combined with other information such as pixel positions and intensities using a covariance matrix. This method successfully achieves a good accuracy. However, this approach consumes excessive time due to large number of scales and orientations used.

One notable work is the Covariance of Oriented Gradient (COG) by Dong *et al*. [43]. The idea of using a covariance matrix for the head pose classification problem was first proposed in this paper. They argued that a head pose can be assumed as changes in the edges information of a face image. This assumption can be realized in a low-resolution image extracted by a gradient approach. This descriptor is mainly composed of a gradient image and orientations of gradient combined with other information such as pixel positions and its intrinsic intensity when using a covariance matrix. The experimental results showed that this method achieves an excellent result for classifying head poses. Referring to the achievements of the last two methods [43, 42], a covariance matrix seems very promising and suitable to combine and to compact multiple features.

CovGA and COG are shown to be effective for human head pose estimation. However, we think that combining the advantage of gradient features in COG with other feature that is effective for analyzing different cue may improve the performance. Weber Local Descriptor (WLD) [74] is a robust descriptor for texture analysis. This descriptor works well for discriminating and classifying textures (Brodatz and KTH-TIPS2 texture databases) by considering two pieces of information at once; (1) differential excitation and (2) orientation of the gradient. The differential excitation elegantly detects the edges and is robust to the illumination change while the orientation of gradient is very powerful for characterizing a directional change in the intensity or color in an image. This descriptor is never used for head orientation estimation. Because of these advantages, we have adopted into our system.

There are two main contributions in this paper. First, we combine different types of features such as pixel position, intrinsic intensity, Weber, gradients and intensity deviations to create a more robust descriptor. The engagement of the last three features is proven to be effective for discriminating each pose/orientation. Secondly, we perform a deeper analysis to measure

the degree of importance of each image's block. The degree of importance is determined by a weight. The application of appropriate weights can provide significant improvement in the results. Determination of the appropriate and optimal weights are our greatest contribution. The use of a Genetic Algorithm (GA) [45] to determine the values of the weights is a further additional contributions.

5.2 Building a descriptor for estimating head orientation

5.2.1 Weber-based feature

The head orientation variations can be perceived as the change of image pattern. Characterizing the image patterns can be performed in most texture domains, e.g.: SIFT [75], LBP [76] and WLD [74]. WLD-based feature has been proven to perform well for texture analysis. The changes of head orientation are closely related to the changes of texture.

5.2.1.1 Generating features

The Weber-based feature [74] is mainly composed of two parts (see Fig. 5.1); (1) The differential excitation and (2) the orientation of the gradient. The differential excitation measures the intensity differences between a current pixel with its neighbors to find the salient variations within an image that is expressed as

$$\begin{aligned} v_1 &= \textstyle\sum_{i=0}^{p-1}(\Delta I_i) = \sum_{i=0}^{p-1}(I_i - I_c), \\ v_2 &= I_c, \end{aligned} \tag{5.1}$$

where I_c is the center pixel, I_i $(i = 0, 1, ..., p-1)$ denotes the $i-th$ neighbors of I_c and p is the number of neighbors. The differential excitation ξ can be expressed as:

$$\xi = \tan^{-1}\left(\frac{v_1}{v_2}\right) = \tan^{-1}\left(\sum_{i=0}^{p-1}\left(\frac{I_i - I_c}{I_c}\right)\right). \tag{5.2}$$

The orientation of the gradient shows a direction of pixels difference in vertical, $v_4 = x_7 - x_1$, and horizontal, $v_3 = x_5 - x_3$. x_1, x_3, x_5 and x_7 are the kernel filters shown in Fig. 5.1. The orientation of the gradient ϕ is expressed as:

$$\phi = \tan^{-1}\left(\frac{v_4}{v_3}\right) = \tan^{-1}\left(\frac{x_7 - x_1}{x_5 - x_3}\right). \tag{5.3}$$

The Arctangent function is applied since it can limit the output to prevent it from increasing or decreasing too quickly when the input becomes larger or smaller. For simplicity and to avoid a negative-value-effect to the *log* operator, both features are then quantized into positive values between 0^o-90^o using the following expression:

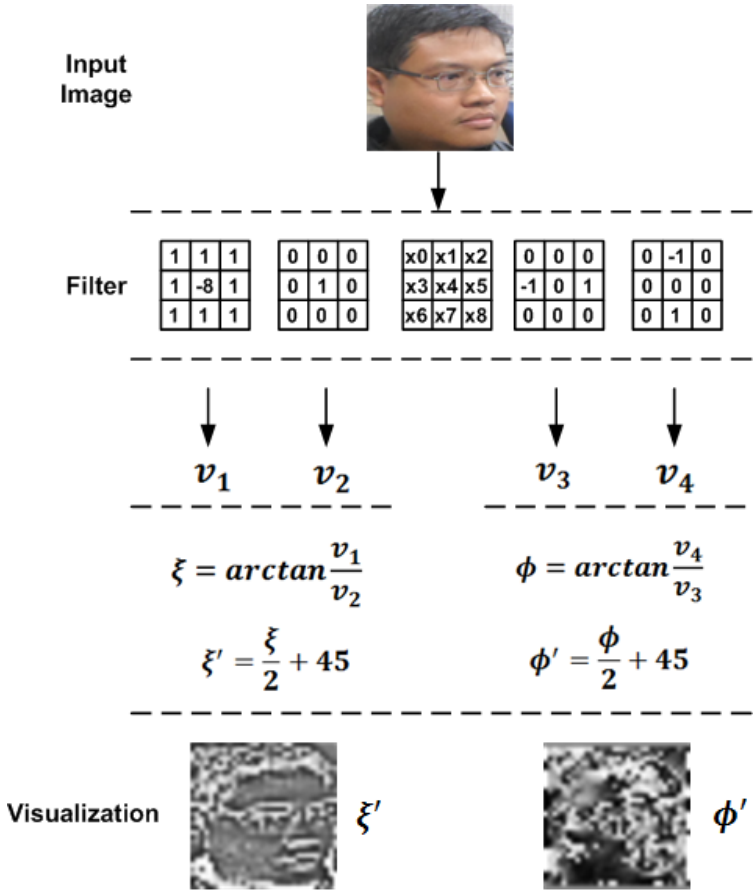


Figure 5.1: Block diagram of Weber-based features generation.

$$\begin{aligned} \xi' &= \tfrac{\xi}{2} + 45, \\ \phi' &= \tfrac{\phi}{2} + 45, \end{aligned} \tag{5.4}$$

where ξ' and ϕ' are the new quantized ξ and ϕ, respectively.

5.2.1.2 Analysis of Weber scales

As described in [74], the size of kernel filters of Weber can be easily scaled. The scale can be generated by regulating the kernel filter size and the radii of the filter as shown in Fig 5.2. The parameter P denotes the number of the neighbors, whereas R determines the radii of the operator. Based on our experiments, applying $R = 3$ gave us improved head orientation estimation results.

5.2.2 Gradients-based feature

The head orientation can be assumed as the changes of edges or shapes of particular parts of the face image [43]. Representation of the changes of edge or shape in the low-resolution image is suitable for characterizing image variations that are insensitive to illumination changes. To strengthen the image characterization, we apply the second order vertical ∇I_{yy} and horizontal ∇I_{xx}

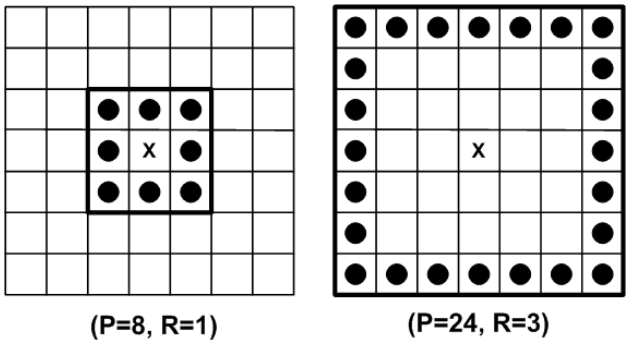


Figure 5.2: Squared symmetric neighborhood for different (P,R).

gradients to the image as follows.

$$\nabla I_{xx} = \frac{\partial^2 I(x,y)}{\partial x^2}, \\ \nabla I_{yy} = \frac{\partial^2 I(x,y)}{\partial y^2}. \tag{5.5}$$

5.2.3 Intensity deviation

The head orientation can also be defined as a change of intensities of a particular region in a given image. We mapped the pixel-based intensity deviation from an image to describe the changes of intensities. Using this map, the changes of location of specific intensities describe the head orientation changes. We apply normalized local and global intensity deviations to identify the distinctive map between each orientation. The specific transition area of the sigmoid function as shown in Fig 5.3 is attracted our attention. We think there are two advantages when using this specific function: the linearity of the output for deviation up to a particular value and can hold the output at a particular significant deviation.

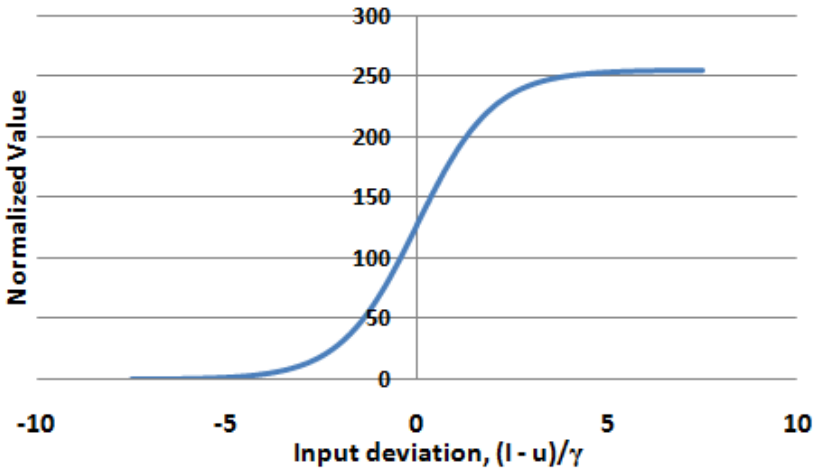


Figure 5.3: Sigmoid function for normalizing intensity deviations ($\gamma = 34$).

The normalized local intensity deviation p is expressed as:

$$p^b_{x,y} = \frac{255}{1+\exp^{(-(I^b_{x,y}-\mu^b)/\gamma)}}, \tag{5.6}$$

where $p^b_{x,y}(b = 1,2,...,B)$ is the normalized local intensity deviation of $b-th$ block, B is the number of blocks, $I^b_{x,y}$ is the pixel's intensity where $(x,y) \subset \mathbb{R}^b$, μ^b is the mean of intensity of the b-th block. γ is a scaling factor for the input deviation. We set $\gamma = 34$ for all experiments (see section 4.1 for the detail).

The normalized global intensity deviation q is expressed as:

$$q_{x,y} = \frac{255}{1+\exp^{(-(I_{x,y}-\mu)/\gamma)}}, \tag{5.7}$$

where μ is the mean of intensity of the image.

5.2.4 Contribution of each feature and the purpose of combination

Weber feature provides a good representation for texture analysis that is insensitive to illumination changes. Our preliminary evaluation shows that this feature is also insensitive to image noise. The gradient feature provides a good representation for object shape, that is also insensitive to illumination changes. Deviation feature can suppress significant intensity deviations, while keeping smaller ones which are useful for classification. The last two features are, however, sensitive to image noise. Combining the three features takes advantages of their strong points and is expected to compensate for the drawbacks, thereby exhibiting a better performance than using a single feature.

5.2.5 Covariance matrix for combining features

We use a covariance matrix as proposed by Tuzel *et al*. [77] to combine all features. We extract a 9-dimensional feature from a grayscale image so that each pixel is composed as a vector of features as follows.

$$\mathbf{I}_i = [x\, y\, I_{xy}\, \xi'_{xy}\, \phi'_{xy}\, \nabla I_{xx}\, \nabla I_{yy}\, p^b_{xy}\, q_{xy}]^T, \tag{5.8}$$

where $\mathbf{I}_i$ is a vector of features of i-th pixel, x,y are the pixel positions, I_{xy} is the image intensity, ξ'_{xy}, ϕ'_{xy} are the new quantized excitation and orientation, respectively. $\nabla I_{xx}, \nabla I_{yy}$ are the second order horizontal and vertical gradients, respectively. p^b_{xy}, q_{xy} are the normalized local and global intensity deviations, respectively.

5.2.6 Symmetric positive definite and distance metric

The covariance matrix is one example of a symmetric positive definite (SPD) matrix. Measurement of the distance between two SPD matrices can be done using $Log-Euclidean$ metric [78]. Following the $Log-Euclidean$ metric, we treat our descriptor using the same manner as presented in [43]. The covariance matrix of block (m,n) is transformed to the matrix logarithm $\log(C_{m,n})$. Each $\log(C_{m,n})$ is then unfolded into a vector space by accommodating $d \times (d+1)/2$ independent values only (half of the upper triangle or the lower triangle of the symmetrical matrix), where d is the number of dimensions of the feature vector.

5.2.7 Weighting scheme for accentuating important parts

A cropped image of a human head usually includes unnecessary parts such as background objects and clothes. Dividing an image into several blocks and controlling the feature's value of each block may improve the discrimination power of a descriptor. Therefore, we use a block importance feature (BIF) scheme, a set of weights that are heuristically predetermined using a Genetic Algorithm (GA) [45].

We chose GA due to the following reasons: (1) it is the most widely used algorithm and has matured as a robust optimization technique [79], and (2) it performs global search which is faster enough compared to the others [80].

For the GA training purpose, some individuals (vectors of genes that represent sets of weights) are initialized using normalized random values. We perform classifier-based optimization and use the hit rate (HR) of the classifier to calculate the fitness value. HR is expressed as

$$HR = \frac{\sum tp}{\sum tp + \sum fp}, \tag{5.9}$$

where tp and fp are the true positive and the false positive, respectively.

The fitness of each individual is evaluated using a combination of the HRs for three principal component analysis (PCA) based classifiers, i.e., PCA+ED (Euclidean distance), PCA+NC (nearest centroid) and PCA+LDA+NC (linear discriminant analysis) as follows.

$$fitness = HR_{(ED)} HR_{(NC)} HR_{(LDA+NC)}. \tag{5.10}$$

This fitness function is intended for learning block importance weights which are effective for various classifiers. Among five classifiers shown below, we use only PCA+ED, PCA+NC, and PCA+LDA+NC for this optimization because they do not need the learning phase, thereby reducing the optimization cost.

Mutation and crossover are then applied to the selected parent. Since the GA works iteratively, the optimization process is repeated using an iteration number as the stopping criteria.

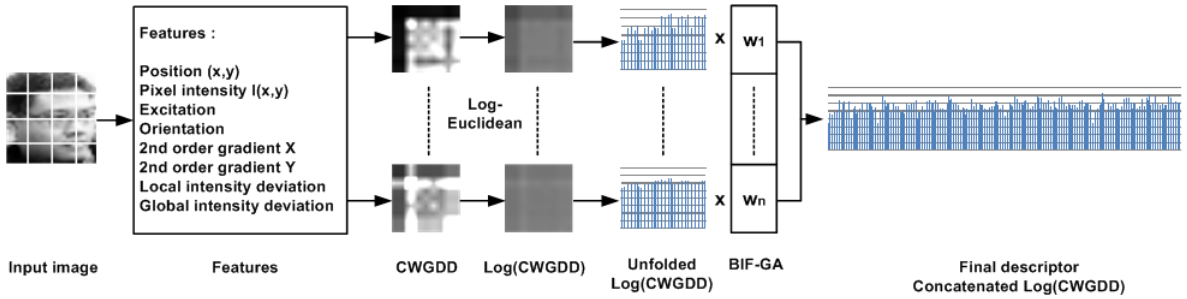


Figure 5.4: Block diagram of our proposed descriptor.

5.2.8 Building a complete descriptor to estimate head orientation

We have presented all of the features we used for supporting our descriptor. Fig. 5.4 shows an input image that is preprocessed using histogram equalization to minimize illumination effects. Afterward, it is divided into B-blocks. A set of determined-features is extracted from each point in each block and processed using our method. By following the $Log-Euclidean$ metric, a $d \times (d+1)/2$-dimensional of feature can be obtained from each block. This feature is multiplied by a BIF-GA weight. A final descriptor is then built using a simple concatenation. For B number of blocks, we have $B \times d \times (d+1)/2$-dimensional of feature.

5.3 Estimating head orientation in a real application

5.3.1 Head and body detection

We extend our work for real application as shown in Fig. 5.5. In a real application, a robust head detector is required. We train our Viola's head detector [47] using 6,000 positive images (mix of original training images of Viola-Jones, Pointing'04 [81], FEI [82] and our AISL databases [83]) and 8,000 negative images. However, its performance frequently degrades in-line with a change in distance. We additionally use a human upper body detector [84] for setting an ROI (region of interest) for head detection.

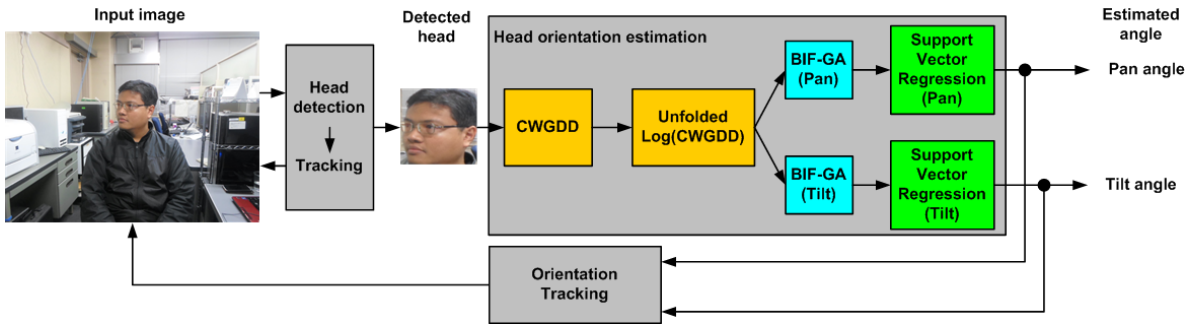


Figure 5.5: Block diagram of an online head orientation estimation system.

5.3.2 Head tracking and orientation smoothing

Our system is composed of two independent processes as shown in Fig. 5.5. The first process is head detection, while the second is head orientation estimation. To get a stable head detection result is very difficult because Viola's detector is specifically designed for detecting faces. Therefore, we apply a Kalman filter to assist our detector with tracking the detection result. Our state model is composed of the detected head position (x_t, y_t), its derivative $(\dot{x}_t, \dot{y}_t)$, and the bounding box size (w_t, h_t). A constant velocity model is utilized to model the head position in an image by considering a time interval Δt:

$$\begin{aligned} x_t &= x_{t-1} + \Delta t \dot{x}_{t-1} + \varepsilon_{x_t}, \\ y_t &= y_{t-1} + \Delta t \dot{y}_{t-1} + \varepsilon_{y_t}, \\ \dot{x}_t &= \dot{x}_{t-1} + \varepsilon_{\dot{x}_t}, \\ \dot{y}_t &= \dot{y}_{t-1} + \varepsilon_{\dot{y}_t}. \end{aligned} \tag{5.11}$$

The bounding box size is maintained as

$$\begin{aligned} w_t &= w_{t-1} + \varepsilon_{w_t}, \\ h_t &= h_{t-1} + \varepsilon_{h_t}. \end{aligned} \tag{5.12}$$

The state can be expressed as a tuple $\mathcal{X} = \{x_t, y_t, \dot{x}_t, \dot{y}_t, w_t, h_t\}$. Our Kalman filter model is expressed as $\mathcal{X}_t = \mathbf{F}_t \mathcal{X}_{t-1} + \varepsilon_t$, where $\mathbf{F}_t$ is the state transition model and $\varepsilon_t = N(0, Q_t)$ is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance $Q_t = \{2, 2, 0.5, 0.5, 2, 2\}$.

On the other hand, the output of the head orientation estimator also fluctuated. To reduce fluctuations of the orientation estimation results, we also utilize a Kalman filter to smooth the head orientation estimation. Our state model is composed of the estimated angles (α_t, β_t) and its derivative $\omega_{\alpha_t}, \omega_{\beta_t}$. Estimation of the pan and the tilt angles use a constant angular velocity model as follows.

$$\begin{aligned} \alpha_t &= \alpha_{t-1} + \Delta t \, \omega_{\alpha_{t-1}} + \varepsilon_{\alpha_t}, \\ \beta_t &= \beta_{t-1} + \Delta t \, \omega_{\beta_{t-1}} + \varepsilon_{\beta_t}, \\ \omega_{\alpha_t} &= \omega_{\alpha_{t-1}} + \varepsilon_{\omega_{\alpha_t}}, \\ \omega_{\beta_t} &= \omega_{\beta_{t-1}} + \varepsilon_{\omega_{\beta_t}}. \end{aligned} \tag{5.13}$$

The state can be expressed as a tuple $\mathcal{Y} = \{\alpha_t, \beta_t, \omega_{\alpha_t}, \omega_{\beta_t}\}$. The Kalman filter model is expressed as $\mathcal{Y}_t = \mathbf{G}_t \mathcal{Y}_{t-1} + \varepsilon_t$, where $\mathbf{G}_t$ is the state transition model and $\varepsilon_t = N(0, P_t)$ is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance $P_t = \{2, 2, 1, 1\}$.

5.4 Experiments

5.4.1 Analysis of γ values

We performed extensive experiments using Pointing'04 head pose database (7 pan angles) to find γ which is suitable for our intensity deviation feature. We evaluated several values of γ within a range [1:128]. We found that $\gamma = 34$ achieved the highest classification accuracy and the lowest angle error in average of five classifiers as shown by Fig 5.6.

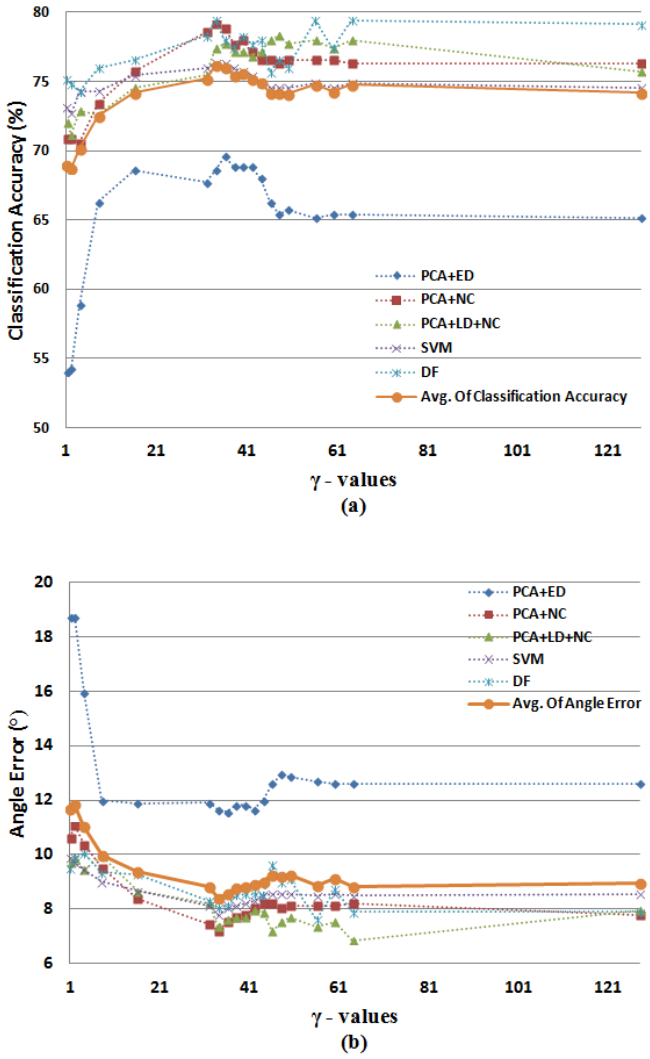


Figure 5.6: Comparison of different γ values; (a) the accuracy, and (b) the MAAE.

5.4.2 Analysis of different block size

To improve the discrimination power of the descriptor, we divided an input image into a set of blocks. However, finding the best size of blocks is rather difficult. To figure out this matter, we conducted a small experiment to find the best size of blocks by trying it on Pointing'04 database using 2×2, 3×3, 4×4, 6×6, 9×9, and 12×12. Based on our experiment, we found that the block size of 4×4 achieves the best accuracy and the block size of 6×6 achieves the second best accuracy.

5.4.3 Optimizing weights using genetic algorithm

5.4.3.1 Dataset

We use 1,680 cropped-images of 105 subjects that are taken from two databases; Pointing'04 head pose database [81] and NCKU database [85]. The ranges of pan and tilt angles are $\pm 90^o$ and $\pm 60^o$, respectively. We select a set of images with interval $\pm 30^o$ only. The first 10 subjects of Pointing'04 and the first 60 subjects of NCKU are utilized as training data, while the remaining are used as a probe set. All images are resized into 36×36 pixel.

5.4.3.2 Optimization results

Based on the result of section 4.2, we only pay attention to analyze the block size of 4×4 and 6×6 in our system. The weights are obtained after running GA for 100 iterations with 20 generated-individuals. Fig. 5.7 and Fig. 5.8 show the optimal weights of 4×4 and 6×6 blocks, respectively. Table 5.1 shows the comparison of the effectiveness using block size of 4×4 and 6×6. The Pointing'04 head pose database is used as the evaluation dataset. For some classifiers, increasing the block size is not give us better results for estimating pan orientation. However, opposite conditions occur for estimating tilt orientation.

Table 5.1: Experimental results of CWGDD and CWGDD+BIF-GA using different size of blocks. The Pointing'04 head pose database is utilized to evaluate the performances.

Pose	**Block**	**BIF**	**Accuracy %**					**MAAE (^{o})**				
				+PCA		**+SVM**	**+DF**		**+PCA**		**+SVM**	**+DF**
			+ED	**+NC**	**+LDA**			**+ED**	**+NC**	**+LDA**		
pan	4x4	no	68.57	**79.14**	**77.43**	75.71	**79.43**	12.09	**7.20**	**7.37**	7.80	**8.06**
pan	6x6	no	**73.14**	77.43	74.29	**81.14**	77.14	**9.00**	8.66	9.26	**6.69**	**8.06**
pan	4x4	yes	69.43	**80.00**	**77.14**	78.57	**78.57**	11.74	**7.20**	**7.80**	6.94	**7.97**
pan	6x6	yes	**70.00**	72.86	75.14	**79.14**	77.43	**10.63**	9.43	8.49	**6.86**	8.14
tilt	4x4	no	58.67	63.29	65.29	**65.71**	74.57	**13.49**	**11.57**	**11.20**	**10.34**	7.80
tilt	6x6	no	**60.86**	**66.86**	**67.14**	64.00	**77.14**	14.83	12.09	11.49	12.17	**6.94**
tilt	4x4	yes	62.86	66.00	67.14	**70.40**	76.00	**12.06**	10.94	10.11	**9.37**	7.37
tilt	6x6	yes	**63.14**	**68.00**	**70.86**	65.71	**75.29**	13.20	**10.63**	**9.34**	11.14	**7.29**

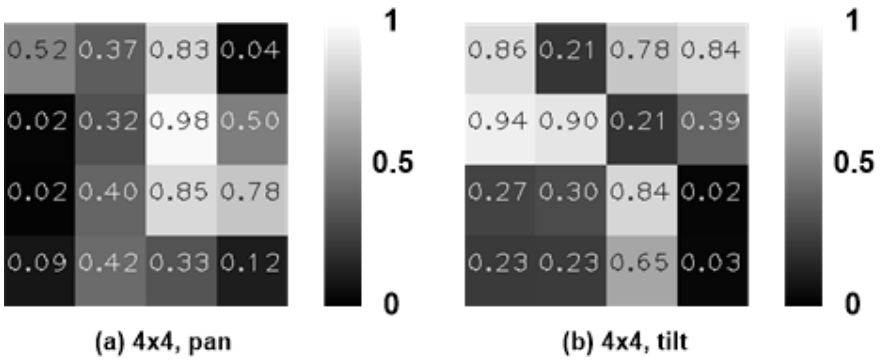


Figure 5.7: The optimal weights for 4x4 blocks; (a) pan and (b) tilt.

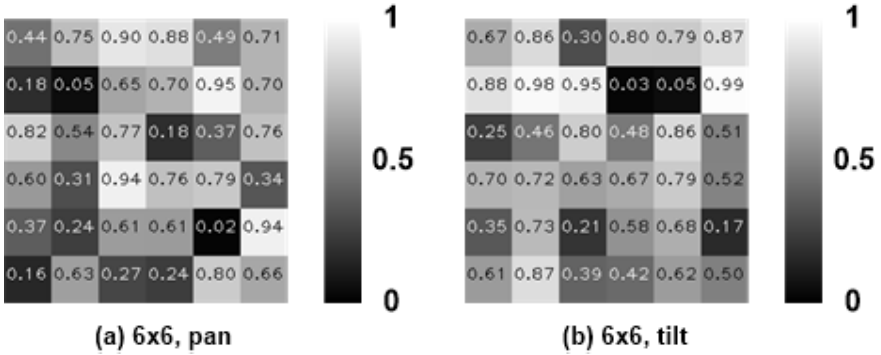


Figure 5.8: The optimal weights for 6x6 blocks; (a) pan and (b) tilt.

Fig. 5.7 shows that the change of weights in the horizontal direction is relatively larger than in the vertical direction for pan and that in the vertical direction is larger for tilt. It seems that this coincides with our intuition. Fig. 5.8 shows the weights for 6×6 blocks and exhibits a similar weight distribution, although the above tendency is less obvious.

5.4.4 Offline experiment using head pose database

5.4.4.1 Dataset

We use three head pose databases that are set into datasets for different purposes as summarized in Table 5.2. Dataset DS1, DS2, DS3, DS4 and DS5 are used independently for offline evaluations. In this section, we compare our method with COG [43], IA-LDQP [41], WLD [74], and CovGA [42]. We re-implement COG, IA-LDQP, and WLD, while we used the result of CovGa directly from their paper. To make fair re-implementations and comparisons, we also use the same parameters settings, i.e., 32×32 pixel (COG and WLD) and 36×36 pixel (CWGDD), 4×4 blocks (COG, IA-LDQP, WLD and CWGDD).

We use Principal Component Analysis + Euclidean Distance (PCA+ED), Principal Component Analysis + Nearest Centroid (PCA+NC), Principal Component Analysis + Linear Discriminant

Table 5.2: Experimental setup of head pose databases

Dataset	Database	Number of Image (Person)	ROI	Angles				Number of Image (Class) in Use	Cross Valid. (k-fold)
				Pan	Tilt	Range	Step		
DS1	CAS-PEAL [71]	4,200 (200)	face	7	3	$\pm 45^o$	$\pm 15^o$	4,200 (7 pans)	4
DS2	Pointing'04 [81]	2,790 (15 (x2))	head	13	9	$\pm 90^o$	$\pm 15^o$	1,050 (7 pans, $\pm 30^o$)	3
DS3	Pointing'04 [81]	2,790 (15 (x2))	head	13	9	$\pm 90^o$	$\pm 15^o$	1,050 (5 tilts, $\pm 30^o$)	3
DS4	AISL [83]	3,420 (20 (x3))	head	19	3	$\pm 90^o$	$\pm 10^o$	1,260 (7 pans, $\pm 30^o$)	4
DS5	AISL [83]	3,420 (20 (x3))	head	19	3	$\pm 45^o$	$\pm 45^o$	1,260 (3 tilts)	4
DS6	NCKU [85]	6,600 (90)	head	19	1	$\pm 90^o$	$\pm 5^o$	630 (7 yaws, $\pm 30^o$)	-

Table 5.3: Parameters setting of each classifier

Method	Parameter	Value
PCA	Number of eigenvalues	100
NC	Number of centroid	10
	Iteration	30
SVMs	Type	multiclass C-SVC
	Kernel type	linear
	C cost	1
DF	Number of tree	500
	Number of random features	5

Analysis + Nearest Centroid (PCA+LDA+NC), Support Vector Machines (SVMs) [55], and Decision Forest (DF) [86] as the classifiers. The parameters used in each classifier are summarized in Table 5.3. The performance of each method is measured using the percentage of accuracy and the stability of estimated angle of orientation that is expressed as the mean absolute of angle error (MAAE). $k-fold$ cross validation is used for evaluation.

5.4.4.2 Experimental results using CAS-PEAL head pose database

The CAS-PEAL dataset (DS1) is used to evaluate the performance of each method in terms of estimating face's pan angle. We reduce the effect of background by zooming up a face region that was precisely cropped using eye position. The tilt angles are grouped based on its pan angle. This database gives a distinctive challenge where the appearances of nearby classes are difficult to distinguish due to close angle. Examples of cropped-faces are shown in Fig. 5.9.

Our experiments show that our CWGDD outperforms the others for all classifiers that are proven by its accuracy and mean absolute angle error (MAAE) as shown in Table 5.4. On average, our method's accuracy is 2.69% higher than COG. The estimation result stability is also impressive. It is proven by its MAAE 0.42^o smaller than COG.

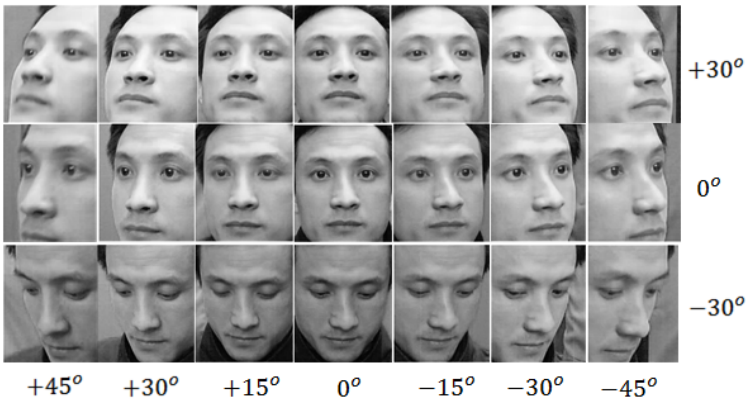


Figure 5.9: Example of CAS-PEAL face pose database.

Table 5.4: Experimental results of estimating the pan angle using CAS-PEAL face pose database

Method	Accuracy (%)					MAAE (^{o})				
	+PCA			+SVM	+DF	+PCA			+SVM	+DF
	+ED	+NC	+LDA			+ED	+NC	+LDA		
COG [43]	70.30	90.00	95.10	85.52	90.62	5.46	1.61	0.74	1.89	1.41
IA-LDQP [41]	47.50	70.00	38.00	78.52	44.30	13.26	5.72	18.69	3.09	17.57
CovGa [42]	N/A	N/A	94.20	N/A	N/A	N/A	N/A	N/A	N/A	N/A
WLD [74]	73.10	82.50	85.80	83.33	60.30	4.75	2.86	2.24	2.87	7.69
CWGDD (Ours)	**74.86**	**91.10**	**95.71**	**91.19**	**93.00**	**4.41**	**1.29**	**0.70**	**1.34**	**1.05**

5.4.4.3 Experimental results using Pointing'04 head pose database

The Pointing'04 database is used to evaluate the performances of each method for estimating the head pan and tilt angles. Each image was manually cropped where we cannot guarantee its precision. Examples of cropped-heads are shown in Fig. 5.10. As in the other works, the pan and the tilt angles are estimated separately. For estimating pan, tilt angles are grouped into the same pan angle class while pan angles are grouped into the same tilt angle class for estimating tilt angle.

The first experiment is conducted to see the performances of each method to estimate the head pan using dataset DS2. Table 5.5 shows that our CWGDD and CWGDD+BIF-GA outperform the others for all classifiers for both the accuracy and the MAAE. On average, our method's accuracy is 7.45% higher than COG, while its MAAE is 4.36^o smaller than COG. CWGDD+BIF-

Table 5.5: Experimental results of estimating the pan angle using Pointing'04 head pose database

Method	Accuracy (%)					MAAE (^{o})				
	+PCA			+SVM	+DF	+PCA			+SVM	+DF
	+ED	+NC	+LDA			+ED	+NC	+LDA		
COG [43]	63.10	65.40	69.70	69.14	75.71	15.26	13.29	12.26	14.49	9.00
IA-LDQP [41]	42.90	41.10	54.60	59.43	43.10	38.29	40.29	25.49	16.43	37.54
WLD [74]	45.10	50.60	67.40	66.57	48.90	21.77	31.80	11.57	11.13	21.69
CWGDD (Ours)	68.57	79.14	**77.43**	75.71	**79.43**	12.09	**7.20**	**7.37**	7.80	8.06
CWGDD+BIF-GA	**69.43**	**80.00**	77.14	**78.57**	78.57	**11.74**	**7.20**	7.80	**6.94**	**7.97**

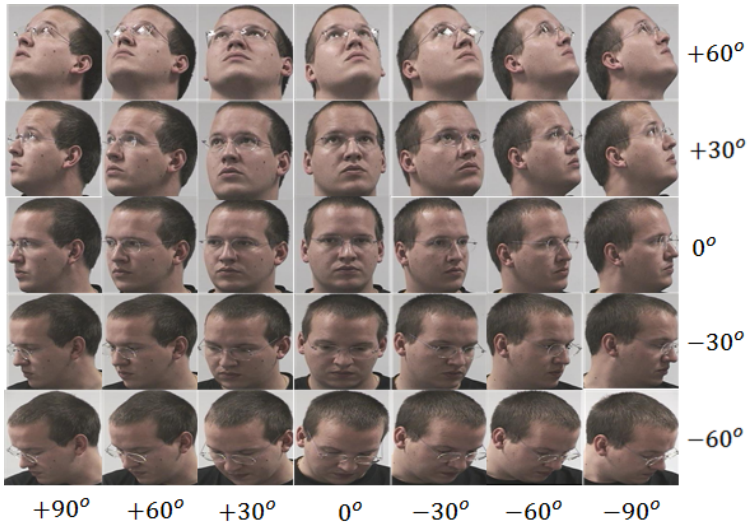


Figure 5.10: Example of Pointing'04 head pose database.

Table 5.6: Experimental results of estimating the tilt angle using Pointing'04 head pose database

Method	Accuracy (%)					MAAE (^{o})				
	+PCA			+SVM	+DF	+PCA			+SVM	+DF
	+ED	+NC	+LDA			+ED	+NC	+LDA		
COG [43]	61.14	65.14	65.43	**71.71**	72.00	13.89	11.40	11.23	**8.66**	9.43
IA-LDQP [41]	43.40	44.00	47.10	52.86	20.00	25.29	23.40	25.49	13.25	36.00
WLD [74]	57.70	55.40	56.60	57.00	46.00	14.23	14.23	13.63	19.60	20.06
CWGDD (Ours)	58.67	63.29	65.29	65.71	74.57	13.49	11.57	11.20	10.34	7.80
CWGDD+BIF-GA	**62.86**	**66.00**	**67.14**	70.40	**76.00**	**12.06**	**10.94**	**10.11**	9.37	**7.37**

GA successfully improves the accuracy by 0.69% higher and reduce its MAAE by 0.17^o smaller than CWGDD.

The second experiment was performed to see the performances of each method to estimate the head tilt using dataset DS3. Table 5.6 shows that our CWGDD performances are bit worse than COG. In average, our CWGDD accuracy is 1.38% lower than COG, however, its MAAE is better than COG by achieving 0.04^o lower. Improving CWGDD with BIF-GA increases its accuracy by 2.77% and reduces the MAAE by 0.91^o.

From the experiments, gradient-based method is particularly effective to characterize a

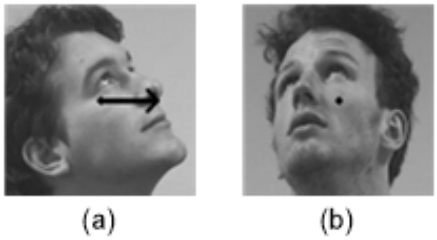


Figure 5.11: Examples of failure cases on Pointing'04 database.

human head tilt. However, our BIF-GA scheme is able to gain the performance of CWGDD. Fig. 5.11 shows examples of incorrect estimation. They are due to an incompletely cropped head area (Fig. 5.11(a)) or a high portion of background (Fig. 5.11(b)).

5.4.4.4 Experimental results using AISL head orientation database

The AISL head orientation database is used to evaluate the performance of each method in estimating the head pan and tilt angles when the background changes. This database simulates the changes of background for indoor environments which might confound the estimation results. In this experiment, again, we cannot precisely crop each image. The pan and the tilt angles are estimated separately. All background variations are combined together as shown in Fig. 5.12.

The first experiment is conducted to see the performances of each method to estimate the head pan using dataset DS4. Table 5.7 shows that our CWGDD and CWGDD+BIF-GA outperform the others for all classifiers. On average, CWGDD accuracy is 2.54% higher than COG. BIF-GA successfully improves the accuracy by 2.60%. The second experiment was performed to see the performances of each method to estimate the head tilt using dataset DS5. Our CWGDD outperforms the other methods for PCA-based classifiers. On average, CWGDD accuracy is 6.60% higher than COG. BIF-GA successfully improves the accuracy by 1.84%. From the experiments, a single cue-based methods cannot get good results because of their sensitivity to the noises.

Table 5.7: Experimental results of estimating the pan and the tilt angles using AISL head orientation database

Method	**Pan - Accuracy (%)**					**Tilt - Accuracy (%)**				
	+PCA			**+SVM**	**+DF**	**+PCA**			**+SVM**	**+DF**
	+ED	**+NC**	**+LDA**			**+ED**	**+NC**	**+LDA**		
COG [43]	67.62	77.78	79.05	80.63	77.46	65.40	64.76	41.59	**77.14**	**71.75**
IA-LDQP [41]	17.46	19.68	16.83	47.62	53.65	34.92	33.65	35.24	63.81	64.76
WLD [74]	34.29	60.95	67.30	65.08	58.73	56.51	54.60	33.97	66.44	61.90
CWGDD (Ours)	66.35	**79.37**	**83.49**	85.40	80.63	67.94	73.02	70.79	73.97	67.94
CWGDD+BIF-GA	**73.65**	79.05	81.59	**90.48**	**83.49**	**72.70**	**73.65**	**73.97**	74.29	68.25

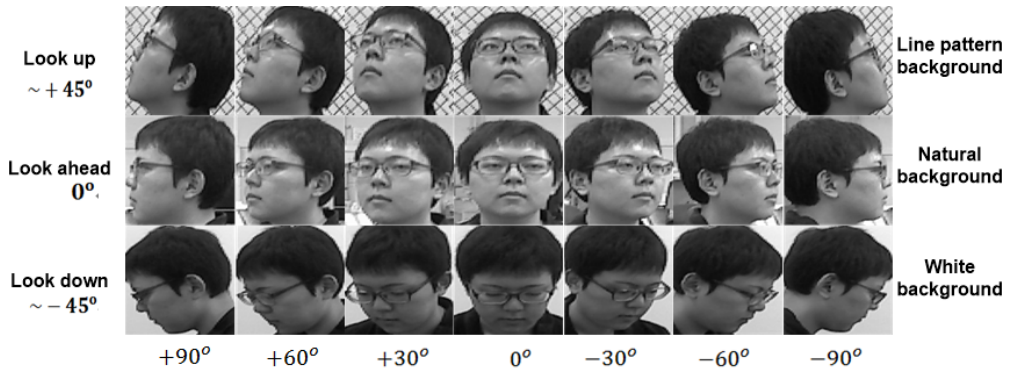


Figure 5.12: Example of AISL head orientation database.

Figure 5.13: Examples of failure cases on AISL database.

Fig. 5.13 shows examples of incorrect results. Our method fails for Fig. 5.13(a) and Fig. 5.13(b) due to a different hair color from black, which causes a wrong intensity mapping when we used a dataset with only black hair persons for training. Fig. 5.13(c) shows an unbalanced cropped head area case.

5.4.4.5 Comparison with a Deep Learning-based method

Deep Learning has recently been gaining much popularity due to its high performance in image and speech processing. Cai *et al*. [87] developed a multi-class classification in head pose estimation based on a Deep Convolutional Neural Network (DCNN). They used an eight-layer DCNN.

We compared our CWGDD+PCA+LDA+NC with Cai's method using the CASPEAL head pose database. For our method, we used a subset containing 4,200 images of 200 subjects whose IDs range from 401 through 600, and performed 3-fold cross validation. They used the same 200 subjects for testing with using the remaining 840 subjects for training. They also enriched the variation of training dataset by shifting and scaling the training images, thereby generating 410,700 images for training. The 4,200 test images were divided into three subsets, each of which includes 1,400 images, and the results for these subsets are averaged to obtain the final accuracy.

Table 8 shows the comparison result. DCNN outperforms ours with 1.31% higher accuracy. This result is, however, obtained by using about 147 times larger training data. It seems necessary to test the methods with more various conditions for more detailed comparison. This is one of our future work.

Table 5.8: The comparison result against the Deep Convolutional Neural Network

Method	**#Training samples**	**#Test samples**	**Classification Accuracy (%)**
DCNN [87]	410,700	1,400	97.17
Ours	2,800	1,400	95.86

5.4.5 Online experiment using videos

We have successfully built a simple yet robust head orientation descriptor that works very fast and is applicable to real applications. To test our method online, we utilized two captured videos of a real scene at our campus. We combine DS2, DS3 and DS6 as the training dataset. COG is utilized as a baseline method because it is the closest competitor to ours. In this experiment, we utilize Epsilon Support Vector Regression (epsilon-SVR) [88] with polynomial kernel as the estimator. The tolerance of termination criterion is set to 0.001, coef0 is set to 0, and degree in kernel function is set to 3. We select block size of 6×6 due to the best estimation results of pan orientation using SVMs as shown in Table. 5.1.

The main objective of this experiment was to compare the feasibility of our descriptor and COG for estimating the head orientation in a video sequence in both indoor and outdoor

Table 5.9: Comparison result in the indoor experiment

Method	#Frame	#Detected Body	#Detected Head	Accuracy (%)	
				Pan	Tilt
COG	460	357	309	59.52	39.46
CWGDD+BIF-GA (Ours)	460	357	305	52.96	61.90

Figure 5.14: Experimental results of an online indoor scene. First, the human upper body is detected and tracked (red bounding box). Head detection and head orientation estimation supported by tracking is performed within the body's bounding box. The estimated head orientation is shown by yellow (pan) and green (tilt) arrows within the bounding box. Miss-orientation estimation (#159) occurs due to a severe illumination, while false positive detections of the human upper body and miss-orientation estimation (#387) occur due to an ambient light and a less fit of the head tracking.

environments. We roughly divide the pan into three general directions: right ($\alpha > +10^o$), upright frontal ($-10^o \leq \alpha \leq +10^o$) and left ($\alpha < -10^o$), while the tilt is also divided into three general directions: up ($\beta > +10^o$), upright frontal ($-10^o \leq \beta \leq +10^o$) and down ($\beta < -10^o$).

The first video is taken in a corridor where a targeted person is walking and following a moving camera with a distance of 1.5 - 2 meters. In this experiment, human upper body detection is used to localize the targeted person's body due to the reduced stability of our head detector for a distant person. Our system performs head detection, estimates the pan and the tilt angles at once, and tracks them to provide more robust estimation. Our system is able to estimate the target's head orientation in frames as shown in Fig. 5.14. However, the pan and the tilt errors sometimes occur during the experiment due to a severe illuminations (#159), an ambient lights and a less fit of the head tracking (#387).

We quantitatively compare the performance of our CWGDD+BIF-GA against COG by manually counting the number of frame, precision detected-body, precision detected-head, pan and tilt orientations accuracies as shown in Table 5.9.

The second video is taken outdoor where a targeted person is walking and following a moving camera with a distance of 2 - 2.5 meters. In this experiment, the performance of our head orientation estimation and the human upper body is better than it was in the first experiment as shown in Fig. 5.15. False body and head detection, and miss-orientation estimation are reduced. The comparison results of our CWGDD+BIF-GA and COG are shown in Table 5.10.

Based on the experimental results, the pan estimation accuracy of COG is slightly better than ours in the indoor experiment. However, COG fails to retain the accuracy for estimating the tilt. In general, our CWGDD+BIF-GA outperforms COG for indoor and outdoor experiments by achieving a better average in the accuracies.

The performance of our feature is stable because it utilizes a more variety of cues, that is, edge/shape (gradient), texture (Weber), and intensity patterns (deviation), while COG uses only gradient features. If a noise level of image is high, this affects the quality of gradient features, thereby degrading the performance of COG-based method. This is supported by our experiments that COG performs good for the off-line databases (CASPEAL and Pointing'04) but worse for AISL database and in real experiments.

Table 5.10: Comparison result in the outdoor experiment

Method	**#Frame**	**#Detected Body**	**#Detected Head**	**Accuracy (%)**	
				Pan	**Tilt**
COG	345	273	260	68.56	62.33
CWGDD+BIF-GA (Ours)	345	270	255	74.33	78.43

Figure 5.15: Experimental results of an online outdoor scene. First, the human upper body is detected and tracked (blue bounding box). Head detection and head orientation estimation supported by tracking is performed within the body's bounding box. The estimated head orientation is shown by yellow (pan) and green (tilt) arrows within the bounding box. The human upper body detection, the head detection and the head orientation estimation achieve good performances in this experiment.

5.4.6 Computation time

Besides the accuracy, we also measured the averaged processing time of each method to complete the descriptor generation. The average time for completing the descriptor generation using COG, IA-LDQP, WLD, CovGa, CWGDD, and CWGDD+BIF-GA are around 0.69 ms, 32.44 ms, 0.98 ms, more than 96.79 ms, 0.79 ms, and 0.79 ms, respectively. The evaluation is conducted using Microsoft Visual C++ running on a personal computer system equipped with 3.60 GHz Intel processor i7 supported by 16 GB of RAM. Our weighting scheme does not burden our method because the optimization process has been undertaken in advance. Increasing the block size from 4×4 to 6×6 increases the computation time by about 0.15 ms.

Online experiments using indoor and outdoor videos show that our method is fast enough by achieving 11 - 16 fps. It implies that all processes such as the human upper body detection and tracking, the head detection, and the head orientation estimation only take about 90.5 - 62.5 ms in total.

Chapter 6

Implementation of Q-Learning based Social Force Guiding Model for the Guide Robot

In this chapter, we present our experimental results of the implementation of the Q-Learning based Social Force Guiding Model (QL-SFGM) for the guide robot. The implementation is validated by two approaches, that are simulations using a realistic 3D simulator and implementation in real environments. The platform and experimental setup, the results of experiments, and the supporting data will be presented in the next sections.

6.1 Robot Platform

We used a Pioneer 3-DX robot model as shown in Fig. 6.1 for both experiments using a realistic 3D simulator and real robot implementation. We modified the robot platform by adding a new structure of the robot to accommodate the placement of a back-facing camera. Two laser range finders (LRFs) were mounted in front and rear of the robot. A notebook PC was carried on the robot to process all tasks, including coordination task module, navigation task module, and our motion planner and control (QL-SFGM) module. Briefly, we will describe each part specification.

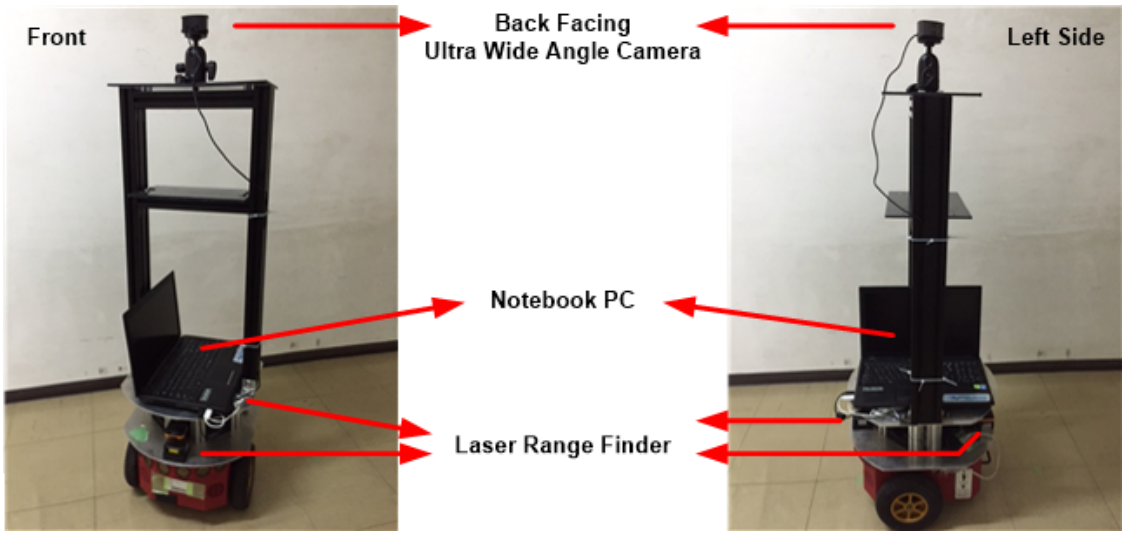


Figure 6.1: Our guide robot structure utilizing a Pioneer 3-DX robot platform.

6.1.1 Robot Specification

The Pioneer is the world's most popular research mobile robot. The Pioneer's versatility, reliability, and durability have made it the reference platform for robotics research. Pioneer 3-DX is a small lightweight two-wheel two-motor differential drive robot. Some important specifications are as follows.

- weight : 9 kg (no load)
- weight (+ additional structure + notebook PC) : 20 kg
- turn radius : 0 cm
- max. forward/backward speed : 1.2 m/s
- rotation speed : 300^o/s

6.1.2 Sensor Specification

Our robot is equipped with two kind of sensors, they are camera and laser range finder. The specification of each sensor will be described as follows.

6.1.2.1 Camera

We used an ultra-wide angle camera BSW180ABK with a specification as follows.

- Viewing angle : 180^o wide-angle
- Interface : USB2.0
- Sensor type : CMOS
- Resolution : 1280 x 480 pixels
- Max. frame rate : 30 fps
- Focal length : 70 cm $\sim \infty$

6.1.2.2 Laser Range Finder

We used two different type of LRF, the front LRF is UTM-30LX and the rear LRF is UHG-08LX. UTM-30LX has a specification as follows.

- Detection range : 0.1 to 30 m, 270^o
- Accuracy : 0.1 to 10 m: $\pm$30 mm, 10 to 30 m: $\pm$50 mm

- Angular resolution : 0.25^o (360^o / 1,440 steps)
- Scan time : 25 msec/scan
- Interface : USB2.0

while UHG-08LX has a specification as follows.

- Detection range : 0.03 to 11 m, 270^o
- Accuracy : 0.1 to 1 m: $\pm$30 mm, 1 to 8 m: $\pm$150 mm
- Angular resolution : 0.36^o (360^o / 1,024 steps)
- Scan time : 67 msec/scan
- Interface : USB2.0

6.1.3 Processor Specification

To performing all tasks and executing the QL-SFGM algorithm, we used a notebook PC with a specification as follows.

- Processor : Intel(R) Core(TM) i7-4700MQ CPU @2.40 GHz
- Installed memory : 16.0 GB
- System type : 64-bit OS
- Operating system : Windows 10

6.2 Software Platform

We build task modules and the motion planning and control (QL-SFGM) module using Microsoft Visual C++ programming. For experiments using the real robot, each task is packaged in the form of Robotic Technology (RT) component using an open source robotic technology middleware which is developed and distributed by Japan's National Institute of Advanced Industrial Science and Technology, named OpenRTM-aist [89].

For experiments using the simulator, we use V-Rep [90], a realistic 3D robot simulator to train our motion planning and control (QL-SFGM) algorithm and evaluate its performance. All modules are built in one project using Visual C++ programming that is remotely connected to V-Rep simulator through an API client-server. Several simulated environments are built for training and testing purposes.

6.3 Supporting Data: Ideal or Comfortable Distance for Coordination Task

Before starting the discussion about the experimental results, we present the experimental result of an ideal distance for coordination task measurement. This distance is important since in coordination task (section 3.2.3) we used this distance to calculate a partner force, $\mathbf{F}^{\rho}$, with respect to his/her position. We have declared that a mean value which represents a comfortable distance when a human partner follows a guide robot, $\mu_{\rho} = 2.2$ meters. This value was obtained from experiments to measure this ideal distance by collecting real data from six people of our lab members. Each person was asked to follow a remote mobile robot. The person's position was detected from his/her leg using a range data based person identification system. The detected position is then converted into a distance. All persons were asked to follow the robot two times and their data was recorded. We asked all participants to just relax and behave as usual when following the robot. Table. 6.1 shows the mean and variance of each person when following the robot.

6.4 Learning the Motion Planning and Control using Simulator

Learning a system under MDP-based problem with a real implementation is very risky because the system initially has no basic skill. Since the MDP-based system is usually based on episodes, the skill grows in line with the number of episodes. Due to this reason, we perform the training based on simulation. To imitate a real situation and environment, we utilize V-Rep to build the simulated situation and environment, the simulated partner, and run the robot by executing our proposed motion planning and control algorithm (QL-SFGM).

Table 6.1: The ideal or comfortable distance for coordination task

Participant	Distance (meter)	
	Mean	Variance
Person 1	2.17	0.28
Person 2	2.98	0.13
Person 3	2.13	0.31
Person 4	1.96	0.32
Person 5	1.39	0.31
Person 6	2.61	0.33
Average	2.21	0.28

6.4.1 Simulating Human Partner Behavior

As described in chapter 3, the behavior of a human partner is composed of three features: keep the distance with the robot, change his/her walking direction and change his/her head orientation. To simulate those features, we build two functions: (1) function to control walking speed and direction while following the robot and (2) function to control human head orientation. In this simulation, we assume that the human partner is always detected, identified and tracked by our robot. This means that we do not show the process of detection and identification here.

In the first function, the walking speed, v_ρ in m/sec, is randomized by following a formulation as follows.

$$v_\rho = \begin{cases} randomize(0.25, 1) & \text{if distance to the robot} > 0.5 \text{ meter} \\ 0 & \text{if distance to the robot} < 0.5 \text{ meter} \end{cases} \tag{6.1}$$

The walking direction, α, is following a formulation as follows.

$$\alpha = tan^{-1}\left(\frac{y_R - y_\rho}{x_R - x_\rho}\right) \tag{6.2}$$

where (x_R, y_R) is the robot position, and (x_ρ, y_ρ) is the partner position. By using this function, a human partner will always follow the robot. In a case when a human partner is blocked by other people, he will avoid other by using local sensor and simple collision avoidance algorithm. He will resume his walking direction after the obstacle has gone.

In the second function, we build three types of head orientation: always looking at the robot, look at the robot then alternate his head direction following a sinusoidal function after a random time step, and look at the robot then turn his head orientation following a sigmoid function after a random time step and keep the last head orientation until reaching the goal. This kind of head orientations may not be natural, however, they are enough to enrich the learning situations.

6.4.2 Learning the Motion Planning and Control

6.4.2.1 Parameter Setting

Before starting the training, we set several robot's parameters: $m = 20$ kg, $v^0_{max} = 1$ m/sec, $\tau = 0.015$ sec, $k_{max} = 20$ N, $\Psi_{max} = 3.6$ m, $\lambda_{max} = 1$, and the ellipse's major and minor radius are 1.8 m and 0.7 m, respectively. The Q-learning algorithm parameters setting is shown in Table. 6.2. All of the parameters with subscript *max* are proportionally divided by the number of actions.

Table 6.2: Q-learning parameters settings

Parameters	Value
Number of training episodes per scenario	2,000
Number of states	43,200
Number of actions	25
Initial ε	0.6
Learning rate α	0.1
Discount factor γ	0.7
$\mathbf{F}^{g}_{max}$	200 N

6.4.2.2 Scenarios

We built three scenarios as shown in Fig. 6.2 to train our robot. These scenarios are used to complete as much as possible states. The reason of choosing more indoor scenarios than outdoor is that controlling the robot behavior in an indoor environment frequently more difficult than in outdoor because the robot's free space is limited by static objects, i.e. walls, furnitures, etc. In many

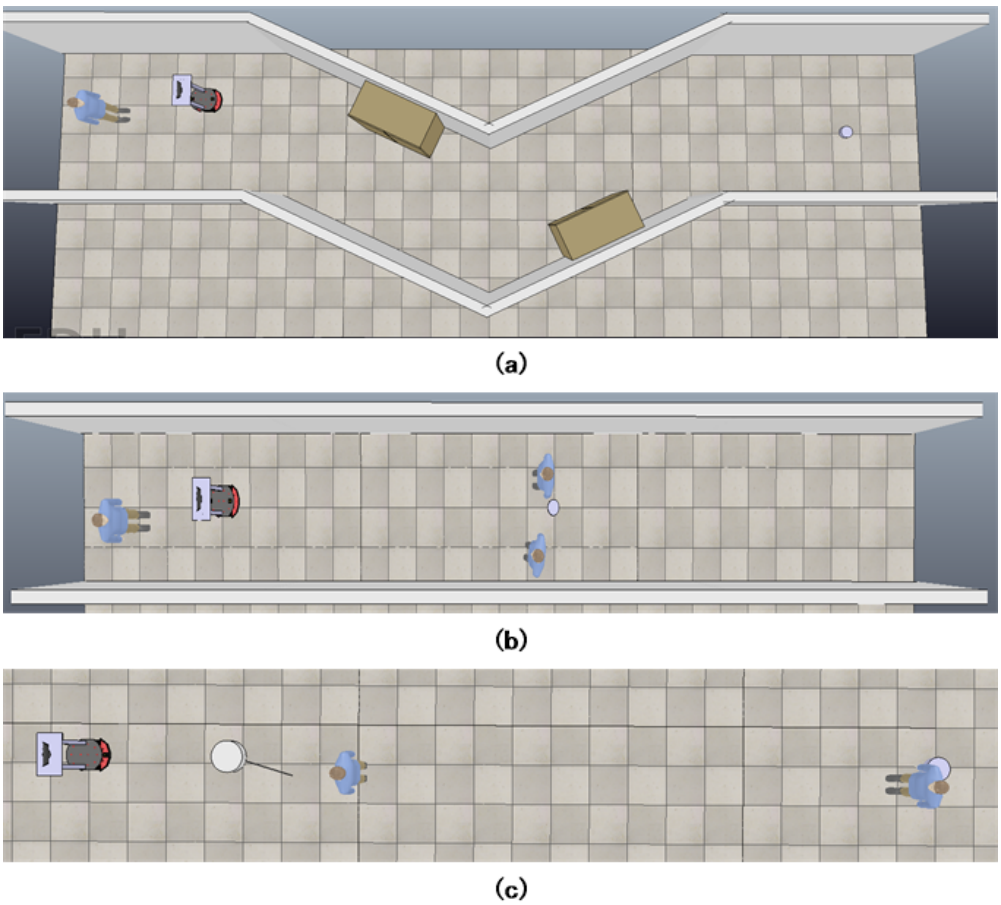


Figure 6.2: Scenario for training. (a) indoor (two cupboards), (b) indoor (two free moving dynamic persons), and (c) outdoor (one static person and one dynamic person with trajectory). A bluish cylinder is the goal.

cases, these kind of static objects affect to the oscillation level of the robot movements. Initially, we have generated 43,200 states by following our design in Section 3.5.2.1 (Configuring a state).

6.4.2.3 Results

As the training result, we presented two results of Q-learning, they are the reward and the error convergence that is shown in Fig. 6.3 and Fig. 6.4, respectively. Those two figures show that our Q-learning based system tries to obtain the optimal reward value in each episode by slowly but surely updating the Q-values for each parameter. Several examples of the behaviors of the robot during learning are also presented in Fig. 6.5.

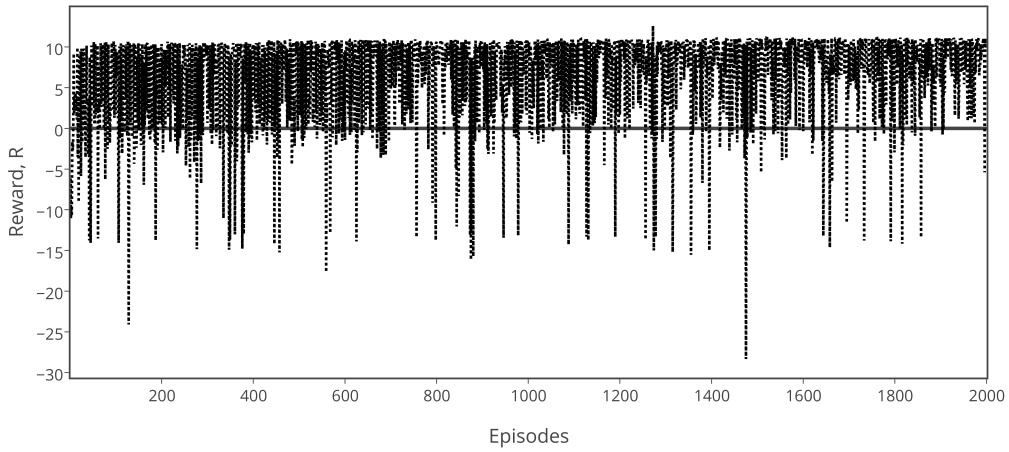


Figure 6.3: The reward value of each episode for training using the first scenario.

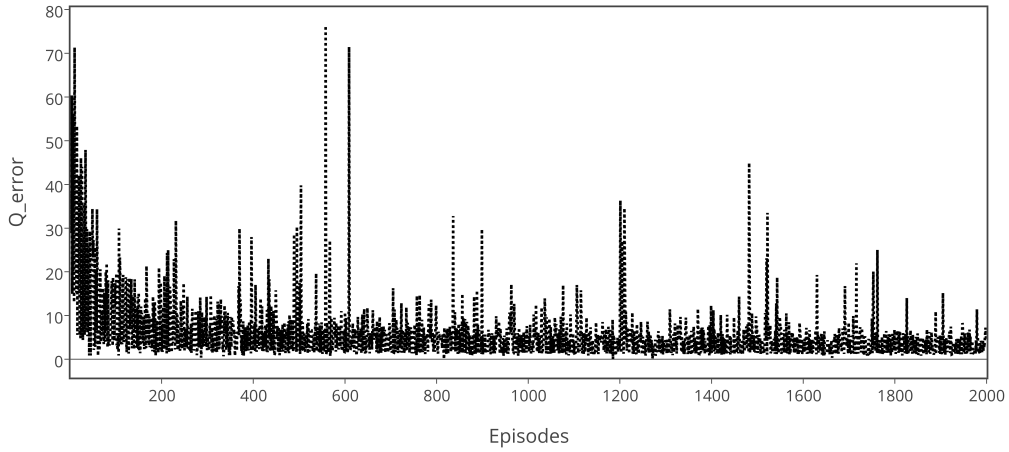


Figure 6.4: Q-learning error convergence for training using the first scenario.

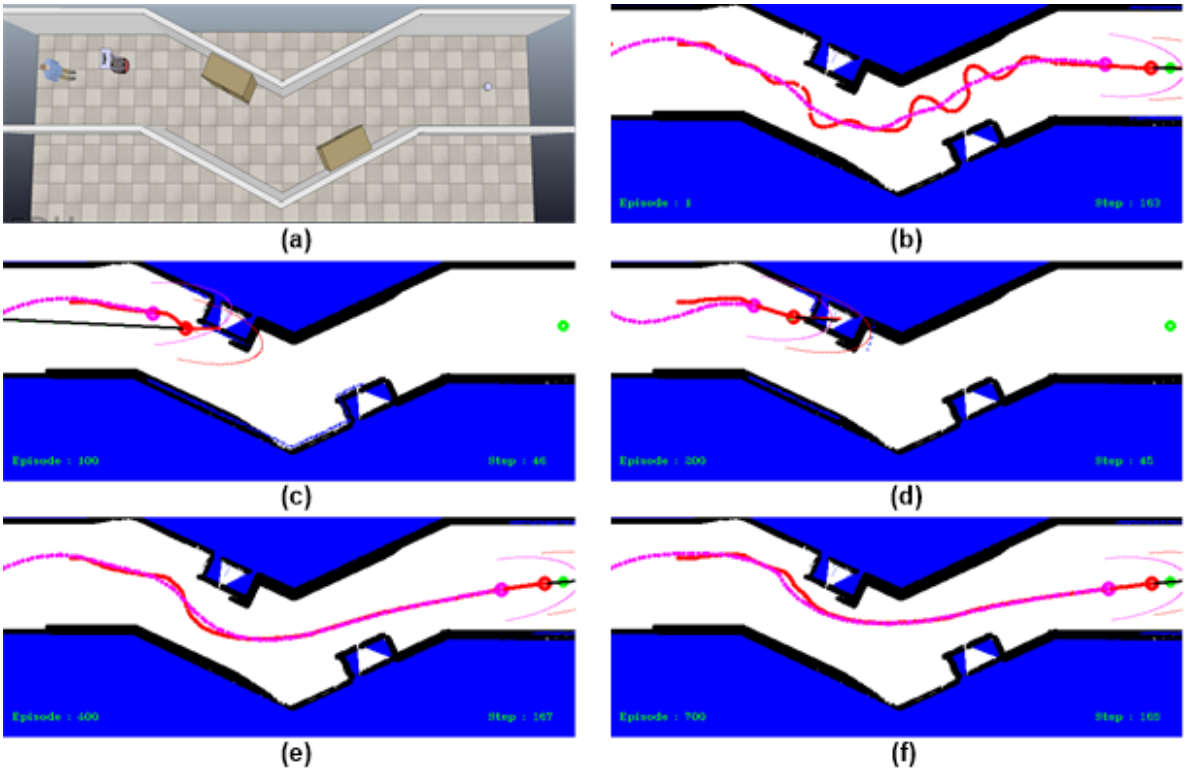
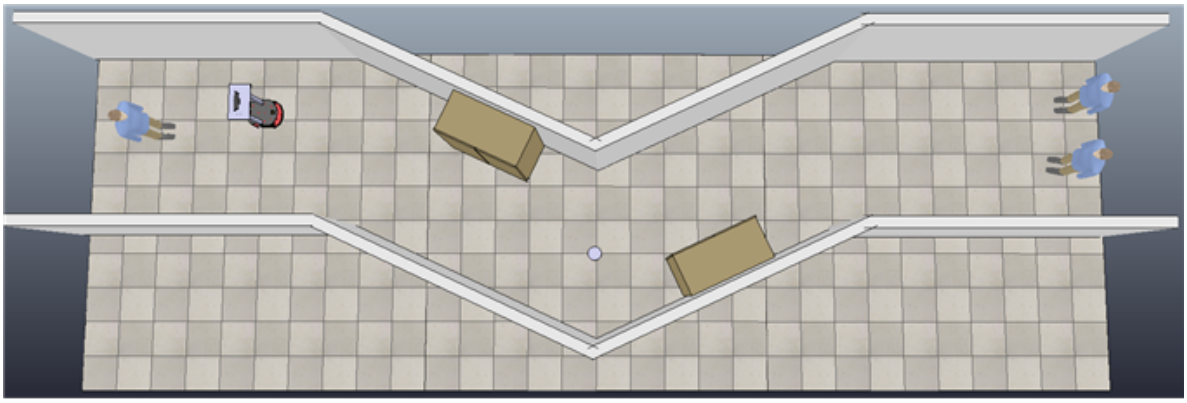


Figure 6.5: Examples of the training behaviors of the robot in indoor scenario (two cupboards). The environment's setting (a), and the progress of the online learning is presented by the last step of each learning episode (b-h). In the first training episode, the robot has no guiding skill at all. It moves in oscillation to reach the goal (b). After 100 episodes, it can start smoother but fails to avoid the cupboard (c). In episode 200, it starts smoother, but also fails to avoid the cupboard (d). The robot can reach the goal smoothly after episode 400 (e). The robot movements become smoother and safer after episode 700 (f). A red circle indicates the robot and a magenta circle indicates the human partner. A green circle is the goal position.

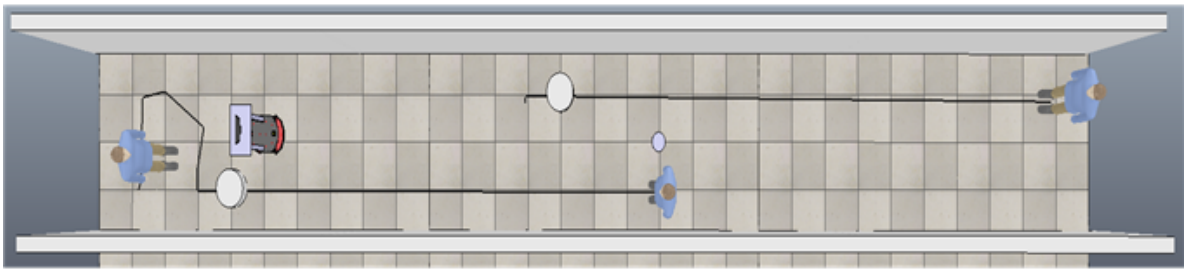
6.5 Testing the Motion Planning and Control using Simulator

6.5.1 Scenarios

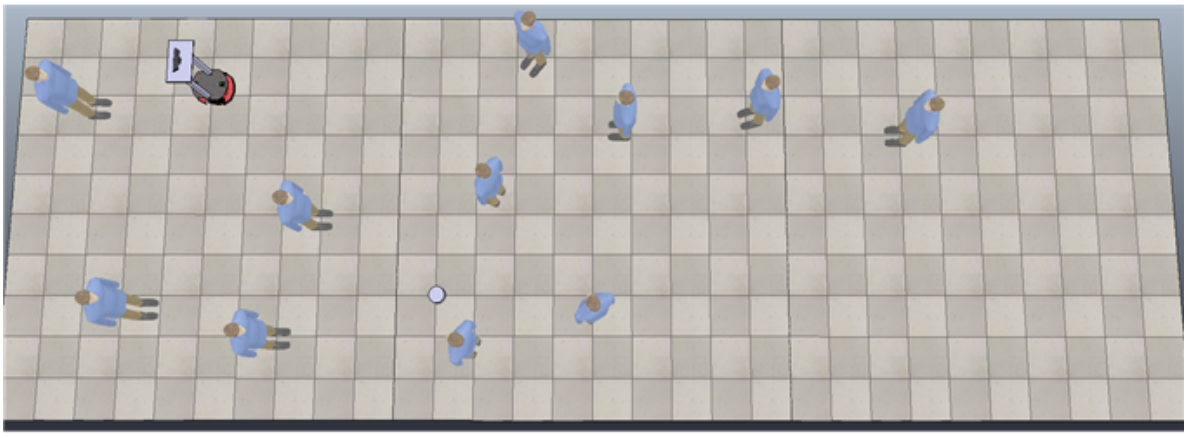
We prepared three scenarios for testing the performance of our motion planning and control algorithm as shown in Fig. 6.6. The first scenario (#1) simulates an indoor environment (a corridor) where the robot will be passed by two freely moving persons when navigating to the goal (bluish cylinder). The second (#2) simulates a narrow indoor environment (a corridor) where the robot will be passed by two persons with fixed trajectories when navigating to the goal. The third (#3) simulates an outdoor environment (the pedestrian's sidewalk) where all persons can move freely. All scenarios are repeated 30 times to see the actual performances of our algorithm. Under such kind of simulations, we expect our robot motion control and behavior will be more robust since the simulated humans cannot be modeled perfectly and frequently interfere our robot, for example,



(a)



(b)



(c)

Figure 6.6: Examples of the scenario for testing: (a) Scenario#1; an indoor environment (a corridor) where the robot will be passed by two freely moving persons when navigating to the goal (bluish cylinder), (b) Scenario#2; a narrow indoor environment (a corridor) where the robot will be passed by two persons with fixed trajectories when navigating to the goal, and (c) an outdoor environment (the pedestrian's sidewalk) where all persons can move freely.

blocking the robot motion, do not care of the robot, or hitting the robot. However, in a real situation, peoples will naturally avoid the robot when they notice the existing of a robot which closes to them.

We also prepared a simple scenario to compare the performance of our QL-SFGM under three schemes: (1) QL-SFGM, (2) QL-SFGM + fixed PID ($k_p = 0.15$, $k_i = 0.025$, and $k_d = 0.5$), and (3) QL-SFGM + optimized PID. In this scenario, we want to show how safe and smooth our

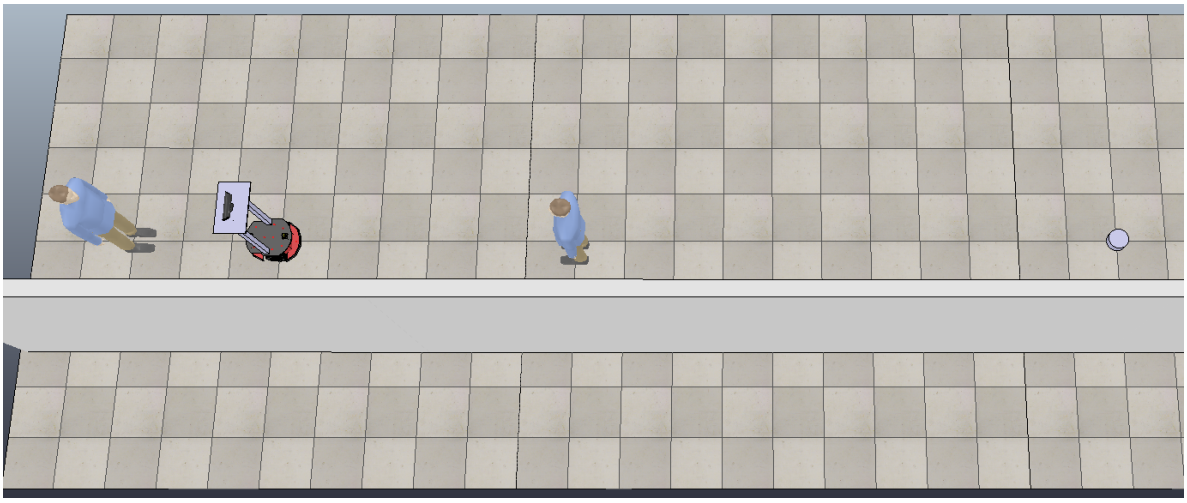


Figure 6.7: The situation that should be done by the robot in order to compare the performance of three QL-SFGM schemes: (1) QL-SFGM, (2) QL-SFGM + fixed PID, and (3) QL-SFGM + optimized PID. The wall is used to ascertain the robot take the same avoidance direction so that the trajectory can be compared visually.

QL-SFGM schemes to respond a static person. Fig. 6.7 shows the situation that should be done by our robot.

6.5.2 Determining Smoothness and Safety

A smoothness of the robot motions is measured using a mean and a standard deviation of the robot altered-headings along its movements toward the goal. A simple moving difference filter is applied on the robot's heading with a kernel $[-1, 0, 1]$ to count how many the robot makes sudden unexpected movements with an angle more than 30^o from its current heading direction. Safety can be measured using a percentage of successful task completion without threatening humans or the robot.

6.5.3 Experiments and Evaluations

Fig. 6.8 shows the simulation results on scenario #1 where our robot attempts to guide the human partner from the starting point towards the goal in a simulated corridor environment. The

Table 6.3: The smoothness of motions

Method	**Mean±Std. Dev. (times)**		
	#1	**#2**	**#3**
QL-SFGM	4.97±3.69	7.67±4.18	12.73±8.94
QL-SFGM + fixed PID	4.77±4.45	6.73±7.44	12.13±9.80
QL-SFGM + optimized PID	4.53±4.24	5.60±4.29	10.90±6.77

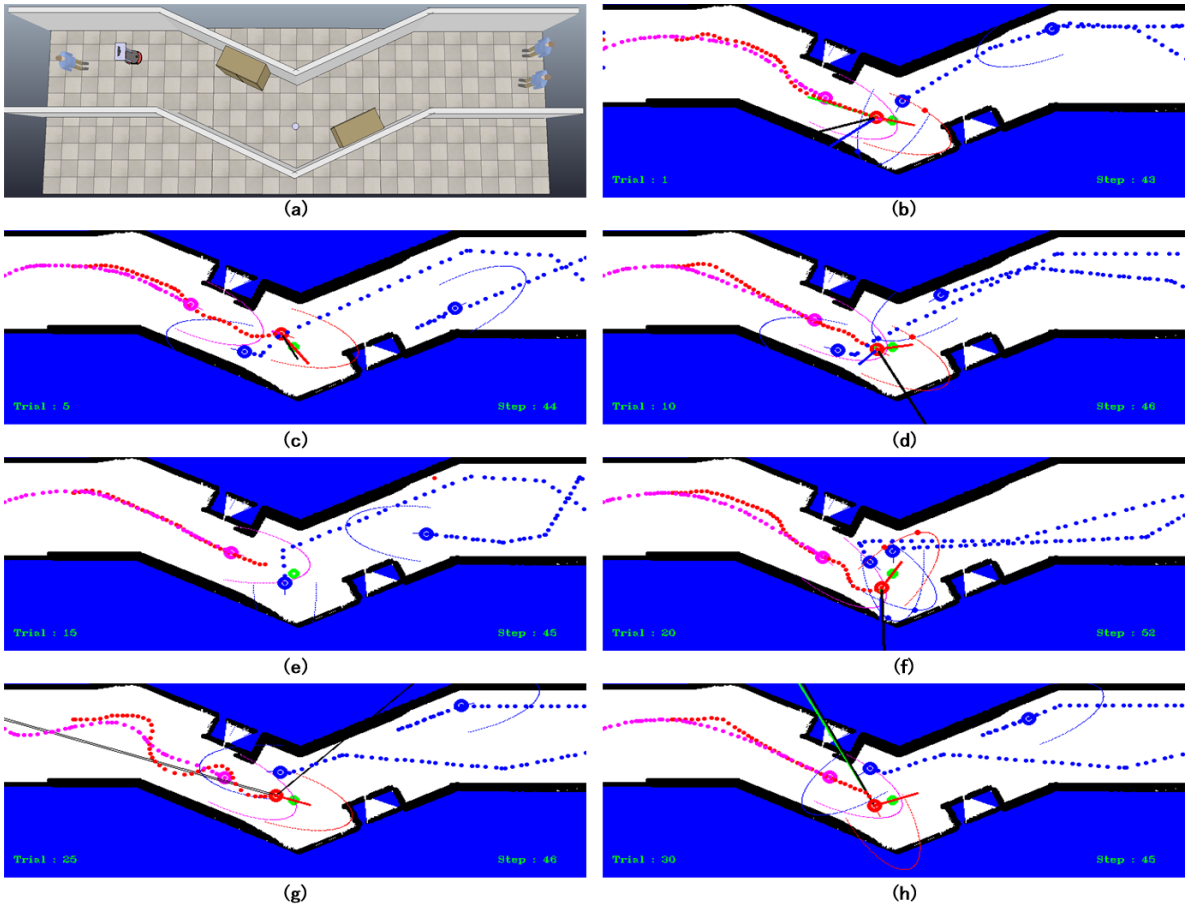


Figure 6.8: Examples of the testing behaviors of the robot when applying QL-SFGM + optimized PID in scenario #1. The environment's setting (a), our guide robot motions toward the goal and its behaviors when responding the other person approaching (b-h). Our robot successfully reaches the goal with a variety of behaviors in (b-d and f-h), while it fails to reach the goal due to a collision with the person in (e). Blue circles indicates the dynamic obstacles, a red circle indicates the robot, and a magenta circle indicates the human partner. A green circle is the goal position. Each figure shows the trajectories produce by the robot, partner and other people, that are shown by colored points with respect to each object.

smoothness and safety are shown in Table. 6.3 and Table. 6.4. Robot completed 21 of 30 trials by applying QL-SFGM + optimized PID (70.0%). It failed to complete nine trials because the robot hit the wall when avoiding a person who demonstrates an over-reactive behavior which very

Table 6.4: The comparison results of the successful trials (%) and the average travel time (sec)

Method	**Successful Trials (%)**			**Avg. Time (sec)**		
	#1	**#2**	**#3**	**#1**	**#2**	**#3**
QL-SFGM	60.00	40.00	56.67	15.05	14.19	8.06
QL-SFGM + fixed PID	60.00	53.33	63.33	18.74	16.61	10.22
QL-SFGM + optimized PID	70.00	66.67	63.33	16.73	15.60	8.59

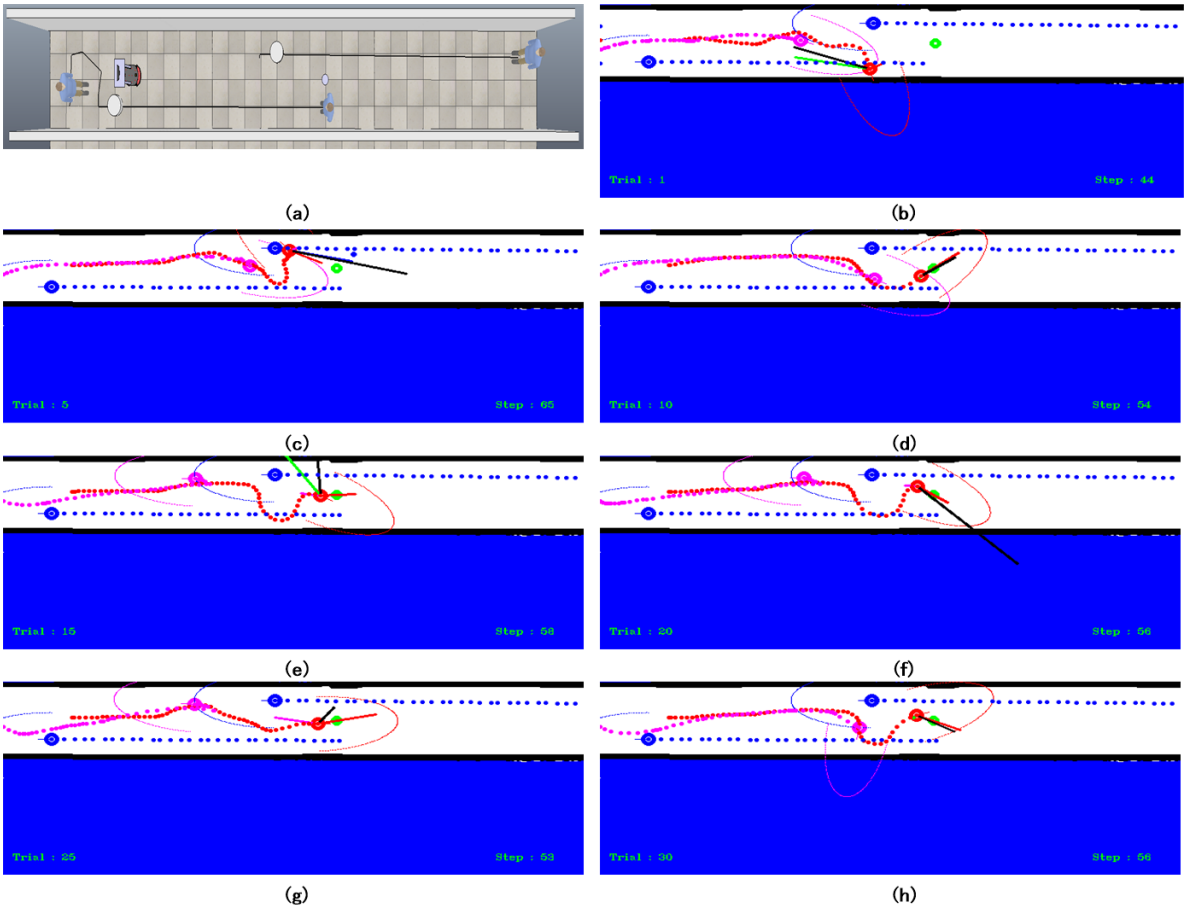


Figure 6.9: Examples of the testing behaviors of the robot when applying QL-SFGM + optimized PID in scenario #2. The environment's setting (a), our guide robot motions toward the goal and its behaviors when passed by the other person (b-h). Our robot successfully reaches the goal with a variety of behaviors in (d-h), while it fails to reach the goal due to a collision with the wall (b) and the other person (c), respectively. Blue circles indicates the dynamic obstacles, a red circle indicates the robot, and a magenta circle indicates the human partner. A green circle is the goal position. Each figure shows the trajectories produce by the robot, partner and other people, that are shown by colored points with respect to each object.

disturbing our robot. The other failures caused by the robot were hit by one of those people.

When applying QL-SFGM + fixed PID, the performance of our robot drastically drops to only 60%. This happens due to our robot sometimes perform an oscillatory behavior when attempting to avoid static obstacles. These unstable behaviors make our robot frequently hit by one of those people or hit the wall. As the consequence, this kind of behavior makes the robot slows down its speed. Therefore, it needs more time to reach the goal position.

When applying QL-SFGM only, our robot achieves a successful rate 60%. This happens due to our robot frequently perform an oscillatory behavior when attempting to avoid static and dynamic obstacles. These unstable behaviors make our robot frequently hit by one of those people or hit the wall. However, this kind of behavior surprisingly makes the robot frequently reach the

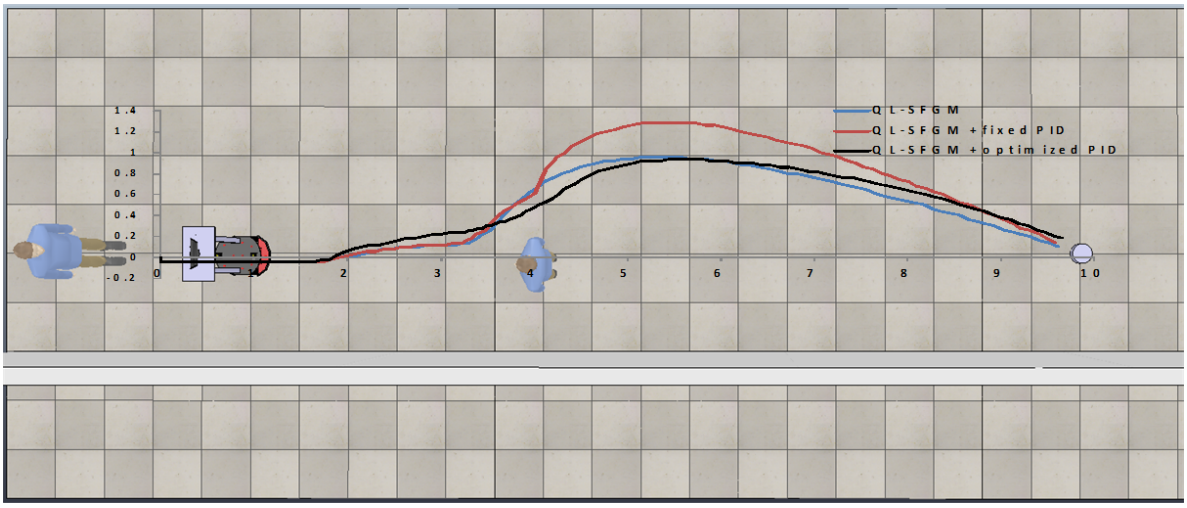


Figure 6.10: The comparison result of the performance of three QL-SFGM schemes: (1) QL-SFGM (blue), (2) QL-SFGM + fixed PID (red), and (3) QL-SFGM + optimized PID (black).

goal position faster than using PID.

Fig. 6.9 shows the simulation results on scenario #2 where the main challenge is the robot should guide the partner in a narrower corridor with two apathetic people (the people just following the trajectory defined in advanced and have no intention to avoid the robot). The performances of our robot are shown in Table. 6.3 and Table. 6.4. Robot completed 20 of 30 trials by applying QL-SFGM + optimized PID (66.67%). It failed to complete 10 trials because it was frequently hit by one of those people.

When applying QL-SFGM + fixed PID, the performance of our robot drastically drops to only 53.33%. This happens due to our robot sometimes perform an oscillatory behavior when attempting to avoid static obstacles. These unstable behaviors make our robot frequently hit by one of those people or hit the wall. Utilizing QL-SFGM only makes the performance of our robot drastically drops to only 40%. This happens due to our robot frequently perform an oscillatory behavior when attempting to avoid static and dynamic obstacles. These unstable behaviors make our robot frequently hit by one of those people or hit the wall.

Finally, we show how the robot responds an obstacle by using each scheme. Fig. 6.10 shows the trajectories of our robot for each scheme. Although somewhat difficult to distinguish, but it can be seen that using optimized PID produce a relatively smaller deviation than using a fixed PID and without PID. This becomes the advantage of our QL-SFGM + optimized PID where robot's response becomes smoother but still effective when facing obstacles.

6.5.4 Computation Time

A faster computation time is required to apply the algorithm on a real system. Our simulation on a desktop PC equipped with 3.6 GHz Intel Core i7 processor and 16 GB of RAM shows

that the computer needs 0.3 Sec in average to calculate the algorithm, to execute the motion control, and to render the video display in V-Rep. The video rendering in V-Rep contributes the highest computation time, especially when handling many objects in a scene.

6.6 Robot Implementation in Real Environments

So far, we have validated our motion planner and control using a realistic 3D simulator. In this section, we extend our work by implementing our motion planner and control for the real robot in the real environment. We evaluate and validate several modules of the task which composing our system as shown in Fig. 6.11. Under Robotic Technology (RT) Component, sensor modules, partner identification module, partner behavior recognition module, QL-SFGM based motion and control module, people detection module, and mobile robot controller module are separately packaged in different components. This separation has several advantages such as localizing errors become easier, each module can be developed by another developer, and optimizing the use of computer processors.

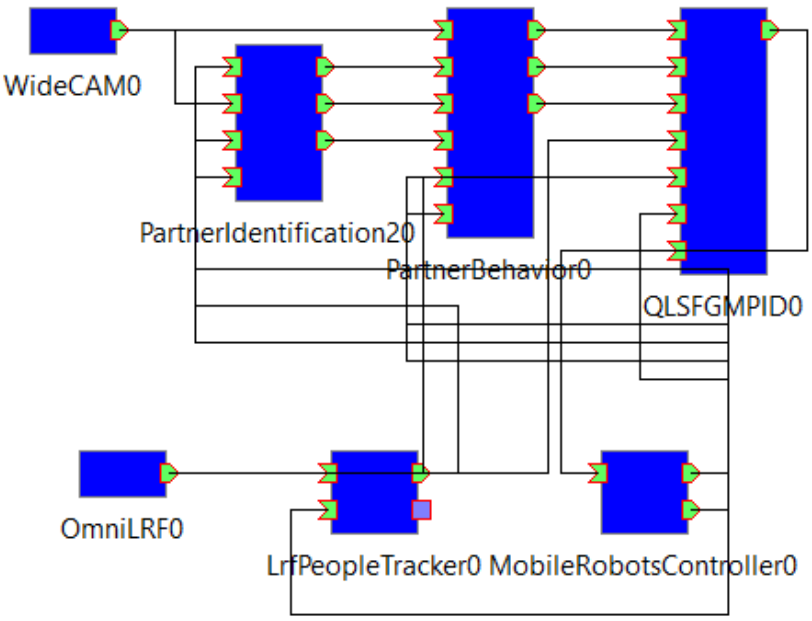


Figure 6.11: Our guide robot system that is managed using Open Robotic Technology Middleware (OpenRTM).

6.6.1 Human Partner Identification

6.6.1.1 Static System

In this section, we evaluate and validate our human partner identification module in a static camera condition. In Fig. 6.12, a candidate of human partner is first detected using a camera which implements Haar-based upper body detection. From the detected area, face and clothing color features are extracted. The image of each feature is stored for an identification process purpose. When the human partner has been determined, a tracker is implemented to the upper body area to help detection and identification. A disturbance from another person is also evaluated. Our system can retain the human partner identity even another person with the same color of clothing blocking the partner for a while.

6.6.1.2 Moving Robot

We then evaluate and validate our human partner identification module for moving the robot. A back-facing camera is mounted on the top of the robot. Fig. 6.13 shows the performance of our human partner identification module when the robot does its job. When the robot moves, the performance of our human partner identification module decreases. The human upper body detector

Figure 6.12: Examples of person identification in a static camera condition (from the camera view): (a-b) a person who is identified as the partner, (c-f) another person tries to disturb the system. Our system consistently detect the previous person as the partner.

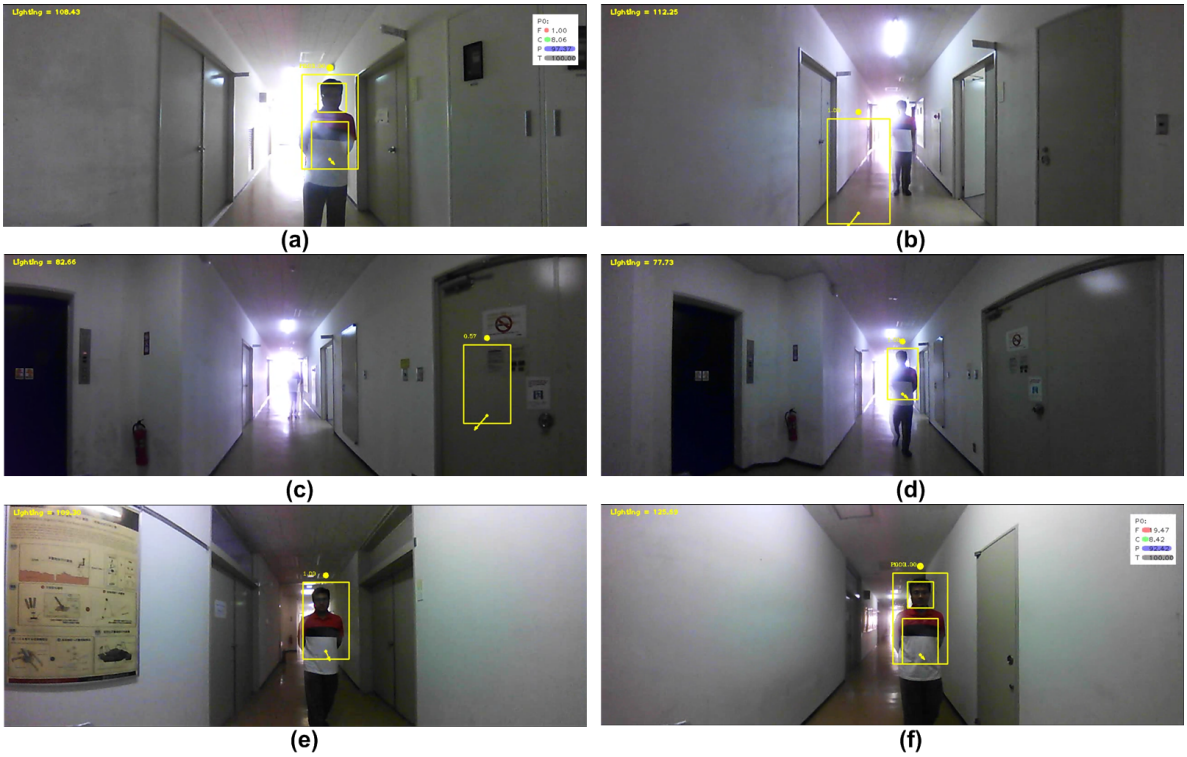


Figure 6.13: Examples of person identification in a moving condition (from the camera view): our system can detect and track the targeted partner. However, in some cases (b,c), the person identification module fails to locate the targeted partner position due to the robot rotations and a harsh illumination condition.

and tracker frequently failed to locate the partner real position due to robot's movements. Even though the robot rotates only a few, however, the tracker speed cannot balance this rotation.

6.6.2 Human Partner Behavior Observation

We observe the partner's behavior, not only from his/her position, but also from his/her head orientation. Fig. 6.14 shows the performance of our head orientation estimation system to define partner attention. This feature is basically performed better in a good illumination condition, but significantly decreasing in a severe illumination condition.

6.6.3 Robot Guiding Behavior

To evaluate our robot guiding behaviors, we conducted experiments in the corridor of D-building 2nd floor, starting from in front of the D4-202 laboratory until a goal position in front of the D3-202 laboratory (around 27.5 meters). We run our robot under a variety illumination conditions to test its whole performance.

Fig. 6.15 shows the examples of our experimental results. In this experiment, our partner

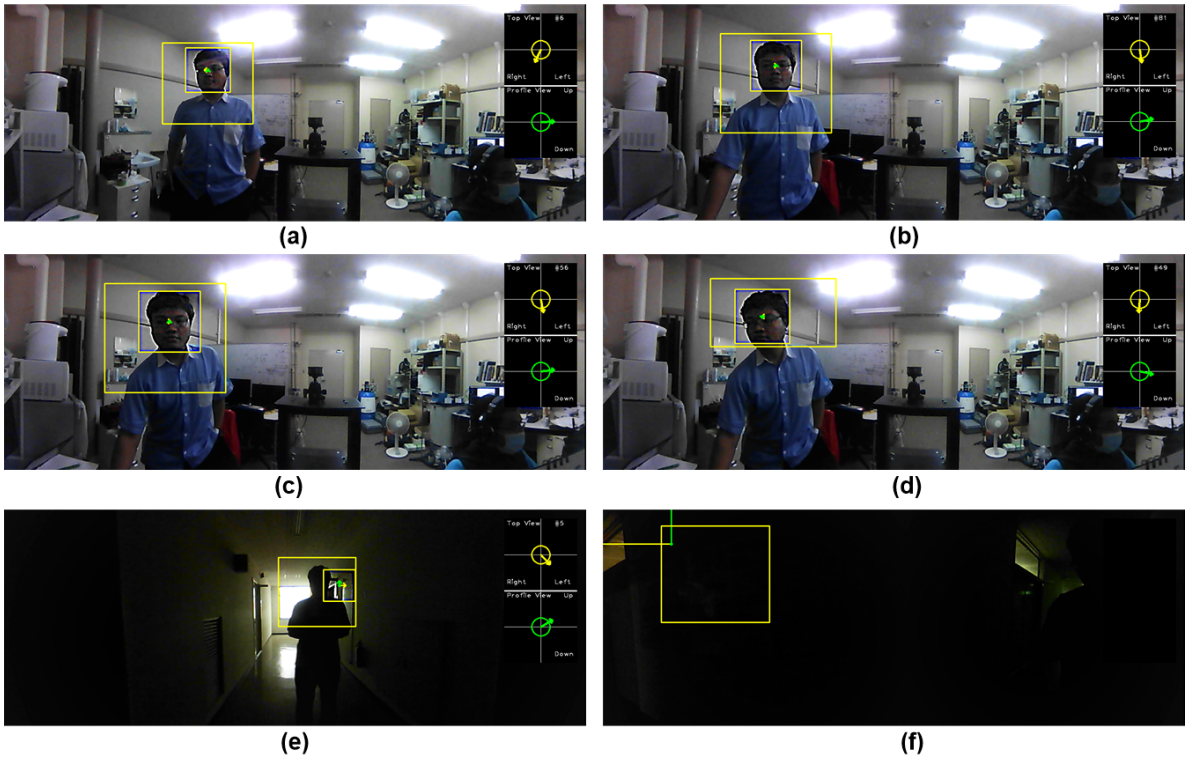


Figure 6.14: Examples of partner behavior observation (from the camera view). Small window on the right side shows the pan and tilt orientation of the partner head. The partner's head orientation can be estimated in (a-d), but failed in (e,f).

identification module can detect and track the position of the human partner. The motion direction of the human partner can also be estimated. Our motion planner and control module successfully navigates our robot to guide the human partner to reach the goal safely.

Fig. 6.16 shows the trajectories of the robot and the human partner in the same experiment as shown in Fig. 6.16. Our motion planner and control module successfully navigates our robot to guide the human partner to reach the goal safely. In this experiment, we evaluate the effect of the coordination distance between the robot and the partner. In the middle of the guide, partner occasionally stopping to simulate if his awareness is decreasing. The robot was able to stop and waiting. Robot resumes guiding process when the partner returned to the coordination zone.

Fig. 6.17 shows another experimental result of our guide robot behavior. Several frames sequentially show the robot behavior when guiding the human partner. In this experiment, we show how the robot safely avoids the other persons, responds the partner when postponing the task, acknowledges continue signal from the partner, and safely reach the goal.

6.6.4 The Importance of the Coordination Task

We compare the behavior performances of our guide robot when utilizing coordination task and without coordination task using our Q-learning based social force motion control and behavior

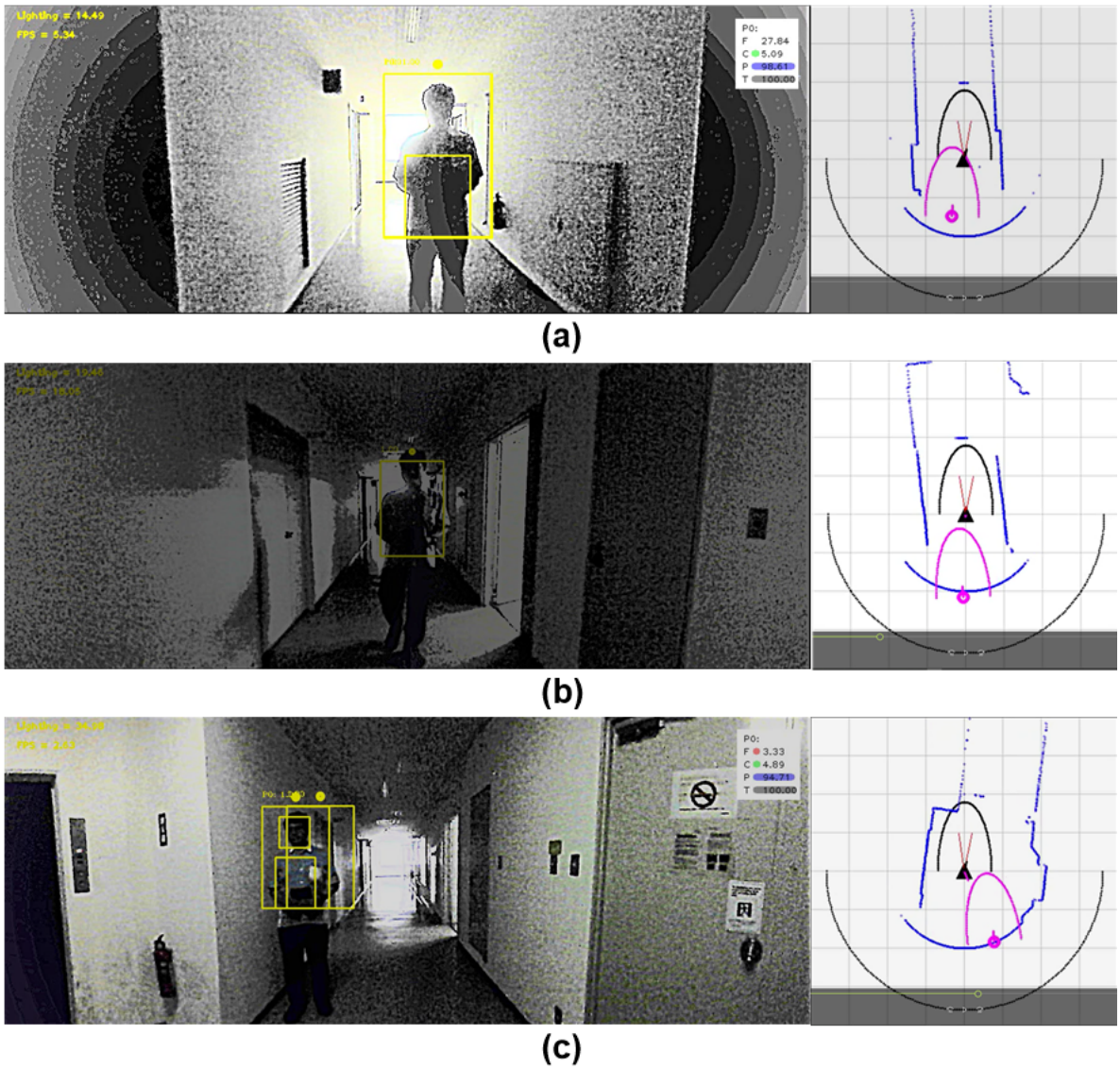


Figure 6.15: Examples of guiding behaviors which were demonstrated by our robot.

model. We compare the robot behavior performances based on two cases: (1) human-robot relative distance over time, and (2) robot speed against human partner distance. The first case is used to show the robot response when the human partner attempts to perform stop and go actions, and the second case is used to show the robot response when the human partner follows the robot in different distances.

Fig. 6.18 shows the comparison results of both models. First, we model the guiding behavior using our QL-SFGM method (shown by the graph in blue). In the beginning, the human partner follows the guide robot from behind with a particular distance. Then, the human partner stops (shown by number 1 within a blue circle). The robot still continues its task up to a critical distance (shown by number 2 within a blue circle) and then it stops. When the human partner continues his task (shown by number 3 within a blue circle), the robot immediately responses by continuing its

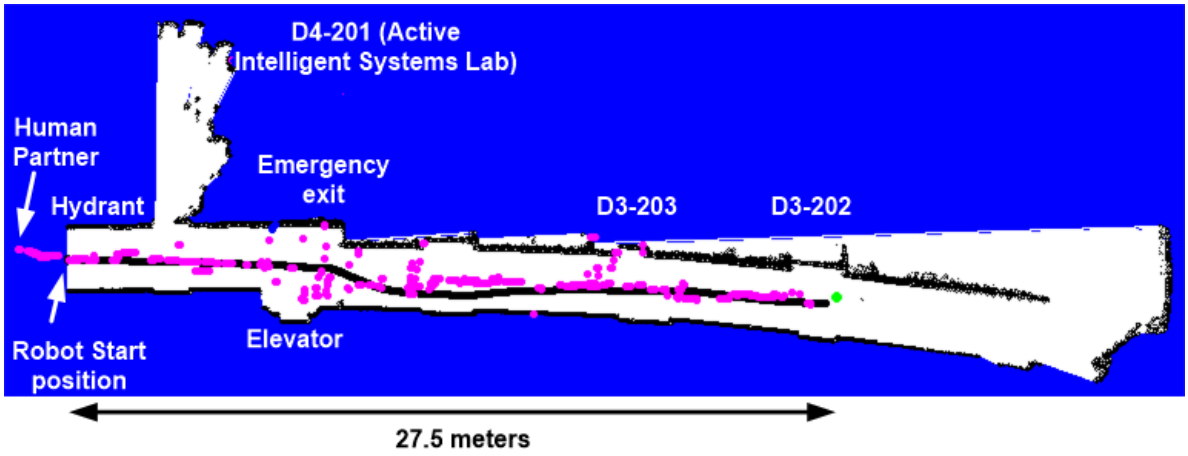


Figure 6.16: Example of the robot and the human partner trajectories in a global map. The robot's trajectory is represented using black color, the human partner's trajectory is represented using magenta color, and green circle for goal position. The detected locations of the human partner are oscillated in some places due to instability of our partner identification module to response robot's camera rotations.

task until reaching the goal position (shown by number 4 within a blue circle).

Second, we model the guiding behavior using our Q-learning based Social Force Navigation Model (QL-SFNM) only and ignoring the human partner actions (shown by the graph in orange). In the beginning, the human partner follows the guide robot also from behind with a particular distance. Then, the human partner stops (shown by number 1 within an orange circle). The robot continues its task (shown by number 2a within an orange circle) until it cannot detect and recognize its target partner due to too far distance (shown by number 2b within an orange circle). Then, the human partner continues to follow the robot and he can be detected from distance ± 7.5 meters (shown by number 3 within an orange circle). Since the robot only executes its task by ignoring the human partner actions, then it can reach the goal position in less number of frames compared to considering the coordination with the human partner (shown by number 4 within an orange circle).

Fig. 6.19 shows the comparison results when the human partner attempts to adjust his walking speed when following the robot. By activating the coordination task and the human partner follows the robot with a constant distance no more than 1.5 meters, the robot attempts to speed up its motion. This is caused by an additional repulsive force generated by the human partner to the robot, so that the robot increases its speed around 0.61 m/s in average. On the other hand, when the human partner follows the robot with a constant distance more than 1.5 meters, the robot attempts to slow down its motion. This is caused by an attractive force generated by the human partner to the robot, so that the robot decreases its speed around 0.51 m/s in average. Without a coordination task, the robot does not care to the human partner state (position and distance) by still executing its motion towards the goal position by a relative constant speed around 0.54 m/s in average. Some spikes (robot achieves a maximum speed, 1 m/s) occur in all experiments when the robot moving

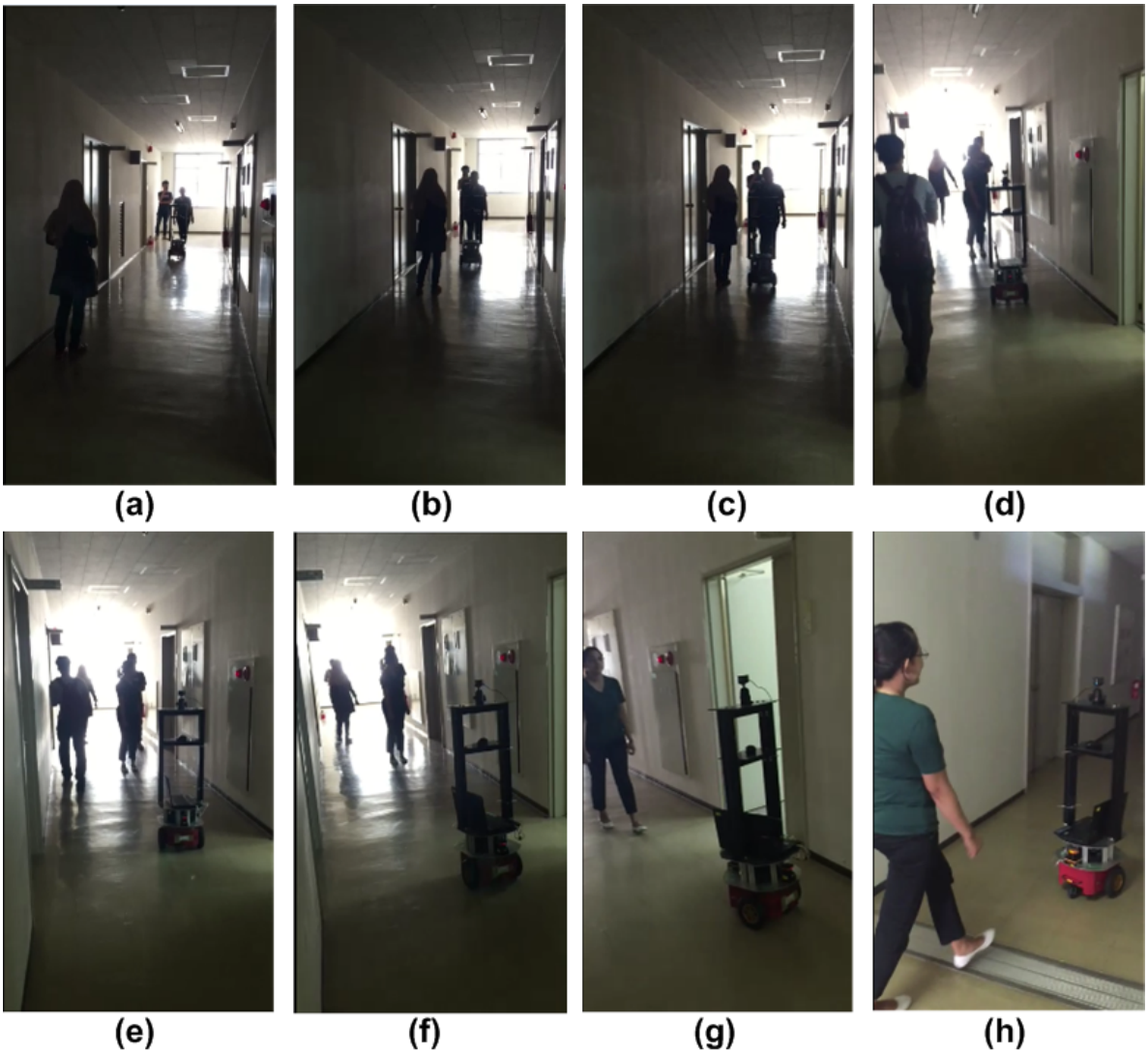


Figure 6.17: Examples of human-robot coordination task in a guiding task (from a handheld camera view): (a) robot starts its task from starting point, (b) another person approaches to the robot, (c) the robot is avoiding the approaching person, (d and e) the human partner postpones her task and another person passes by the robot, (f) the robot is waiting the partner, (g) the partner continues her task and the robot responds her, and (h) the robot successfully passes a static object (wall) and continues its task towards the goal.

too close to the wall and attempts to avoid it by receiving more force from the wall. In most cases, a robot that runs without coordination with the human partner can reach the goal faster than one that considers coordination.

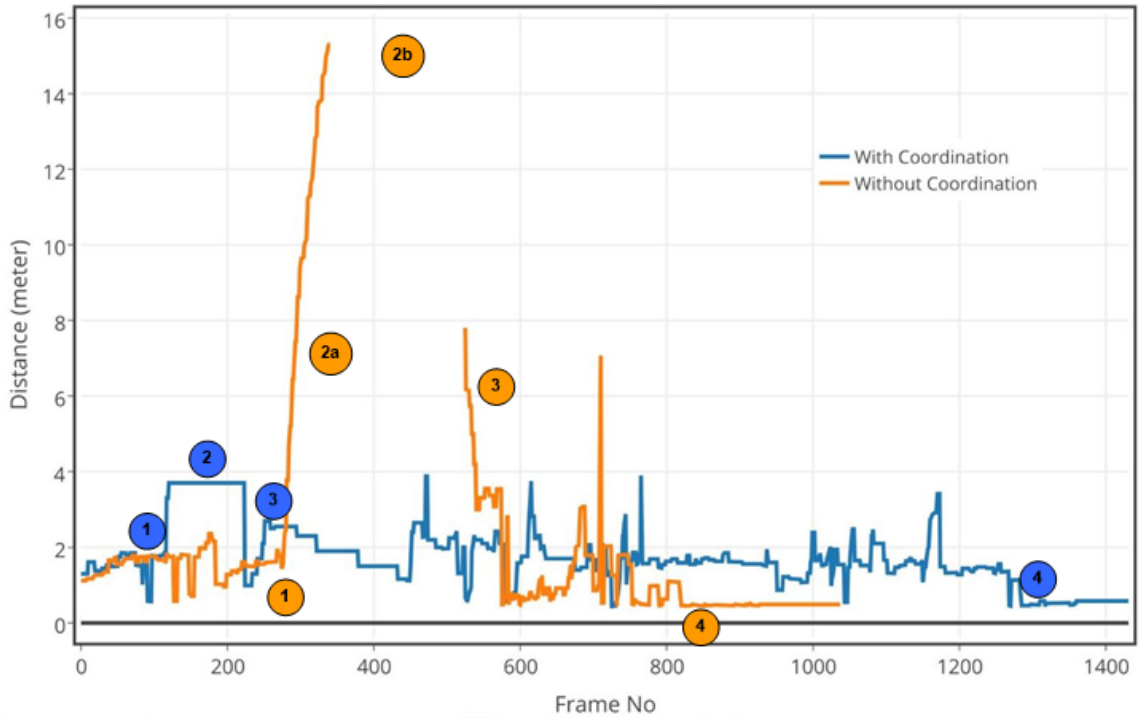


Figure 6.18: The importance of the coordination task which is shown using a relationship between human-robot distance vs time.

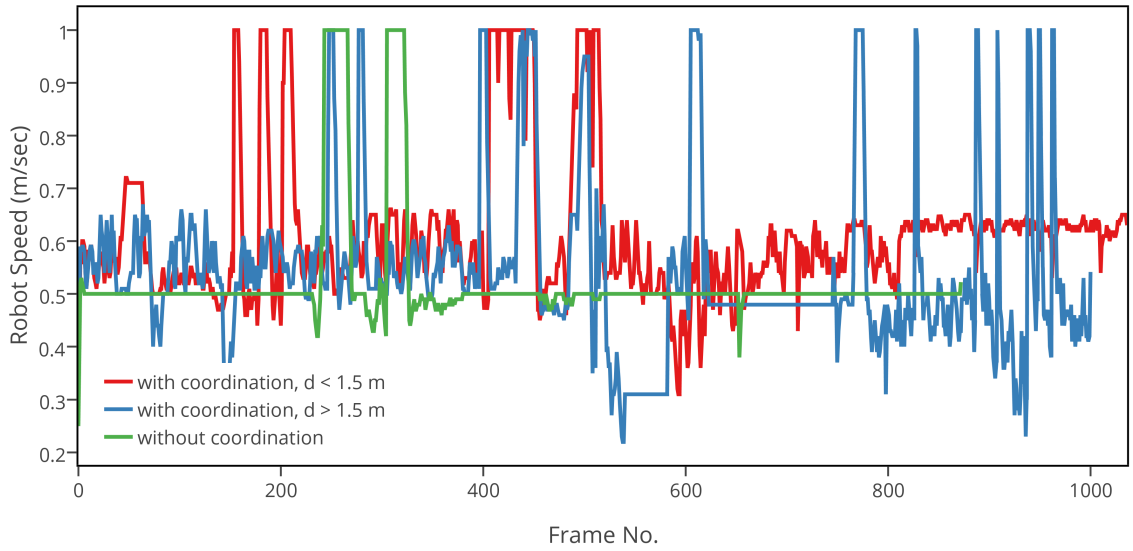


Figure 6.19: The importance of the coordination task which is shown using a relationship between robot speed vs time for every performed human partner distance (d).

6.6.5 Preparation of Survey based Evaluation

6.6.5.1 Respondents

A research topic about human-robot interaction is very interesting and will be more challenging when involving real human opinion to evaluate the objectives. To deal with this requirement,

Table 6.5: The summary of our survey respondents.

Department	Grade			
	B	**M**	**D**	**RS**
Mechanical Engineering	0	2	2	1
Electrical and Electronic Information Engineering	0	1	2	0
Computer Science and Engineering	0	6	3	0
Environment and Biotechnology	1	3	1	0
Architecture and Urban System Engineering	0	4	3	2
Total	**1**	**16**	**11**	**3**

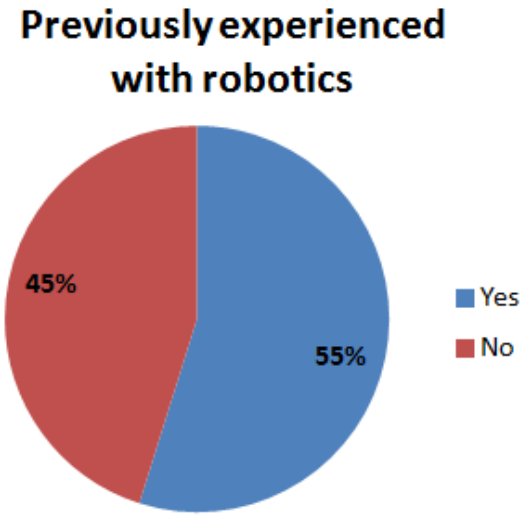


Figure 6.20: The number of respondents with and without experience with robotics.

we provide a combination of experiment based survey, which is participated by 31 people. Most of them are the students on our campus. We summarized the respondents as shown in Table. 6.5. From 31 respondents who joined with our experiment, 55% of them are previously familiar with robotics and the remaining 45% are not familiar with robotics as shown in Fig. 6.20.

6.6.5.2 Survey Questions

We simply design our survey using general questions so that we expect that the respondents can fill it without any difficulties. Our survey form is shown as follows.

Survey Form 1. *A Survey on A Socially Aware Guide Robot with Illumination Invariant Face Recognition and Self Learning of Guiding Behavior*

1. In your opinion, how is the behavior of the robot when it guides you ?

(a) properly
(b) gentle
(c) too reactive
(d) careless
2. Do you feel comfortable when guided by our robot ?
(a) very comfortable
(b) comfortable
(c) ordinary
(d) uncomfortable
(e) very uncomfortable
3. Do you think that the robot understands you when you do something or don't care to the robot ?
(a) yes
(b) no
4. What is the robot doing when you stop and postpone the guiding task ?
(a) stop and waiting for me
(b) don't care and continue its task
(c) confusion
5. When the robot guiding you, how the robot responds to other person(s) that passed by ?
(a) safely avoid the person
(b) hit the person
(c) stop and do not know what should do
6. When the robot guiding you, how the robot responds to static object(s) blocking the way ?
(a) safely avoid the object
(b) hit the object
(c) stop and do not know what should do
7. If you have any comments, please feel free to write it down here.
.....
.....
.....

Are you previously familiar with robotics ? Yes / No
Dept. : .....
Grade : .....

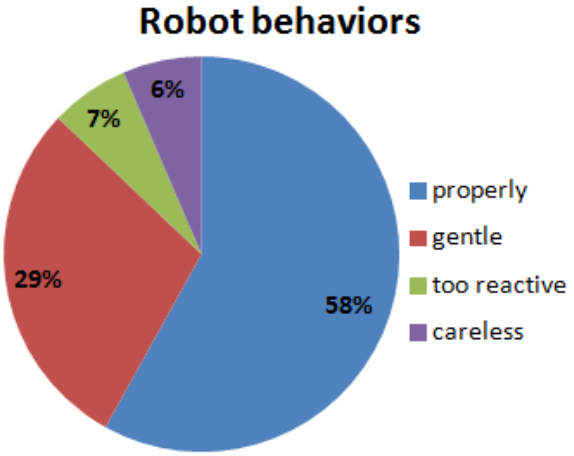


Figure 6.21: The robot behaviors based on respondent's opinions.

6.6.6 Smoothness Evaluation

We conduct a survey based and real experiments based evaluations for measuring the smoothness of our guide robot movements. This smoothness evaluation is important to see how smooth our guide robot to move in the environments when it executes the motion planning and control actions. Before filling the survey questions, each respondent is asked to join with our experiment. The respondent is asked to act as a guided partner and follows the robot from behind. We ask them to just relax when following the robot towards the destination. After finishing the experiment, they are asked to fill out our survey form. In the form, we ask their subjective opinion about our robot behavior when guiding them. Fig. 6.21 shows the survey result with respect to the smoothness of our guide robot behavior. 87% of total respondents said that our guide robot performs properly and gentle when guiding them, and only 13% of total respondents said that our guide robot performs too reactive and sometimes careless.

Our real experiments based evaluations are conducted by running the robot to complete 20 times of trials. We use the same method as explained in simulation section. We use the same kernel filter to measure the number of alternate headings of our robot when guiding the human partner from starting point to the destination. As the result, our robot can achieve a very good result by making alternate headings only 0.3 ± 0.47 times. Based on the result of the survey and real experiments, we can say that our robot can perceive and respond to each circumstance as our expectation.

6.6.7 Safety Evaluation

Since safety also became one of the targets of our evaluation objectives, we also conduct a survey based and real experiments based evaluations for measuring the safety of our guide robot

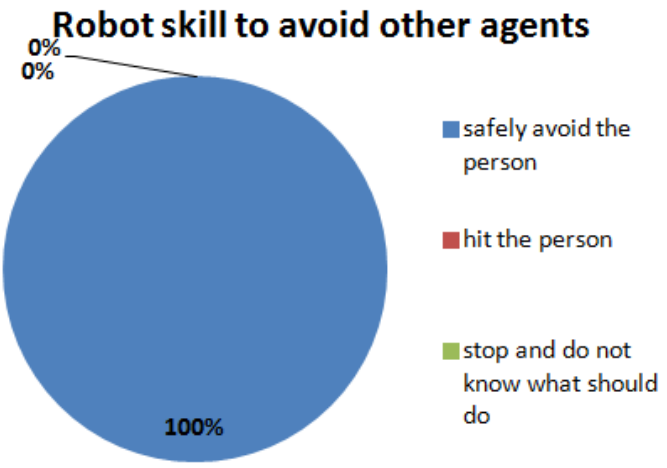


Figure 6.22: The robot skill to avoid dynamic obstacles based on respondent's observations.

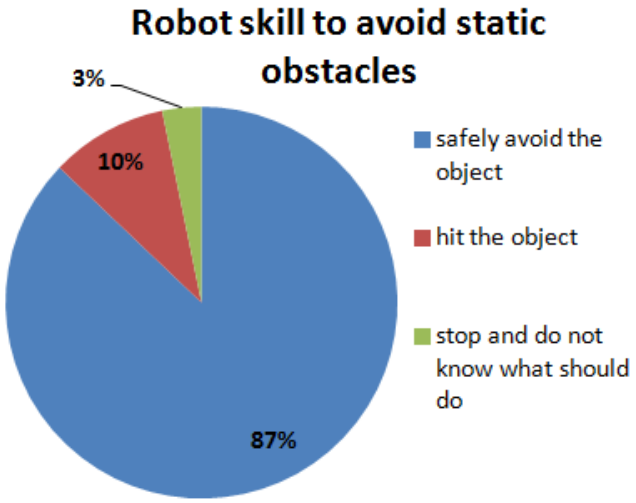


Figure 6.23: The robot skill to avoid static obstacles based on respondent's observations.

movements. This safety evaluation is important to see how safe our guide robot to plan the motion and control its movements among human social environments. The survey questions are divided based on two different obstacles: dynamic obstacle and static obstacle. When the respondent acts as a guided partner and follows the robot from behind, we ask them to observe the robot response when passing by other people (as a dynamic obstacle) and when facing a static obstacle encounters its motion, for example walls, fire extinguisher. After finishing the experiment, they are asked to fill out our survey form. In the form, we ask their subjective observation about our robot safety when guiding them. Fig. 6.22 shows the survey result with respect to the safety of our guide robot movements with respect to the dynamic obstacles. 100% of total respondents said that our guide robot can safely avoid other people who pass by. Fig. 6.23 shows the survey result with

respect to the safety of our guide robot movements with respect to the static obstacles. 87% of total respondents said that our guide robot can safely avoid static objects, 10% of them said that our robot hit the object (wall), and 3% of them said that our robot just stops and does not know what should do.

Our real experiments based evaluations are conducted by running the robot to complete 20 times of trials. We assume our robot can safely guide the human partner when it can complete the task by arriving at the destination. As the real experimental results, our robot successfully guides the human partner and arrives at the destination in all trials or achieving 100% of successful trials. Based on the result of the survey and real experiments, we can say that our robot can safely guide the human partner as our expectation.

6.6.8 Comfort Evaluation

The human partner comfort is one of our main objectives which differentiates our socially aware guide robot implementation with the conventional robot implementation. To evaluate this objective, we also conduct a survey based and real experiments based evaluations. It is very important for us to build a guide robot which can guarantee the human partner always feel comfortable. The survey questions are divided based on three different things: the robot responsibility to the partner, the robot understanding, and the human partner feelings when guiding by our robot. When the respondent acts as a guided partner and follows the robot from behind, we ask them to observe the robot response when the human partner postpones their task due to their interest in something, for example paintings, announcements, photos, etc. At the same time, we also ask them to evaluate the robot understanding to them, and their feelings whether they are comfortable or not when guiding by our robot.

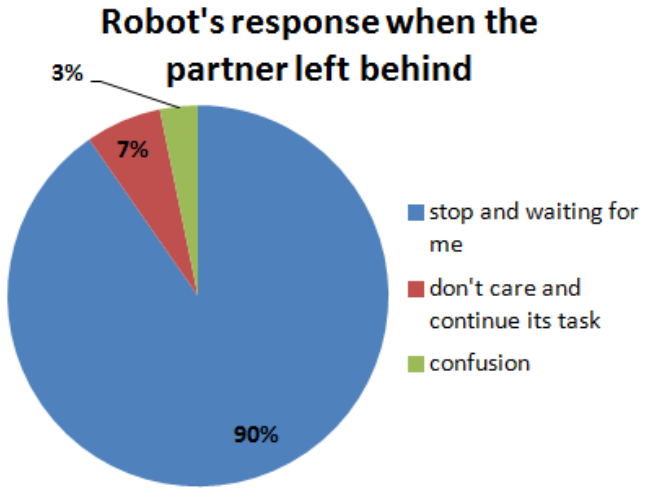


Figure 6.24: The robot responsibility to the human partner when the partner postpones the task.

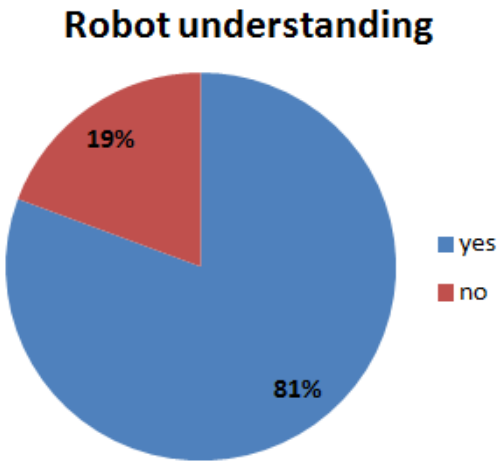


Figure 6.25: The robot understanding to the human partner based on the respondent's opinions.

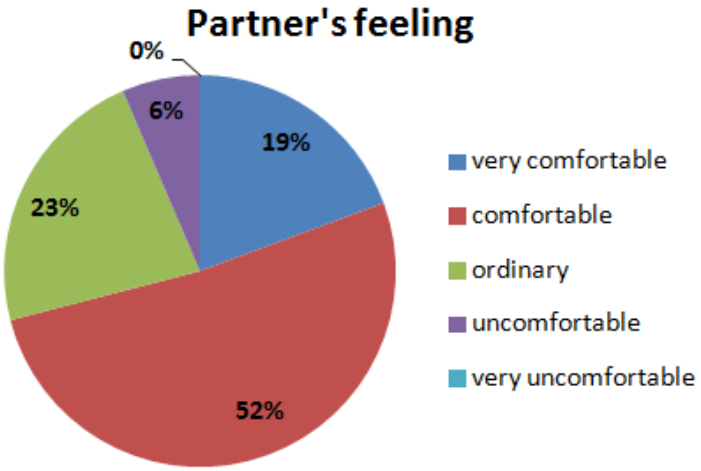


Figure 6.26: The subjective feeling of the human partner when guiding by our guide robot.

After finishing the experiment, they are asked to fill out our survey form. In the form, we ask their subjective observation about our robot responsibility when guiding them. Fig. 6.24 shows the robot responsibility to the human partner. 90% of total respondents agreed that the robot responds their postponed task by waiting for them in a visible distance. 7% of total respondents said that the robot doesn't care about them and still continues its task, and only 3% of them said that the robot is in confusion.

Another evaluation objective is about the robot understanding of the partner. This evaluation is conducted to see the robot response when the human partner is walking at very low speed or very high speed, and still has a correlation with the human partner postpones the task. 81% of total respondents feel that the robot understands to their actions and only 19% of them feel it does not understand them or not really know whether the robot truly understands or not.

Table 6.6: Some examples of the valuable comments from the respondents.

No.	Comments
1	In the moment when the robot hit the static object, I didn't know which direction it would take
2	The guiding behavior is good, but it is little unreliable when robot is in near of wall or person
3	Problem occurred when I'm too close with the robot, it stopped for a while until I stood in the proper distance. But overall it was good and comfort enough
4	More feeling as guidance if the robot is dressed as a robot-like or human-like
5	The distance for simulation is quite short
6	The velocity is proper, it was guiding with good direction and comforting the participant
7	It is better to get more smooth in the movements
8	I felt the robot is too far from me when I stop and look poster
9	The robot did its job properly. I was successfully guided from the start point to the end point. A few problems are: (1) when the space has bad lighting, it become difficult for the robot to detect me and (2) the screen freezes for a few seconds
10	Robot could guide me properly and comfortable
11	Enhancing robot movement in dark corridor because sometime the robot do not move smoothly

The last evaluation objective is the subjective feeling of the human partner with respect to their comfort. 71% of total respondents feel that they are comfortable or very comfortable with our guide robot behaviors and services. 23% of them said that they feel ordinary, and 6% of them feel uncomfortable. Based on the result of the survey, we can conclude that our robot behavior is still comfortable enough because the robot to be more responsible as we expected.

6.6.9 Comments from Respondents

To improve our guide robot performances, we also attempt to obtain feedbacks from our respondents. We will try to pay more attention to these feedbacks for the further development of our guide robot in the next future. We summarize some comments from the respondents as shown in Table. 6.6.

6.6.10 Computation Time

Besides the performance of each module, we also record the computation time needed by each module. Our human partner identification module requires 125 - 160 msec (6 - 8 fps) to identify the partner. Our human partner behavior module requires only $\approx$1 msec to estimate the partner head orientation. Lastly, our motion planner and control module requires only 15 msec to analyze and produce a control action.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

We have successfully built a socially aware guide robot system that is composed of three main components: (1) person identification system, (2) partner's behavior recognition system, and (3) motion planning and control system. For every component, we have contributed our own algorithm to resolve the challenges that exist.

To deal with a challenge in the person identification system, such as the change in illumination when the guide robot is operated in real environment, we have developed a novel illumination invariant method based on an extended reflectance model combined with a Fuzzy Inference System (FIS) whose rules are optimized using a Genetic Algorithm (GA). The proposed method is very effective to produce invariant face appearances in varying illumination conditions. It only requires a very low computation time, but achieve a high performance. Another advantage of our illumination invariant method is that by placing our illumination invariant method before the face detector, it can help the detection results since our method guarantees that facial feature still can be seen clearly.

To deal with a challenge in the partner's behavior recognition system, such as estimating the human head orientation in a wide range of angle, we have developed a novel descriptor for estimating the human head orientation. We combined many features such as Weber, gradient, and intensity deviation collectively, and compacting those features using covariance technique that was proven to be effective to characterize the differences of each head orientation than just using a single feature. We have shown that it can work very fast while still maintaining a strong discrimination ability.

We have also dealt with a challenge in the motion planning and control system, such as the acceptability of our robot's behavior by the human. We have developed a novel motion planner and control for a mobile guide robot that considers social awareness. The guiding skill and behavior are generated independently using Q-Learning approach to find the optimal socio-psychological-based control action for each situation.

Combining all components into one guide robot system has been done. Experiments using a realistic 3D simulator and real robot implementation in real environments show that using the self-learning approach, the over-reactive behaviors can be minimized. Smoothness and safety can be obtained in a real environment, as well as in the simulations. In the results, we have also shown the importance of maintaining coordination with the human partner. This coordination is effective to increase robot's responsibility to the human partner.

7.2 Future Work

We have presented our motion planner and control, and its implementation for a real mobile robot. Up to now, based on the experimental results in a limited distance, the robot performance when planning motions and executing controls is meet with our expectation. The guiding behavior in real implementation demonstrated by the robot is also good enough. However, the possibility of further development is still open. We have a plan to implement our robot system to be able to work on a large scale of environments. This means that equipping our robot with a large-scale localization and mapping is important.

Another possible plan is improving our person identification module to be able to work with large variations in the environment such as indoor and outdoor environments and in crowded environments. Combining multiple features such as face, clothing color, human height, gait analysis will be our next focus in the near future.

Our next focus is improving our illumination invariant face recognition system performances to deal with a distant person. Working with a distant person certainly decreasing the quality of the person's appearance due to low pixel's numbers. This problem makes many face recognition systems are not reliable for recognizing faces because a lack of facial features or a blur face image. Considering super-pixels approaches will be more interesting.

In addition, we also want to improve our illumination invariant face recognition system by utilizing Deep Learning approach. Even though we have obtained a very good result using our optimized Fuzzy system, however, we still use features as a result of our own thinking. Due to sophisticated results obtained using Deep Learning in many applications and in line with the era of big data, we believe that involving the Deep Learning in our illumination invariant face recognition system will produce a better result than the current achievement.

Bibliography

[1] C. Zielinski, "Robotics - Quo vadis?", In Journal of Measurements and Control, vol. 5, pp. 9-19, 2010.

[2] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an Interactive Museum Tour-Guide Robot", Artificial Intelligence, vol. 114, no. 1–2, pp. 3–55, 1999.

[3] K.O. Arras, N. Tomatis, and R. Siegwart, "Robox, a Remarkable Mobile Robot for the Real World", in: ISER, pp. 178–187, 2002.

[4] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C.R. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva", International Journal of Robotics Research, vol. 19, no. 11, pp. 972–999, 2000.

[5] A. Clodic, S. Fleury, R. Alami, R. Chatila, G. Bailly, L. Brèthes, M. Cottret, P. Danès, X. Dollat, F. Eliseï, I. Ferrané, M. Herrb, G. Infantes, C. Lemaire, F. Lerasle, J. Manhes, P. Marcoul, P. Menezes, V. Montreuil, and U.P. Sabatier, "Rackham: an Interactive Robot-Guide", in: IEEE RO-MAN, 2006.

[6] I. Nourbakhsh, C. Kunz, and T. Willeke, "The Mobot Museum Robot Installations: a Five Year Experiment", 2003.

[7] I. Macaluso, E. Ardizzone, A. Chella, M. Cossentino, A. Gentile, R. Gradino, I. Infantino, M. Liotta, R. Rizzo, and G. Scardino, "Experiences with Cicerobot, a Museum Guide Cognitive Robot", in: S. Bandini, S. Manzoni (Eds.), AI*IA 2005, Springer-Verlag, pp. 474–482, 2005.

[8] B. Jensen, N. Tomatis, L. Mayor, A. Drygajlo, and R. Siegwart, "Robots Meet Humans Interaction in Public Spaces", IEEE Transactions on Industrial Electronics, vol. 52, no. 6, 2005.

[9] E. Pacchierotti, H.I. Christensen, and P. Jensfelt, "Design of an Office-Guide Robot for Social Interaction Studies", IEEE/RSJ International Conference on Intelligent Robotics Systems, pp. 4965-4970, 2006.

[10] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, "An Effective Guide Robot in a Shopping Mall", The 4th ACM/IEEE International Conference on Human Robot Interaction, pp. 173-180, 2009.

[11] J.J. Rainer and R. Galan, "URBANO: A Tour-Guide Robot Learning to Make Better Speeches", The Second International Conference on Advanced Cognitive Technologies and Applications, pp. 66-71, 2010.

[12] M. Ghosh and H. Kuzuoka, "An Ethnomethodological Study of a Museum Guide Robot's Attempt at Engagement and Disengagement", Journal of Robotics, 2014.

[13] V. Evers, N. Menezes, L. Merino, D. Gavrila, F. Nabais, M. Pantic, P. Alvito, and D. Karreman, "The Development and Real-World Deployment of FROG, the Fun Robotic Outdoor Guide", ACM/IEEE International Conference on Human-Robot Interaction, pp. 100, 2014.

[14] C. Feng, S. Azenkot, and M. Cakmak, "Designing a Robot Guide for Blind People in Indoor Environments", ACM/IEEE International Conference on Human-Robot Interaction, pp. 107-108, 2015.

[15] T. Kruse, A.K. Pandey, R. Alami, and A. Kirsch, "Human-Aware Robot Navigation: A Survey", Robotics and Autonomous Systems, vol. 61, pp. 1726-1743, 2013.

[16] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a Socially Acceptable Collision Avoidance for a Mobile Robot Navigation Among Pedestrians Using a Pedestrian Model", International Journal of Social Robotics, vol. 6, no. 3, pp. 443-455, 2014.

[17] A.K. Pandey, "Towards Socially Intelligent Robots in Human Centered Environment", Doctoral Thesis, Universite de Toulouse, 2012.

[18] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot Companion: A Social-Force Based Approach With Human Awareness-Navigation in Crowded Environments", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1688-1694, 2013.

[19] E. Hall, "The Hidden Dimension", Anchor Books, 1966.

[20] F. Zanlungo, T. Ikeda, and T. Kanda, "Social Force Model With Explicit Collision Prediction", A Letters Journals Exploring The Frontiers of Physics, vol. 93, 2011.

[21] P.E. Hart, N.J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", IEEE Transactions on Systems Science and Cybernetics SSC4, vol. 4, no. 2, pp. 100-107, 1968.

[22] Y. Koren, and J. Borenstein, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation", Proceeding of the IEEE Conference on Robotics and Automation, pp. 1398-1404, 1991.

[23] S.M. LaValle and J. Kuffner, "Rapidly-Exploring Random Trees: Progress and Prospects", in: Proc. of 4th Int. Workshop on Algorithmic Foundations on Robotics, WAFR'00, 2000.

[24] D. Helbing and P. Molnar, "Social Force Model for Pedestrian Dynamics", Physical Review E, vol. 51, no. 5, pp. 4282-4286, 1995.

[25] M. Luber, J.A. Stork, G.D. Tipaldi, and K.O. Arras, "People Tracking with Human Motion Predictions from Social Forces", IEEE International Conference on Robotics and Automation, pp. 464-469, 2010.

[26] X. Yang, H. Dong, Q. Wang, Y. Chen, and X. Hu, "Guided Crowd Dynamics Via Modified Social Force Model", Physica A, pp. 1-11, 2014.

[27] J.L. Drury, J. Scholtz, and H.A. Yanco, "Awareness in Human-Robot Interaction", IEEE Conference on Systems, Man and Cybernetics, 2003.

[28] K.T. Song and W.J. Chen, "Face Recognition and Tracking for Human-Robot Interaction", IEEE International Conference on Systems, Man and Cybernetics, pp. 2877-2882, 2004.

[29] Y. Zhang, K. Hornfeck, and K. Lee, "Adaptive Face Recognition for Low-Cost, Embedded Human-Robot Interaction", Intelligent Autonomous Systems, vol. 12, pp. 863-872, 2013.

[30] R. Gross and V. Brajovie, "An Image Pre-processing Algorithm for Illumination Invariant Face Recognition", 4th International Conference on Audio and Video Based Biometric Person Authentication, pp. 10-18, 2003.

[31] G.P. Nam and K.R. Park, "New Fuzzy-based Retinex Method for the Illumination Normalization of Face Recognition", Int. Journal of Advanced Robotic System, vol. 9, pp. 103, 2012.

[32] H. Wang, S.Z. Li, Y. Wang, and J. Zhang, "Self Quotient Image for Face Recognition", International Conference on Image Processing, pp. 1397-1400, 2004.

[33] M. Nishiyama, T. Kozakaya, and O. Yamaguchi, "Illumination Normalization using Quotient Image-based Techniques", Recent Advances in Face Recognition, pp. 97-108, 2008.

[34] Y.K. Park, S.L. Park, and J.K. Kim, "Retinex Method Based on Adaptive Smoothing for Illumination Invariant Face Recognition", Signal Processing, 88, pp. 1929-1945, 2008.

[35] C.A. Perez and L.E. Castillo, "Illumination Compensation for Face Recognition based on Genetic Optimization of the Self-Quotient Image Method", International Symposium on Optomechatronic Technologies, pp. 322-327, 2009.

[36] W. Chen, M.J. Er, and S. Wu, "Illumination Compensation and Normalization for Robust Face Recognition Using Discrete Cosine Transform in Logarithm Domain", IEEE Transaction on System, Man And Cybernetics, vol. 36, no. 2, pp. 458-466. 2006.

[37] W. Kim, S. Suh, W. Hwang, and J.J. Han, "SVD Face: Illumination-Invariant Face Representation", IEEE Signal Processing Letters, vol. 21, no. 11, pp. 1336-1341, 2014.

[38] Y. Luo, Y.P. Guan, and C.Q. Zhang, "A Robust Illumination Normalization Method Based on Mean Estimation for Face Recognition", ISRN Machine Vision, pp. 1-10, 2013.

[39] Y.A. Ju, H.J. Soo, N.C. Kim, and M.H. Kim, "Face Recognition Using Local Statistic of Gradients and Correlations", 18th EUSIPCO, pp. 1169-1173, 2010.

[40] B.S.B. Dewantara and J. Miura, "OptiFuzz: A robust illumination invariant face recognition system and its implementation", Machine Vision and Applications, vol. 27, no. 6, pp. 877-891, 2016.

[41] B. Han, S. Lee, and H. S. Yang, "Head Pose Estimation Using Image Abstraction and Local Directional Quarternary Patterns For Multiclass Classification", Pattern Recognition Letters, vol. 45, pp. 145-153, 2014.

[42] B. Ma, A. Li, X. Chai, and S. Shan, "CovGa: A Novel Descriptor Based On Symmetry of Regions For Head Pose Estimation", Neurocomputing, 2014.

[43] L. Dong, L. Tao, and G. Xu, "Head Pose Estimation Using Covariance of Oriented Gradients", IEEE ICASSP, pp. 1470-1473, 2010.

[44] B.S.B. Dewantara and J. Miura, "Estimating Head Orientation using a Combination of Multiple Cues", IEICE Trans. on Information and Systems, vol. E99-D, no. 6, pp. 1603-1614, 2016.

[45] J.H. Holland, "Genetic Algorithms, Scientific American", pp. 66-72, 1992.

[46] R.S. Sutton and A.G. Barto, "Reinforcement Learning: An Introduction", The MIT Press, 2012.

[47] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", Conference on Computer Vision and Pattern Recognition, 2001.

[48] N. Dalal and B. Triggs, "Histogram of Oriented Gradients for Human Detection", International Conference on Computer Vision & Pattern Recognition, pp. 886-893, 2005.

[49] K.O. Arras, O.M. Mozos, and W. Burgard, "Using Boosted Features for the Detection of People in 2D Range Data", Proceeding of IEEE International Conference on Robotics and Autonomous, pp. 3402-3407, 2007.

[50] Z. Zainuddin, S. Kodagoda, and G. Dissanayake, "Torso Detection and Tracking using a 2D Laser Range Finder", Proceeding of Australasian Conference on Robotics and Automation, 2010.

[51] D. Hoeimm, A.A. Efros, and M. Hebert, "Putting Objects in Perspective", IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2137-2144, 2006.

[52] W. Choi and S. Savarese, "Multiple Target Tracking in World Coordinate with Single, Minimally Calibrated Camera", European Conference on Computer Vision, pp. 553-567, 2010.

[53] E.H. Mamdani, "Application Of Fuzzy Logic To Approximate Reasoning Using Linguistic Synthesis", IEEE Transactions on Computer C, vol. 26, no. 12, pp. 1182-1191, 1977.

[54] K. Koide and J. Miura, "Identification of a Specific Person Using Color, Height, and Gait Features for a Person Following Robot", Submitted to Robotics and Autonomous Systems.

[55] C. C. Chang and C. J. Lin, "LIBSVM : a library for support vector machines", ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, pp. 27:1-27, 2011.

[56] R.E. Schapire and Y. Singer, "Improved Boosting Algorithm Using Confidence Rate Predictions", Proceeding of the Annual Conference on Computational Learning Theory - COLT'98, vol. 37, pp. 297-336, 1998.

[57] X. Xie and K.M. Lam, "Face Recognition Under Varying Illumination based on a 2-D Face Shape Model", Pattern Recognition, vol. 38, no. 2, pp. 221-230, 2005.

[58] T. Zhang, Y.Y. Tang, B. Fang, Z. Shang, and X. Liu, "Face Recognition Under Varying Illumination Using Gradientfaces", IEEE Transactions on Image Processing, vol. 18, no. 11, pp. 2599-2606, 2009.

[59] T. Rui, Z. Yang, F. Liu, S. Jiang, and H. Li, "Block-Based Face Recognition Using WLD", PCM 2013, pp. 811-822, 2013.

[60] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Recognition With Local Binary Patterns", ECCV 2004, pp. 469-481, 2004.

[61] J.Y. Zhu, W.S. Zheng, and J.H. Lai, "Logarithm Gradient Histogram: A General Illumination Invariant Descriptor for Face Recognition", The 10th IEEE International Conference and Workshop on Automatic Face and Gesture Recognition, pp. 1-8, 2013.

[62] R. Guo, Q. Dai, and D. Hoeim, "Single-Image Shadow Detection and Removal using Paired Regions", IEEE Conference on Computer Vision and Pattern Recognition, pp. 2033-2040, 2011.

[63] E. Land, "Recent Advances in Retinex Theory", Vision Research, vol. 26, no. 1, pp. 7-21, 1986.

[64] P. Peer and F. Solina, "An Automatic Human Face Detection Method", Proceedings of the 4th Computer Vision Winter Workshop, pp. 122-130, 1999.

[65] A. Elgammal, C. Muang, and D. Hu, "Skin Detection - A Short Tutorial", Encyclopedia of Biometrics by Springer-Verlag Berlin Heidelberg, 2009.

[66] G. Grove, C. Zerweck, and J. Damia, "Human Skin Coloration using the RGB Color Space Model", The 4th International Symposium of L'Oreal Institute for Ethnic Hair and Skin Research, 2007.

[67] B.S.B. Dewantara and J. Miura, "Fuzzy-based Illumination Normalization For Face Recognition", IEEE Workshop on Advanced Robotics and Its Social Impacts, pp. 131-136, 2013.

[68] K. Nanaa, M. Rizon, M.N.A. Rahman, A. Almejrad, A.Z.A. Aziz, and S.B. Mohamed, "Eye Detection Using Composite Cross-Correlation", American Journal of Applied Sciences, vol. 10, no. 11, pp. 1448-1456, 2013.

[69] K. Fukui and O. Yamaguchi, "Face Recognition Using Multi-viewpoint Patterns for Robot Vision", 11th International Symposium of Robotics Research, pp. 192-201, 2003.

[70] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 643-660, 2001.

[71] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao, "The CAS-PEAL Large-Scale Chinese Face Database and Baseline Evaluations", IEEE Transactions on System, Man and Cybernetics - Part A: Systems and Humans, vol. 38, no. 1, pp. 149-161, 2008.

[72] M.A. Turk and A.P. Pentland, "Eigenfaces for Recognition", Journal of Cognitive and Neuroscience, vol. 3, pp. 71-86, 1991.

[73] E.M. Chutorian and M.M. Trivedi, "Head Pose Estimation in Computer Vision: A Survey", IEEE Transaction on PAMI, vol. 31, no. 4, pp. 607-626, 2009.

[74] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao, "WLD: A Robust Local Image Descriptor", IEEE Transactions of Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1705-1720, 2009.

[75] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[76] T. Ojala, M. Pietikainen, and D. Harwood, "A Comparative Study of Texture Measures with Classification Based on Feature Distributions", Pattern Recognition, vol. 29, pp. 51-59, 1996.

[77] O. Tuzel, F. Porikli, and P. Meer, "Region Covariance: A Fast Descriptor for Detection and Classification", ECCV, 2006.

[78] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors", Magnetic Resonance in Medicine, Wiley-Liss Inc., 2006.

[79] N. Yusup, A. M. Zain, and S. Z. M. Hashim, "Evolutionary Techniques in optimizing machining parameters: Review and recent applications (2007– 2011)", Expert Systems with Applications, vol. 39, pp.909-927, 2012.

[80] G. George and K. Raimond, “A Survey on Optimization Algorithms for Optimizing the Numerical Functions”, International Journal of Computer Applications, vol. 61, no. 6, pp. 41-46, 2013.

[81] N. Gourier, D. Hall, and J. L. Crowley, "Estimating Face Orientation From Robust Detection of Salient Facial Features", Proceedings of Pointing 2004, ICPR, International Workshop on Visual Observation of Deictic Gestures, Cambridge, UK, 2004.

[82] C.E. Thomaz and G.A. Giraldi, "A New Rangking Method for Principal Components Analysis and its Application to Face Image Analysis", Image and Vision Computing, vol. 28, no. 6, pp. 902-913, 2010.

[83] B.S.B. Dewantara and J. Miura, "The AISL Head Orientation Database and Preliminary Evaluations", IEEE International Electronic Symposium, pp. 146-150, 2015.

[84] I. Ardiyanto and J. Miura, "Partial Least Squares-based Human Upper Body Orientation Estimation with Combined Detection and Tracking", Image and Vision Computing, Vol. 32, No. 11, pp. 904-915, 2014.

[85] J.C.J. Chen and J.J.J. Lien, "Multi-View Face Detection and Pose Estimation", 18th IPPR Conference on Computer Vision, Graphics and Image Processing, pp. 933-940, 2005.

[86] www.bavuwe-software.com, "Decision Forest Parallel Implementation", 2012.

[87] Y. Cai, M. Yang, and J. Li, "Multiclass classification based on a deep convolutional network for head pose estimation", Frontiers of Information Technology and Electronic Engineering, vol. 16, no. 11, pp. 930-939, 2015.

[88] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. N. Vapnik, "Support Vector Regression Machines", in Advances in Neural Information Processing Systems 9, NIPS 1996, pp. 155–161, MIT Press., 1997.

[89] htpp://www.openrtm.org/, "Open Source Robotic Technology Middleware".

[90] http://www.coppeliarobotics.com/, "Virtual Robot Experimentation Platform".

Acknowledgements

First of all, I would like to give invaluable thanks only to Allah SWT, my God who always bless me so that I can survive and finish my study. Secondly, I would like to express a sincere gratitude to my parent, my little family, and my brothers who always support me and pray for me so that I can survive and finish my study. Special thank goes to my family who always waiting for me patiently.

I would like to express a sincere gratitude to my supervisor Prof. Jun Miura for the useful comments, discussions, remarks, and engagement through the learning process of this doctoral dissertation. I appreciate his vast knowledge and skill in the computer vision and robotic areas, and his assistance in writting journals, papers, reports, and dissertation, also for giving me chances to attend conferences. I gain a lot of knowledge during my study here.

I also express my gratitude to Prof. Michio Okada and Prof. Shigeru Kuriyama as the examiner members who kindly read my dissertation and gave many suggestions.

I am deeply grateful to the former Assistant Professor Dr. Junji Satake and the current Assistant Professor Dr. Shuji Oishi for supporting my research. My sincere thank also goes to Ms. Mikiko Kobayashi, who has helped me in many ways.

I thank my lab-mates at Advance Intelligent System Lab: D2 Kenji Koide; M2 Taku Kudo, Takahiro Sakai, Keiichiro Nishi, Wataru Miyazaki, and Masaru Genma; M1 Yoshitaka Kohari, Seiichiro Une, Nagai Minamoto, Mitsuhiro Demura, Yutaro Chikada, and Diar Fachruddin Sasongko; B4 Chanvisouth Shirdxay, Tanaka Shodai, Tanaka Shuguang, Sho Terada, Tsubasa Kato, Miki Kojima, Minote Tomoki, and Kazuho Morohashi; Research Student Liliana Villamar Gomez, and Researcher Gonzales Escalona Jose Uriel, for all the fun we have had in the lab. Also I deeply thank to the former lab-mate, Dr. Igi Ardiyanto, who has become my discussion partner in the lab. Special thanks to Koide-kun who always help me in conducting experiments using real robot, and Nishi-kun for borrowing me his turn-table system to collect head orientation database.

Special thanks goes to the former batch-1 of TUT-EEPIS cooperation: Dr. Agus Indra Gunawan, Dr. Bambang Sumantri, and Mr. Hero Yudo Martono for all the fun we have had in Japan. I also thank to my Indonesian friends who always help me collecting photos for my database and participating in survey: my colleagues in EEPIS Mr. Hary Oktavianto, Mr. Reesa Akbar, Mr. Zen Samsono Hadi, and Mr. Idris Winarno; PPI members Teuku Mahlil, Iwan Sukarno, Agus Suharto, Abraham Bhaskara, Sholihatta Aziz, Yogi hendrawan, Jody Setiawan, Rasyif Uchiha, and Kelvin Ng Wijaya my neighbor in apato.

Finally, I have to express my gratitude to the Directorate General of Higher Education, the Ministry of Research, Technology and Higher Education of Indonesia which has supported my study by providing the scholarship.

*

List of Publications

Journals

[1] B.S.B. Dewantara and J. Miura. "Estimating Head Orientation using a Combination of Multiple Cues". IEICE Transactions on Information and Systems, vol. E99-D, no. 6, pp. 1603-1613, 2016.

[2] B.S.B. Dewantara and J. Miura. "OptiFuzz: A Robust Illumination Invariant Face Recognition System and Its Implementation". Machine Vision and Applications (MVA), vol. 27, no. 6, pp. 877-891, 2016.

Conferences

[1] B.S.B. Dewantara and J. Miura. "Fuzzy-based Illumination Normalization for Face Recognition". IEEE Advance Robotics and Its Social Impacts (ARSO), pp. 131-136, 2013.

[2] B.S.B. Dewantara and J. Miura. "The AISL Head Orientation Database and Preliminary Evaluations". IEEE International Electronics Symposium (IES), pp. 146-150, 2015.

[3] B.S.B. Dewantara and J. Miura. "Generation of a Socially Aware Behavior of a Guide Robot Using Reinforcement Learning". IEEE International Electronics Symposium (IES), 2016. (Accepted)