Fusion d'informations multi-capteurs pour la commande du robot humanoïde NAO

Thanh Long Nguyen

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES

Spécialité : **STIC – Traitement de l'Information**

Arrêté ministériel : 25 mai 2016

Présentée par

Thanh Long NGUYEN

Thèse dirigée par **Didier COQUIN**
et codirigée par **Reda BOUKEZZOULA**

préparée au sein du **Laboratoire LISTIC: Laboratoire d'Informatique, Systèmes, Traitements de l'information et de la Connaissance**
dans **l'École Doctorale SISEO – Sciences et Ingénierie des Systèmes de l'Environnement et des Organisations**

Fusion d'informations multi-capteurs pour la commande du robot humanoïde NAO

Thèse soutenue publiquement le **5 Avril 2017**,
devant le jury composé de :

Madame Véronique BERGE-CHERFAOUI
Maître de conférences HDR, Université de Technologie de Compiègne, Rapporteur
Monsieur Olivier COLOT
Professeur, Université de Lille 1, Rapporteur
Monsieur Kacem CHEHDI
Professeur, Université de RENNES 1, Président
Madame Michèle ROMBAUT
Professeur, Université de Grenoble Alpes, Examinateur
Monsieur Didier COQUIN
Professeur, Université de Savoie Mont Blanc, Directeur de thèse
Monsieur Reda BOUKEZZOULA
Maître de conférences HDR, Université de Savoie Mont Blanc, Co-Directeur de thèse

Declaration of Authorship

I, Thanh-Long NGUYEN, declare that this thesis titled, “Multi-sensor Information Fusion
- Application for NAO Robot” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed:

Date:

"*I'd like to send my gratefulness to my academic training: University of Grenoble Alpes and LISTIC laboratory, Polytech Annecy-Chambery. I also thank to my supervisors: Didier COQUIN, Reda BOUKEZZOULA, and Stéphane PERRIN for their helps during my thesis.*"

Thanh-Long NGUYEN

To my wife, who gave me a lot of encouragements to finish this thesis.

Résumé de la Thèse

Le Contexte

De nos jours, la robotique joue un rôle très important tant dans le monde industriel que dans notre vie quotidienne. La robotique apparaît dans de nombreux domaines, comme la santé, l'armée, l'environnement et les divertissements. C'est le résultat de nombreux travaux de recherche.

Un robot est généralement équipé de plusieurs capteurs permettant de recevoir des informations du monde extérieur, de les interpréter et d'exécuter des actions. La qualité des informations dépend non seulement de la qualité des capteurs mais aussi de l'interprétation de l'environnement dans lequel le robot évolue. Ces facteurs vont influer sur les performances de prise de décision du robot, car ils sont liés aux incertitudes et aux imprécisions des informations recueillies. Par exemple, un robot sous-marin peut ne pas détecter un obstacle immergé si le sonar n'est pas de bonne qualité ou si le rapport signal/bruit est faible. C'est pourquoi la décision ne peut être certaine et précise.

En fait, il existe de nombreuses solutions pour surmonter les problèmes d'incertitudes et d'imprécisions [40] [47] [63] [9] [55]. La solution la plus utilisée est d'intégrer plusieurs capteurs pour la même tâche. C'est ce qu'on trouve dans les systèmes de décision multi-capteurs, et cela est appliqué dans de nombreux robots modernes qui doivent faire des opérations critiques. Le but principal de l'ajout de capteurs supplémentaires est d'accroître la certitude et de réduire autant que possible les décisions imprécises. Théoriquement, avoir plus de capteurs signifie avoir des informations supplémentaires, d'où une amélioration de la prise de décision. Mais cela nécessite une méthode efficace pour combiner les informations de ces capteurs.

En fait, la fusion d'informations issues de multi-capteurs en robotique a été validée expérimentalement dans de nombreuses recherches liées à différents secteurs d'activité. Par exemple, [21] propose une nouvelle approche pour la localisation simultanée et la cartographie (SLAM) en considérant un système multi-capteur composé d'un sonar et d'une caméra CDD. La transformation de Hough est utilisée pour extraire des caractéristiques géométriques à partir des données issues du sonar et du système de vision, puis un filtre de Kalman étendu (EKF) est utilisé pour fusionner l'information au niveau des caractéristiques. D'autre part, [19] considère un scénario de navigation à l'intérieur d'un bâtiment dans lequel les robots mobiles perdent en fiabilité lorsqu'ils se déplacent à grande vitesse. Pour remédier à ce problème, ils combinent un réseau de capteurs sans fil avec un RFID passif, et la fusion permet aux robots d'effectuer une navigation plus précise et d'éviter des obstacles statiques. [11] traite de l'incertitude et du traitement de l'imprécision lors du processus de localisation d'un robot mobile équipé d'un capteur extéroceptif et d'odomètres. Les données imprécises fournies par les deux capteurs sont fusionnées par propagation de contraintes sur des intervalles, sur la base de l'utilisation du modèle des croyances transférables de Smets. Dans [10], un algorithme de fusion de données est utilisé pour un robot de marche à six jambes (DLR Crawler) pour estimer sa position actuelle par rapport à la position de départ. L'algorithme est basé sur un filtre d'information de retour indirect qui fusionne les mesures issues d'une unité de mesure inertielle (IMU) avec des mesures d'odométrie des jambes 3D et des mesures d'odométrie

3D issues d'une caméra stéréo. Dans [78], le schéma de fusion de données est aussi utilisé pour la reconnaissance de l'activité humaine. Les données provenant de deux capteurs inertiels portables fixés sur un pied sont fusionnées pour déterminer le type d'activité, à partir d'une classification grossière. Ensuite, un module de classification plus précis, basé sur la discrimination d'heuristiques ou des modèles de Markov cachés est appliqué pour distinguer plus précisément les activités. [8] décrit une interface multimodale flexible basée sur des modalités de parole et de gestes afin de commander un robot mobile. Un cadre d'interprétation probabiliste et multi-hypothèses est utilisé pour fusionner les résultats des caractéristiques de la parole et du geste. D'autres exemples sur des applications de fusion multi-capteurs liées à la robotique peuvent également être trouvés dans [41] [53] [32] [43].

Contribution de cette thèse

Influencé par de nombreuses recherches en robotique et en fusion, nous avons travaillé sur un projet dans ce domaine en prenant comme plate-forme de validation, le robot humanoïde NAO, développé par la société Aldebaran Robotics. Il est de petite taille (hauteur de 55 cm), possède 25 degrés de liberté, ce qui lui permet de faire beaucoup de tâches complexes et mimer les comportements des humains. Il est notamment équipé de plusieurs capteurs pour recevoir des informations du monde extérieur: deux caméras HD pour le traitement de la vision, quatre microphones pour la reconnaissance vocale, un émetteur et un récepteur à ultrasons pour détecter les obstacles, deux capteurs tactiles pour les mains et un pour la tête. Deux bumpers aux pieds lui permettent de détecter obstacles par contacts. Il possède également de nombreux capteurs aux articulations pour la détermination de la position spatiale de ses membres.

Dans cette thèse, nous allons considérer les cas où le robot NAO reconnaît les couleurs et les objets colorés à l'aide de ses capteurs. Les microphones sont utilisés pour reconnaître les commandes vocales de l'homme, les capteurs sonar sont utilisés pour éviter les obstacles pendant son déplacement, et surtout, une caméra sur sa tête est utilisée pour détecter et reconnaître les objets. En fait, il existe de nombreuses approches pour la reconnaissance de la couleur dans la littérature, cependant, les appliquer au robot NAO, n'est pas chose facile, à cause des incertitudes et des imprécisions qui sont liées à ses capteurs et à l'environnement dans lequel il se déplace (conditions d'éclairage, d'occultations, ou de la confusion parmi les choix possibles car certaines couleurs/objets sont très similaires). Nous allons étudier l'effet de ces incertitudes et de ces imprécisions sur la capacité de prise de décision du robot NAO et proposons un système multi-caméras pour améliorer la fiabilité de la décision. Nous avons exploré la fusion de données issues de capteurs homogènes et celles issues de capteurs hétérogènes.

Comme nous l'avons vu plus haut, lors de la reconnaissance d'objets par un robot, nous ne pouvons pas exiger des conditions de travail idéal car il y a toujours des incertitudes et des imprécisions. Pour des tâches critiques, la fusion de données multi-cateurs est la solution adoptée pour de nombreuses applications. Cette recherche apporte une vue intéressante sur la façon dont on peut améliorer les facultés de perception d'un robot humanoïde. De plus, selon notre étude bibliographique, il n'existe pas d'autres travaux dans la littérature qui envisagent l'utilisation de la fusion de données issues de plusieurs

caméras pour la reconnaissance couleur/objet par le robot NAO. Pour cette raison, nous nous attendons à ce que ce travail soit une bonne référence pour de futurs travaux.

Hypothèse de la recherche

Il est clair que l'objectif de cette thèse est d'étudier l'effet des incertitudes et des imprécisions ainsi que l'importance de la fusion d'informations dans la robotique. Par conséquent, il soulève deux questions: avec un seul capteur, un robot peut-il accomplir ces tâches sans se tromper ? Sinon, des capteurs supplémentaires peuvent-ils apporter plus de fiabilité au système de prise de décision? Dans cette thèse, nous répondons à ces deux questions dans des scénarii de reconnaissance de couleur et d'objets colorés, en prenant le robot NAO comme plateforme applicative.

Tout d'abord, nous allons montrer que les résultats de l'utilisation d'une seule caméra du robot ne sont pas suffisamment fiables dans un environnement non contrôlé. Parfois, le robot donne un résultat totalement incorrect, c'est-à-dire que le nom de la cible détectée est faux. Où peut-être, il hésite entre deux réponses, ce qui ne lui permet pas de prendre la bonne décision. Dans le cas de la détection de la couleur, nous avons introduit un seuil ϵ qui contrôle le compromis entre la certitude et la fiabilité du système de décision, ainsi que le choix de sa valeur. Si nous voulons que les décisions soient plus sûres, nous diminuons la fiabilité du système, et vice-versa. Deuxièmement, nous soulignons que l'ajout de capteurs (caméras) supplémentaires améliore la prise de décision du système, et donc sa fiabilité. La fusion de multi-caméra réduit le cas d'incertitude et d'imprécision, et les résultats donnés pour un système multi-caméras sont meilleurs que les résultats de chaque caméra prise individuellement. En effet, nous démontrons l'hypothèse ci-dessus par de nombreuses analyses qui seront détaillées dans cette thèse, ainsi que par des résultats expérimentaux qui ont été testés sur le robot NAO, en utilisant plusieurs caméras de même nature (capteurs homogènes) et de natures différentes (capteurs hétérogènes) dans les scénarios de reconnaissance proposés.

Aperçu de la méthodologie

Dans cette thèse, la théorie des fonctions de croyance est considérée comme le choix le plus approprié pour faire de la fusion de données issues de plusieurs capteurs. Cette théorie nous permet de combiner des informations multi-sources au niveau de la prise de décision. En effet, un des avantages important de la théorie des fonctions de croyance est qu'elle peut modéliser d'une bonne façon l'incertitude et l'imprécision basées sur un modèle analytique ou sur la perception humaine. Cependant, la partie la plus difficile de cette approche est la façon dont nous pouvons construire la fonction de masses qui représente le degré de croyance en chaque hypothèse. Selon le type de scénario, nous le faisons différemment.

Dans le cas de la détection de la couleur, le robot NAO est invité à trouver un objet dont la couleur est décrite par un terme linguistique, par ex. "rouge", "brun", "orange" ... et c'est une tâche difficile pour le robot car la définition de chaque couleur varie selon les personnes, selon leur conception des frontières entre couleurs. Lors de la première étape, le robot reconnaît la commande vocale en mettant en œuvre le module de reconnaissance de la parole. NAO se déplace ensuite pour trouver la cible en utilisant une de ses caméras.

Par souci de simplicité, les objets demandés sont des balles et chacune a une couleur bien précise. Afin de détecter la forme de la balle, nous appliquons la transformation de Hough sur les images acquises. La valeur moyenne des pixels de la balle détectée est utilisée comme entrée du système flou de Sugeno que nous avons proposé. La sortie du système flou est une valeur numérique qui indique le nom de la couleur détectée. En effet, chaque couleur est affectée d'une valeur numérique constante, par exemple 4 pour le rouge, 5 pour l'orange, ... Cependant, lorsque la sortie floue se situe entre deux valeurs constantes, par exemple 4.35, le robot NAO a du mal à prendre la bonne décision. Les imprécisions peuvent provenir de nombreux facteurs défavorables telles que la qualité des capteurs (caméra) et les conditions d'éclairage de la pièce dans laquelle le robot se déplace. Afin de faire face à ces difficultés, nous ajoutons une caméra 2D au système, afin d'améliorer la prise de décision. A partir de la sortie floue de chaque caméra, nous construisons les fonctions de masse sur la base d'un seuil ϵ prédéfini. Ensuite, nous appliquons la règle de combinaison de Dempster pour fusionner les informations des caméras. Enfin, la décision est prise en choisissant le maximum de la probabilité pignistique. Une fois que le robot a décidé le nom de la couleur, il s'avance vers la balle choisie et la touche avec sa main.

Il convient de noter que la méthode ci-dessus est appliquée à la fusion de données homogènes (caméras 2D). Nous allons étendre cette méthode à la fusion de données hétérogènes et l'appliquerons à la reconnaissance d'objets colorés. Dans ce scénario, le robot NAO est invité à reconnaître un objet coloré situé devant lui. Une caméra IP (2D) et une caméra Axus (3D) ont été ajoutées aux côtés de la caméra de NAO pour former un système de caméras hétérogènes. Afin d'extraire les points caractéristiques des objets dans les scènes, nous utilisons des caractéristiques issues de la méthode SURF (pour les données 2D) et issues de la méthode SHOT (pour les données 3D). Après la collecte des points caractéristiques, nous construisons une fonction de masses pour chaque caméra basée sur les deux meilleures correspondances entre les points testés et les points caractéristiques sauvegardés dans une base d'apprentissage. C'est-à-dire, que chaque point caractéristique de l'objet à détecter votera pour une hypothèse dans le jeu de l'espace de puissance de la théorie des fonctions de croyance. Après une étape de normalisation, l'opérateur de combinaison de Dempster est utilisé pour fusionner ces masses. L'objet reconnu sera l'objet qui correspond à la probabilité pignistique maximale. Pour les tests, afin de travailler dans un environnement incertain, nous avons sélectionné des objets qui ont de nombreuses similitudes. De plus, pour rendre la reconnaissance plus complexe, les objets sont orientés avec des angles différents de l'apprentissage, par rapport au robot NAO. Les expérimentations ont montré que la fusion de décisions issue de capteurs hétérogènes améliore le taux de reconnaissance.

Structure de la thèse

La thèse est organisée comme suit. Tout d'abord, nous commençons par le chapitre 2 qui présente le contexte de ce travail, lié à l'augmentation de la perception du robot NAO à reconnaître des objets colorés. Dans ce chapitre, nous expliquons le choix des espaces colorimétriques retenus ainsi que la sélection d'un système flou de Sugeno pour la reconnaissance de la couleur d'une balle, avec une seule caméra. Ce système flou est ensuite décrit en détail ; l'étape de fuzzification et les règles d'inférence. Pour faire nos expérimentations, nous avons associé à chaque couleur, une valeur numérique précise. En

effet, plus nous voulons être précis et plus le taux de reconnaissance est faible et vice-versa. Or, la mesure issue d'un capteur ne peut donner un résultat précis, à cause des conditions d'éclairage et de traitement. Nous discutons alors de la fiabilité du système de reconnaissance et nous expliquons les difficultés liées à l'utilisation d'une seule caméra, en tenant compte d'un seuil ϵ d'incertitude.

Le chapitre 3 commence par introduire un système de plusieurs caméras homogènes, qui sera utilisé pour la reconnaissance des couleurs décrit au chapitre 2. Il présente ensuite quelques informations générales sur la théorie des fonctions de croyance, et décrit certains opérateurs de combinaison et des critères de décision. Ensuite, la construction des fonctions de masse basée sur la valeur du seuil ϵ introduit au chapitre 2 est présentée. Pour illustrer l'idée proposée, nous fournissons quelques exemples avant d'exposer les résultats expérimentaux appliqués sur le robot NAO.

Dans le chapitre 4, nous introduisons le contexte de la reconnaissance d'objets colorés pour le robot NAO, et avons un regard sur quelques travaux existants dans la littérature. Nous expliquons également l'intérêt de la fusion de capteurs hétérogènes, notamment dans ce cas de la reconnaissance d'objets. Le système utilisant plusieurs caméras hétérogènes est décrit, puis la combinaison par l'opérateur de Dempster est détaillée. Enfin, nous traitons un exemple illustratif pour mieux comprendre l'idée proposée, et nous présentons les résultats expérimentaux réalisés avec le robot NAO.

Enfin, le chapitre 5 conclut la thèse et expose quelques perspectives. En outre, nous fournissons trois annexes à la fin de la thèse afin que le lecteur trouve facilement les informations techniques liées aux expérimentations pour la reproductibilité de ces tests. Tous les travaux référencés dans cette thèse sont répertoriés dans la section Bibliographie.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

Nowadays, robotics acts a very important role in our industrial life and it's likely that they are our future. Robotics appears in many domains from laboratories to real applications in healthcare, army, environment, and entertainment. For that reason, it receives a big attention from many researches, and our work is one of them.

As a matter of fact, a robot is normally equipped by several sensors allowing receiving information from the external world. The quality of information not only depends on the quality of sensors but also the exploited environment. Sometimes, these factors affect to the performance of the robot when causing uncertainties and imprecisions, and lead to severe consequences. For example, a sub-marine robot fails to detect an underwater obstacle (e.g. a big fish) due to a low quality of its sonar or external noises (e.g. from the enemy), a dangerous contact might be happened. That's why the decision should be certain and precise.

Actually, there are many solutions to overcome the problems of uncertainties and imprecisions, e.g. [40] [47] [63] [9] [55] . However, the most popular way is to integrate more than one sensor for the same task, it's so called multi-sensor decision system, and this is applied in many modern robots which operate critical operations. The main purpose of adding extra sensors is to increase certainties and reduce as much as possible imprecise decisions. Theoretically, having more sensors means having additional information, and the improvement should work, and we just need a good method to combine the information from these sensors.

In fact, data fusion in robotics has been experimentally validated in many researches with different applied domains. For example the work in [21] proposes a novel approach for the simultaneous localization and map building (SLAM) by considering a multi-sensor system composed of sonar and a CDD camera. The Hough transformation is used to extracts geometrical features from raw sonar data and vision image, then the Extended Kalman Filter (EKF) is employed to fuse the information at the level of features. On the other hand, [19] considers the indoor navigating scenario in which mobile robots loose reliability when moving at high speed. They combine a wireless sensor network with a passive RFID, and the fusion allows the robot to perform more precise navigation and avoid static obstacles. [11] deals with uncertainty and imprecision treatment during the localizing process of a mobile robot equipped with an exteroceptive sensor and odometers. The imprecise data given by the two sensors are fused by constraint propagation on intervals, based on the use of the Transferable Belief Model of Smets. In [10], a multi-sensor

data fusion algorithm is used for a six-legged walking robot DLR Crawler to estimate its current pose with respect to the starting position. The algorithm is based on an indirect feedback information filter that fuses measurements from an inertial measurement unit (IMU) with relative 3D leg odometry measurements and relative 3D visual odometry measurements from a stereo camera. In [78], multi-sensor fusion scheme is used for human activity recognition. Data from two wearable inertial sensors attached on one foot are fused for coarse-grained classification to determine the type of activity. Then, a fine-grained classification module based on heuristic discrimination or hidden Markov model is applied to further distinguish the activities. The work presented in [8] describes a flexible multi-modal interface based on speech and gestures modalities in order to control a mobile robot. A probabilistic and multi-hypothesis interpreter framework is used to fuse results from speech and gesture components. More examples about the application of sensor fusion in robotics can also be found in [41] [53] [32] [43].

1.2 Contribution of This Research

Influenced from many researches in robotics and fusion in literature, we opened a project in this domain taking a humanoid robot as the platform for the validation. The robot's name is NAO and it was developed by the Aldebarans company. It has a small size with a height of 55 cm, however, having 25 degrees of freedom allows it to do many complex tasks and mimic human behaviors. Notably, it is equipped with several sensors to receive information from external world: two HD cameras, four microphones, a sender and a receiver for ultrasonics, two tactile sensors for hands and one for the head, two bumpers at the feet, one inertial unit, as well as 24 joints sensors.

In this thesis, we consider the cases where the NAO robot recognizes colors and objects using its sensors. The microphones are used to recognize commands from human, the sonar sensors are employed to avoid obstacles during its displacement, and especially, a camera on its head is used to detect and recognize targets.

Actually, there exist many approaches for the color and object recognition in the literature, however, during the robot's operation, uncertainties and imprecisions are unavoidable. These may come from the quality of sensors, or from the exploited environment such as lighting conditions, occlusion, or from the confuse among possible choices e.g. some colors/objects are too similar. We study the effect of these uncertainties and imprecisions on the decision-making ability of the NAO robot, and propose a multi-camera system to improve the reliability of the robot's decision. We have explored the performance with both types of fusion: homogeneous and heterogeneous sensors (cameras).

As discussed above, during the operation of a robot, we cannot demand an ideal working condition because there are always uncertainties and imprecisions. For critical tasks, the fusion of multi-sensor becomes more and more important. This research brings an interesting view on how a humanoid robot finds difficult in its tasks of recognition and how we can improve the faculty of perception of a humanoid robot. Additionally, according to our bibliography, there is no other works, which consider using multi-camera data fusion for the color/object recognition of the NAO robot. For that reason, we expect that this work is going to be a good reference for other researches of the same domain.

1.3 Hypothesis of the Research

It is clear that the objective of this thesis is to study the effect of uncertainties and imprecisions as well as the importance of multi-sensor data fusion in robotics. Therefore, it raises two questions: with only a single sensor, can a robot accomplish his tasks with a high efficiency against uncertainties and imprecisions or not? And if not, can additional sensors bring more reliability to the decision system? In this thesis, we answer the above two questions in scenarios of color and object recognition for the NAO robot.

First, we will show that the results of using only a camera of the robot are not reliable enough under experimented environment. Sometimes, the robot gives a totally incorrect result, i.e. the name of the detected target is false. Or may be, it hesitates between two or more outputs, which does not allow it to make a certain decision. For the case of color detection, we also introduce a threshold ϵ, which controls the trade-off between the certainty and the reliability of the decision system. If we want that the decisions are more certain, we may decrease the reliability of the system, and vice versa.

Second, we emphasize that the presence of additional camera sensors helps the decision system improve the reliability. The fusion of data from multi-camera reduces the case of uncertainty and imprecision, and the results given by multi-camera are better than the results of each individual camera in average.

Indeed, we demonstrate the above hypothesis by many analyses, which will be detailed in this thesis, and also by experimental results which were tested on the NAO robot with multi-camera of both homogeneous and heterogeneous types in the proposed recognition scenarios.

1.4 Outline of the Methodology

In this thesis, the Belief function theory is considered as the most appropriate choice for doing the fusion of data from multi-sensor. This theory allows us to combine information from multi-source at the decision level. Indeed, one of the advantages of the Belief function theory is that it can well model uncertainty and imprecision based on an analytical model or human perception. However, the most difficult part of this approach is how we can construct mass values which represent the degree of belief each hypothesis. Depending on the context of scenario, we do it differently.

In the case of color detection, the NAO robot is requested to find an object whose color is described by a linguistic term e.g. "red", "brown", "orange"... and it is a difficult task for the robot because the definition of each color varies for different people, depending on their conception. At the first step, the robot recognizes human command by using an implemented speech recognition module. It then walks around to find the target by using one of its camera on head. For the sake of simplicity, the requested objects are balls and each one has a color. In order to detect the ball's shape, we employ the Hough transformation in acquired images. The average pixel values of the detected ball are used as the inputs for a Sugeno Fuzzy system that is proposed by us. The output of the Fuzzy system will be a numerical value indicating the name of the detected color. Indeed, each

color is assigned a constant number and these numbers are arranged in an intuitive order to human eyes. However, there must be uncertainty when the Fuzzy output lies between two constant numbers so that we cannot make a decision, or imprecisions may come from the many unfavorable factors such as the quality of sensors and the lighting condition. In order to deal with these difficulties, we add more 2D cameras to the detection system to look for the same target. From the Fuzzy output of each camera, we construct the mass values based on a predefined threshold. After that, the Dempster-Shafer combination is applied to fuse the information from the cameras. At the final step, the decision is made by choosing the singleton hypothesis in the combined mass that has the maximum of pignistic probability. Once the robot has decided the name of the color, it moves and touches the ball with its hand.

It is worth noting that the above method applies the fusion of information provided by homogeneous sensors (2D cameras). On the other hand, we apply the fusion of heterogeneous sensor data for the case of object recognition. In this scenario, the NAO robot is requested to recognize an object frontwards, and these objects are trained in the preprocessing step. An IP camera (2D) and an Axus camera (3D) are added to the two sides of the NAO camera to form a multiple heterogeneous cameras system for the object recognition. In order to extract the feature points of objects in scenes, we employ the SURF (for 2D data) and the SHOT (for 3D data). After the feature points are collected, we construct a mass function for each camera based on the correspondences of tested feature points and trained feature points in the learning base. That is, for each feature point of the detected object, it will vote for one hypothesis in the power set of the Belief function theory, and doing the same thing for all the feature points constructs a mass function after a normalization step. Then, the Dempster-Shafer combination is used to fuse these masses and derive the final decision. In this work, we challenge the uncertainties and imprecisions by selecting the objects that have many similarities for the test. Moreover, the objects are turned around so that it makes difficult to be recognized.

For the above two scenarios, with homogeneous sensor data fusion and heterogeneous sensor data fusion, the results are very positive. Indeed, our experimental works show that using a single camera cannot guarantee the reliability sufficiently, but it is well done when having more sensors with the Dempster-Shafer theory.

1.5 Structure of the Thesis

To ensure a coherent structure for the thesis with regards to what have been presented in this introduction part, the next parts are organized as the following.

First, we begin with Chapter 2 introducing the scenario in which the NAO robot is requested to find a colored object. In this chapter, we explain the choice of color space as well as the selection of the Sugeno Fuzzy system for the detection with a single camera. This Fuzzy system is then described in detail from the fuzzification step and inference rules to some practical considerations. After that, we discuss about the reliability of the detection system and we demonstrate the difficulties of using only one camera, taking into account an uncertainty threshold and its promise.

Second, Chapter 3 begins by introducing a homogeneous multi-camera system for the color detection described in Chapter 2. It then presents some background information about the Dempster-Shafer theory and some of its combinations and decisions. After that, the method for constructing mass values based on the threshold value introduced in Chapter 2 is presented. To illustrate the proposed idea, we provide some examples before showing the experimental results.

In Chapter 4, we introduce the context of the object recognition for the NAO robot, and have a look at some existing works in literature. We also explain the advantage of heterogeneous sensor data fusion, notably in this case of object recognition. The methodology with the multi-camera system using the Dempster-Shafer theory is then described in detail. After that, we provide an illustrative example to easily draw the proposed idea. We present the experimental results and discussion at the end of the chapter.

Finally, Chapter 5 concludes the thesis with a review and some perspectives. In addition, we provide three appendices at the end of the thesis so that the reader can find technical information easily. All of the referenced works in this thesis are listed in the Bibliography section.

Chapter 2

Sugeno Fuzzy System for the Color Detection

It is clear that the human eye is able to distinguish and identify the color of any object. More than that, it can also detect the subtleties and details in spite of very small differences between two colors. Now we consider the same situation but for a mobile robot which moves in an indoor/outdoor environment. This robot is equipped with several sensors and it is provided functional and decision making capabilities to control the color detection process. From that two interesting questions come: can the robot give high reliability when detecting the color of an object in real-time? And is it capable of mimicking human perception to build the detection system?

In this chapter, we try to answer the above questions by considering a scenario in which a NAO robot detects the color of an object, using an intelligent decision system. This problem is not new but there has been no model to completely solve it due to the great variability of form and the colors of detected objects (orientation and size, change of lighting conditions, overlap among colors...).

The reader will notice that this chapter refers to many notions of analyses and representations (color spaces, Fuzzy system, image processing techniques...). We draw attention to the fact that our objective is not to detail these concepts since they have been already well developed in the literature, but we give a brief description of their operation and the bibliography associated with the techniques used in this thesis. For more details, the reader is invited to consult the cited publications.

The content of this chapter is organized as follow. First, Section 2.1 gives a brief description of the color detection in the context of a NAO robot. Section 2.2 shows some well-known available color spaces and analyses their strengths and weaknesses. After that, Section 2.3 and 2.4 discuss about some existing methods for solving the problem of color detection, and the use of the Sugeno Fuzzy system. Next, an interesting discussion about the reliability of the proposed method is put in Section 2.5, then an experimental study is shown in Section 2.6. Section 2.7 presents the improvement of the performance for the detection system. Finally Section 2.8 concludes the chapter.

FIGURE 2.1: The NAO robot.

2.1 The NAO Robot in the Context of Color Detection

2.1.1 The NAO Robot

In this work, we use a humanoid NAO robot as the platform of the development and validation (see Fig. 2.1). This robot was developed by the Aldebaran-robotics company and the one used for our work is of the fourth generation. Being in a humanoid form, it is equipped with several sensors:

- 2 cameras on its head.
- 4 microphones.
- Ultrasonic sensor (2 transmitters, 2 receivers).
- Tactile sensor: on top of the head and at the two hands.
- 2 bumpers at the feet to detect contacts.
- Inertial unit.
- Joint position sensors.

Additionally, the robot is designed with 25 degrees of freedom which allow it to facilitate the motions. The manufacturer also provides a Software Development Kit (SDK) which allows NAO users to develop their programs for the robot. More information about the NAO robot can be found in Appendix A.

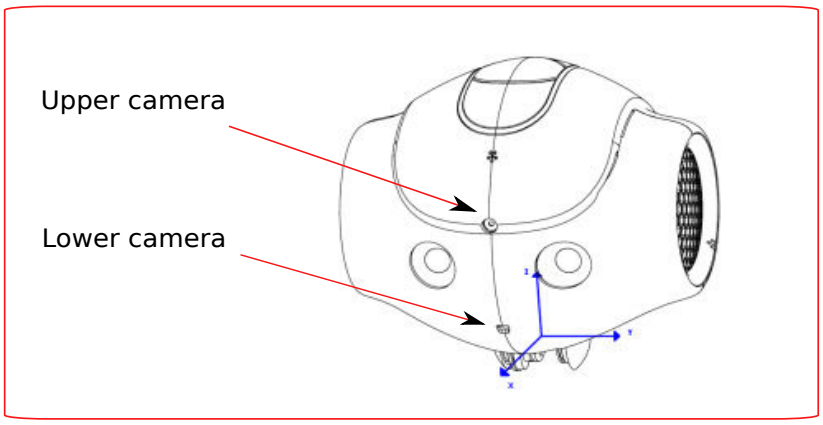

FIGURE 2.2: NAO cameras.

It is worth focusing on the cameras of the robot since they are used to capture images for the detection system. In fact, the robot has two cameras of the MT9M114 model with fixed focus (Fig. 2.2). The output of each camera is $960p@30fps$ and they give images of size 640×480. Despite that the cameras are not of a very good type, we still use them because we do not want to attach external cameras to the robot. Therefore, these cameras may not provide a good performance for the detection system and can cause uncertainties or imprecisions. However, solving this problem is also one of the main objectives that we consider in this work.

2.1.2 The Color Detection

This work is put in a global context of the recognition of 3D objects in an imprecise and uncertain environment where a humanoid robot is requested for the task. In this chapter, we consider only the side of the color detection since colors should be the first elements for a robot vision to perceive the world. Our objective is to provide a robot the capability of making decision and the proposed method is applied in a scenario of the color detection.

In the scenario, the NAO robot is put in a smart environment in which it can connect to other devices by a wireless network. A decision system is provided for the NAO robot so that it can determine the color of an detected object (e.g. a ball) observed by one of its cameras. The robot receives oral commands from human through its microphones to find an object, e.g in the form "NAO, please find the purple ball!". It then walks around to lookup the target and responds to human through its speaker if it finds the target (Fig. 2.3).

For the sake of simplicity to concentrate on the color detection, and without losing generality, the tested objects used in this work are balls with a monochrome color for each one. The colors are described in human terms and belong to the set of 9 names:
$Blue, Purple, Pink, Red, Brown, Orange, Yellow, Green, Cyan.$

As illustrated in Fig. 2.4, the process of color detection is composed of several intermediate steps as below. For more detail of the use of these steps, the reader is invited to Appendix B.

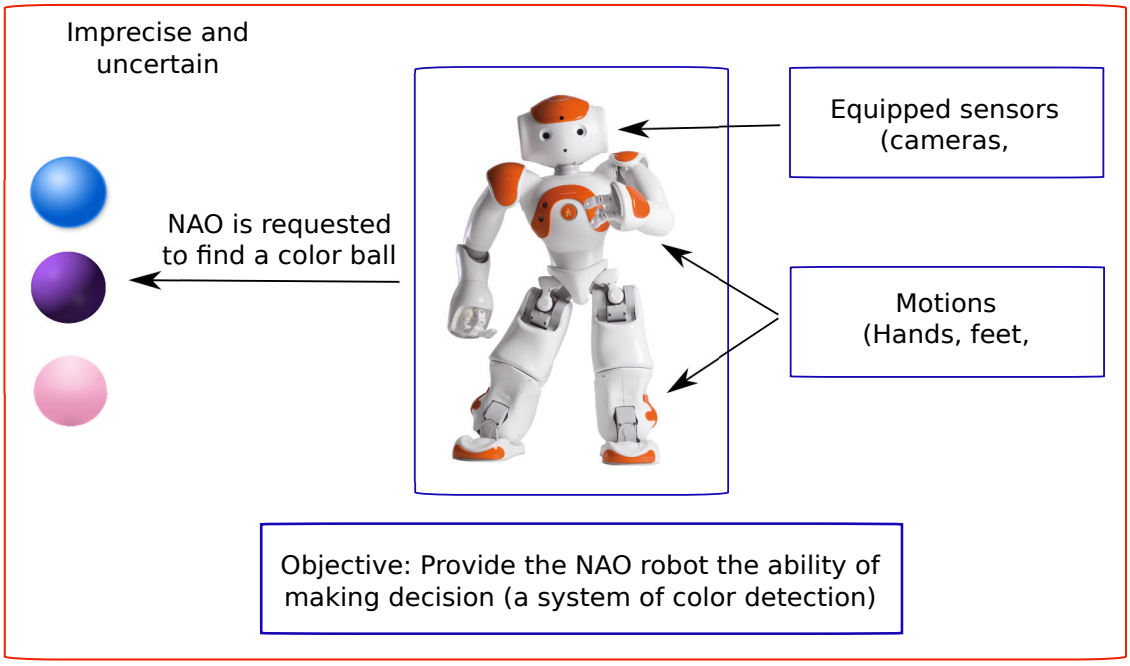

FIGURE 2.3: The NAO robot is requested to find a colored ball.

- Speech Recognition: The NAO robot needs to recognize the human command to know which colored object to be search. This step is done by using a module named ALSpeechRecognition in the SDK of the robot. We first register some commands inside the recognition engine, then during the execution, the robot recall the command that has the highest confidence.
- Move around: After receiving the human command, the robot will move around to find the objects. The ALMotion module is responsible for the movement. NAO tries to turn around to find the objects having the shape and the color specified.
- Capturing images: The NAO robot uses one of its camera for the capture during its movement. We subscribe to the ALVideoDevice module to get images. The SDK also provides a built-in OpenCV so that we can easily handle the image processing.
- Object Extraction: As mentioned above, we tested with balls in this work, and to detect the balls in images, the Hough transformation ([18]) is employed. This technique initially allows detecting analytic shapes such as lines, circles...

Actually, the color detection for robotics is not a new problem, however it has never been provided a really thorough method, since the problem differs in situations, exploited environments, executed platforms... For example in [76], a humanoid robot is requested to find and take a color ball. The object is previously learned by taking sample images then manually cropped around the region of interest, after that its histogram is projected onto the video to calculate probabilities. In [28], a similarity-based method is applied to recognize colors for a soccer robots system. The author define a coefficient to compare the similarity between colors, and estimate the uniform between two colors based on

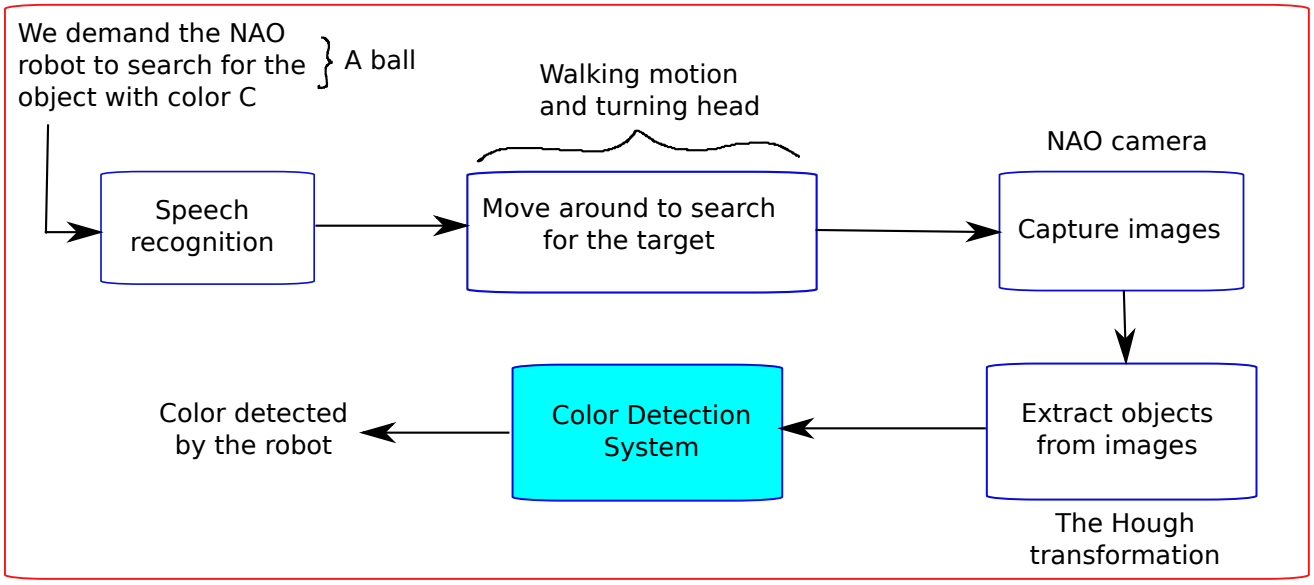

FIGURE 2.4: Steps to find the colored target.

predefined threshold values. In [37], the color adjustment, the thresholding algorithm, and the median filter are used to detect traffic lights, which supports for people who have color blindness. There is no doubt that these methods are simple to use and have presented good performances, however they consider so many fixed parameters obtained by a lot of experiments, besides that the intuitive perception is not taken into account carefully.

In the context of this work, no matter what kind of technique is implemented, the manipulated information still remains uncertainties and imprecisions. These problems may come from the overlap among close colors, the quality of sensors, the reliability of the methods for information extraction, and the presence of imperfection during the movement of the robot. Additionally, the change in the condition of observation and/or the operation can also affect the reliability of the detection system.

2.2 Consideration of Color Spaces

As the matter of fact, the choice of color space is very important for any visual processing task because it may strong affect the system performance (e.g. a study in [69]), and this work is not an exception. According to [34], a color space so defined is just the inverse of the RGB space, with white at the origin. Indeed, there are so many color spaces, each of them shows strengths and weaknesses, however the main objective is to try to maximize the color description as close as to the human eye perception.

This section introduces the three most well-known color spaces: RGB, HSV, and CIE-L*a*b. After that, we analyse their characteristics and explain the choice of color space for this work.

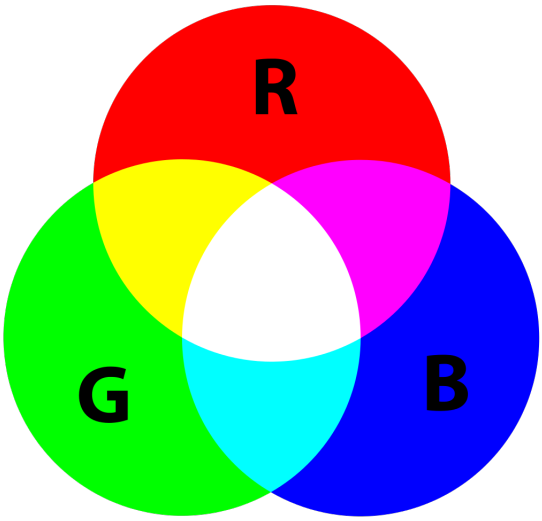

FIGURE 2.5: RGB color space.

2.2.1 RGB

The RGB color space is a very basic space based on the RGB color model. The three components R, G, B stand for Red, Green, and Blue, respectively, and this color space is constructed by all possible combinations from these three components. In fact, the idea leading to the specification of this color space was influenced by some works about human visual system in which it was stated that there are three types of photoreceptor which are approximately sensitive to the red, green, and blue region of the spectrum ([70]). From that, three types of cones are supposed and called L (Long), M (Middle) and S (Short) wavelength sensitivity, and most of images-capturing devices use an LMS-fashion light detector. The RGB color space is a device-dependant color space because the RGB values depend on the specific sensitivity of the device. Fig. 2.5 shows the illustration of the RGB space.

2.2.2 CIE-L*a*b

The CIE (Commission International de l'Eclairage , [33]) system describes colors as a luminance component Y, and two additional components X and Z. CIE-XYZ values are constructed from psychophysical experiments and correspond to the color matching characteristics of human visual system. CIE-L*a*b (or Lab from now for the sake of simplicity) represents a perceptually uniform color space obtained through a non-linear mapping from XYZ coordinates. In this space, L stands for the luminance, and ab represents the chroma. Interestingly, the colors in Lab are uniformly distributed in an ab plane, from green to red along the a axis and from blue to yellow along the b axis (Fig. 2.6). However, in Lab the pair (a, b) can be viewed as a pure color, and the L coordinate gives only the lightness of the color seen from human eyes ([20]). One color is defined by a point (a_1, b_1) in an ab plane of a given luminance L, and the colors are changed gradually and uniformly in the plane around this point.

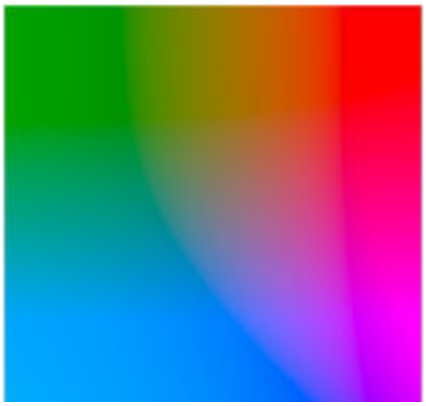

FIGURE 2.6: Color distribution in ab plane for L = 134.

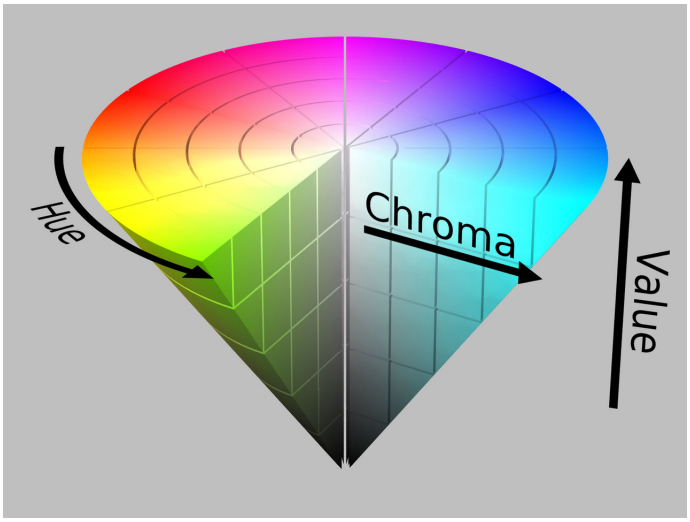

FIGURE 2.7: HSV color space.

2.2.3 HSV

HSV is one of the most common cylindrical-coordinate representations of points in an RGB color model and it is known to be intuitive to human eyes ([34]). HSV stands for Hue, Saturation and Value (brightness). In each cylinder, hue is represented by a circle around the vertical axis, the distance from the axis corresponds to saturation, and the distance along the axis determines value (Fig. 2.7).

The hue of a color refers to which pure color it resembles. For example all tints, tones and shades of red have the same hue. This component has a range from 0 to 360 (degree) corresponding to three primary colors: red, blue, green, and three secondary colors: yellow, cyan, and magenta. The saturation describes how white the color is and it ranges from 0 to 1. For example a pure green is fully saturated (saturation is 1), tints of green have saturations less than 1, and 0 means white. Finally, the value of a color describes the lightness of that color: how dark it is. A value of 0 means that it is totally black, and a value of 1 indicates a color with max light.

2.2.4 The Choice of Color Spaces

Actually, the RGB color space is the most popular for available image formats due to its simplicity and convenience. However, it is known that it cannot sufficiently of effectively distinguish the difference of the color separation degree. [70] indicates that for applications with natural images, the high correlation between its components become a difficulty. Moreover, its psychological non-intuitivity is another problem due to the difficulty of the visualization of a color defined with the R, G, B attributes for a human. Additionally, the low correlation between the perceived difference of two colors and the Euclidean distance in the RGB space are also challenges. When comparing to other advanced color spaces, the RGB color space normally shows its worse performance, for example in a study of comparison between HSV and RGB in a CBIR system in [35]. Under those circumstances, in this work we did not test with this color space.

The Lab color space is designed to approximate human vision, and the human eyes perceive gradual change of color as a uniform one, that's why this color space brings advantages. There are some works comparing the performance between Lab and HSV. In some cases, Lab gives better results, like [7] with a content-based image retrieval, but in some other cases like [6] with a test in color image segmentation, HSV performs better than Lab. From that, we cannot say which one is better in every case but it depends on the specificity of the work. In this scenario of color detection using the Sugeno Fuzzy system, we found that it is easier to construct the Fuzzy rule base with the HSV color space because the decompositions of each component into linguistic labels are easier due to the its nature to human perception. In this work we provide the experimental results with both these color spaces to see the comparison between them.

2.3 Methods for the Color Detection

This section focuses on how we derive a good approach for the color detection of the NAO robot. As discussed previously, the HSV and the Lab color spaces are used for the processing of images due to their suitability to the perception of human eyes. From an image captured by the NAO camera, the components H, S, V (or L, a, b) are determined, and they are considered as the inputs of the detection system (see Fig. 2.8).

Unfortunately, it is unlikely to be able to obtain an analytic model (in terms of mathematics) which is capable of giving a relation between the triplet (H, S, V) or (L, a, b) and the searched color. This is due to the complexity of the detection process and many other factors such as:

- The influence of the camera's quality and its setting.
- The presence of imperfection due to the change of operating conditions.
- The disturbance by the movement of the robot.
- The overlap among close colors.

Under those circumstances, there is no phase of analytical modelization for the input-output relation to be introduced. Instead, we consider the use of Artificial Intelligence

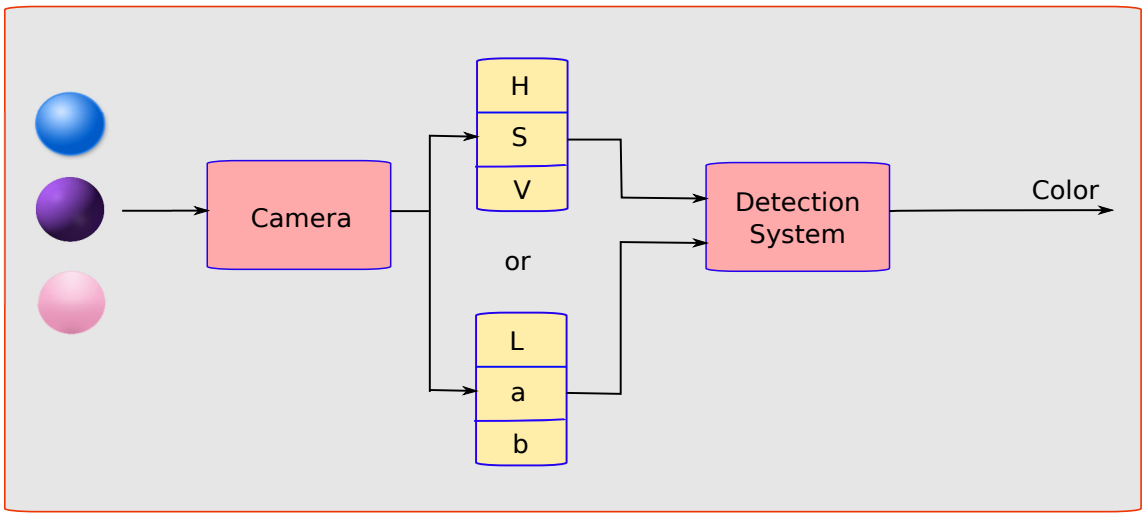

FIGURE 2.8: The system determines the color of object from an input (H, S, V) or (L, a, b).

(AI) techniques which allow representing that relation more easily. Indeed, AI approaches can bring formulations which can lend themselves more easily in cases where obtaining an analytical model is difficult, taking into account the human perception.

In the scope of AI, the Genetic Algorithm (GA), the Neural Network (NN), and the Fuzzy system (FS) are by far the most exploited in the literature. In this thesis, the Fuzzy system is given more attention due to the reasons that will be explained later.

2.3.1 Neural Network based Methods

The Neural Network is now used in many domains and important applications such that the recognition and identification, automation and robotics... We first talk about its basic element: an artificial neuron is defined as a calculation unit which takes into account the weighted sum of all of its input. The summation is then transformed by a transferring function (linear, threshold, sigmoid...) to produce the output of the neuron.

A Neural Network is just a set of neurons connected together, usually in organized layers. At each layer, every neuron has to connect to all the inputs of that layer. Interestingly, each connection is assigned a weight representing for the synaptic efficacy. A negative weight tries to inhibit the input while a positive one increases the input. In order to construct the network, we have to decide the number of layers, the number of neurons for each layer as well as the interconnection mechanism.

The main attractive property of using a Neural Network is that it can learn from the environment to improve the performance through a learning process. Generally, training a Neural Network means we have to find appropriate values for the weights of connections, and the methods for such training are so called network learning algorithm.

In the context of the color detection or even robotics, several works have taken the training advantages of the Neural Network. [15] is an example of using the Neural Network

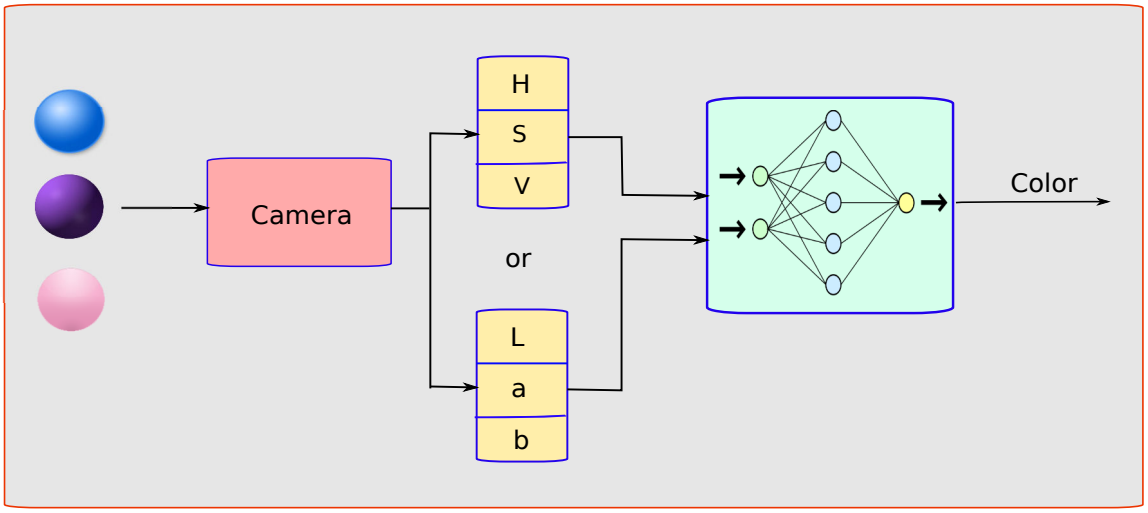

FIGURE 2.9: The color detection using the Neural Network.

in color image segmentation. The original colors of the detected objects are labelled by using a network with multi-sigmoid function. In [50], a color recognition system takes the HSV values as inputs for a three-layer Neural Network. The output of the network is then to determine the detected color belonging to a plant or the soil.

In the context of our work, a learning approach based on the Neural Network can be employed to design the detection system, as illustrated in Fig. 2.9.

In general, the Neural Network can provide a very good performance. However, while its main asset is the ability to learn, its structure and parameters never have a physical justification which is easy to interpret. Furthermore, the knowledge and human perception cannot be used to build them. In the viewpoint of practical implementation, this method employs digital techniques which have difficulties in iterative algorithms of adaptation, and more particularly in the context of mobile robots (real-time) where different conditions of operation and exploitation are taken into account. For those reasons, the use of the Neural Network in our case is not selected.

2.3.2 Genetic Algorithm based Methods

In general, the Genetic algorithm is an optimization approach which is influenced by the genetic evolution in nature: selection, crossing, and mutation. The goal of the Genetic algorithm is not to find an exact analytical solution, but to find satisfactory solutions in different criteria.

Normally, the implementation of the Genetic algorithm follows five considerations:

- sizing parameters: population size, number of generations, stop conditions...
- An encoding principle for each element of populations.
- A mechanism for generating the initial population.

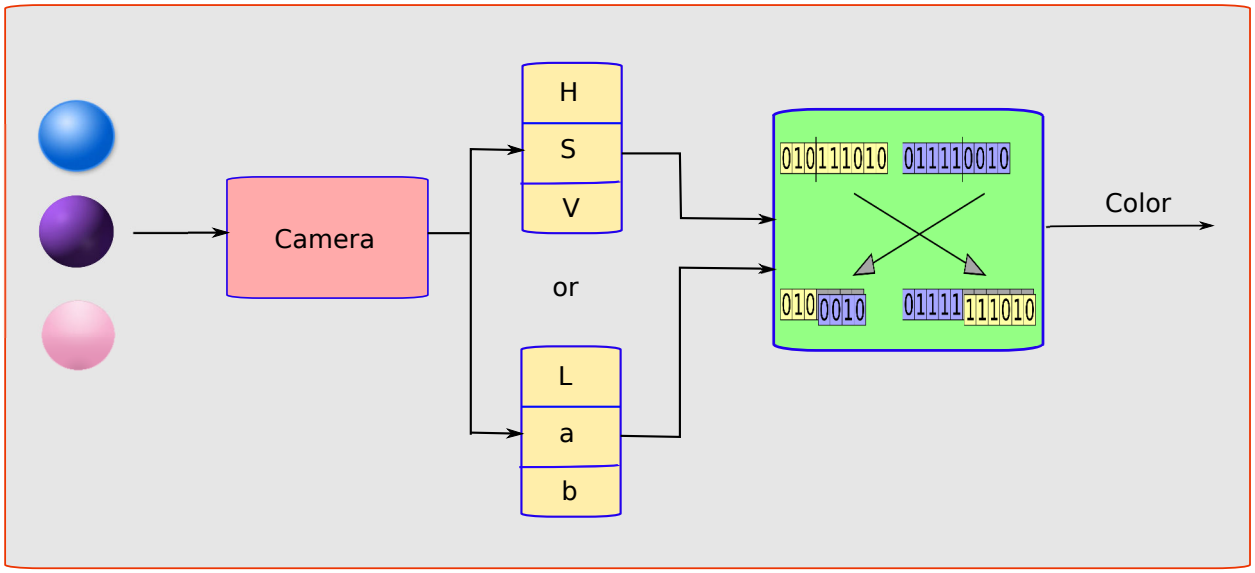

FIGURE 2.10: The color detection using the Genetic algorithm.

- An objective function for optimizing (fitness function).
- Operators to diversify the population over generations.

The performance of the Genetic algorithm in the problems of learning is still indisputable and it can be exploited in our problem of the color detection (see Fig. 2.10).

In fact, the Genetic algorithm has been widely applied in literature. For example in [66], an approach for the color recognition of a semi-autonomous soccer robot is proposed. They first define a new illumination invariant color space based on the dichromatic reflection model. After that, the Genetic algorithm is employed to determine the most discriminating color model among a multidimensional set of color models. On the other hand, [13] uses the Genetic algorithm to evolve an optimal chrominance space transformation instead of adapting the thresholds that define a specific color region, which improves the reliability and performance of color classification.

In spite of the success of the Genetic algorithm, its implementation still brings some disadvantages:

- It takes too much time of calculation, so it should not be adapted for real-time systems.
- The adjustment of a Genetic algorithm remains a delicate issue (including the problem of genetic drift).
- The generational replacement and the choice of representation remain difficult problems, particularly in an imprecise and uncertain environment.

Due to the above the drawbacks we cannot find a good reason to apply the Genetic algorithm in our context.

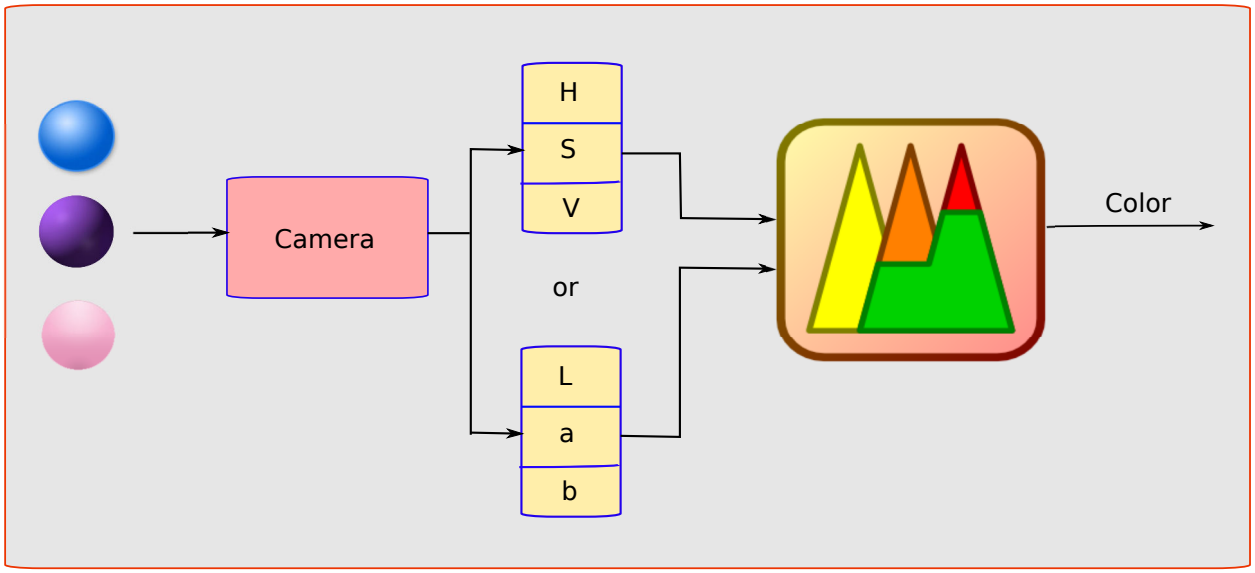

FIGURE 2.11: The color detection using the Fuzzy system.

2.3.3 Fuzzy System based Methods

As an alternative to the Neural Network and the Genetic algorithm, the Fuzzy system can also be exploited (Fig. 2.11). In fact, Fuzzy inference system is a universal approximator too but it has two strong points. First, its construction is done based on the human perception, and second, it has an excellently descriptive capability due to the use of linguistic variables. Therefore it appears natural to move towards a fuzzy approach where the detection system is constructed based on a fuzzy model.

If in the mathematical point of view, systems are classified according to the nature of the equations that characterize them (linear, non-linear...), the Fuzzy system is in turn listed based on its natural structure. Normally there are two main families of Fuzzy systems: Fuzzy systems with symbolic conclusions (Mamdani systems) and Fuzzy systems with functional conclusions (Takagi-Sugeno-Kang - TSK systems) which can take different forms: linear, polynomial, statistical, or dynamic equations. These two types of systems follow a collection of rules: "if...then". In the two cases, the premisses of rules are expressed symbolically. Only the expressions of the conclusions of rules allow distinguishing the two families of systems. The Mamdani system uses symbolic conclusions and the Sugeno system uses the numerical conclusions. In the internal point of view, a calculating mechanism is associated with each type of system. For the Sugeno system, it is purely numerical and expressed easily the analytical way according to a unique, common approach to all the systems of the family. On the other hand, the implementation of the Mamdani system may be considered in different ways (the choice of inference operators and the method of defuzzification).

In the literature, the idea of using the Fuzzy system in the color detection, recognition has been reported in many works. For instance in [39], [42], and [48], the authors propose using the Mamdani typed fuzzy systems to solve the problem of color detection. However, it is noted that many fuzzy-based color detection systems use Mamdani inference where

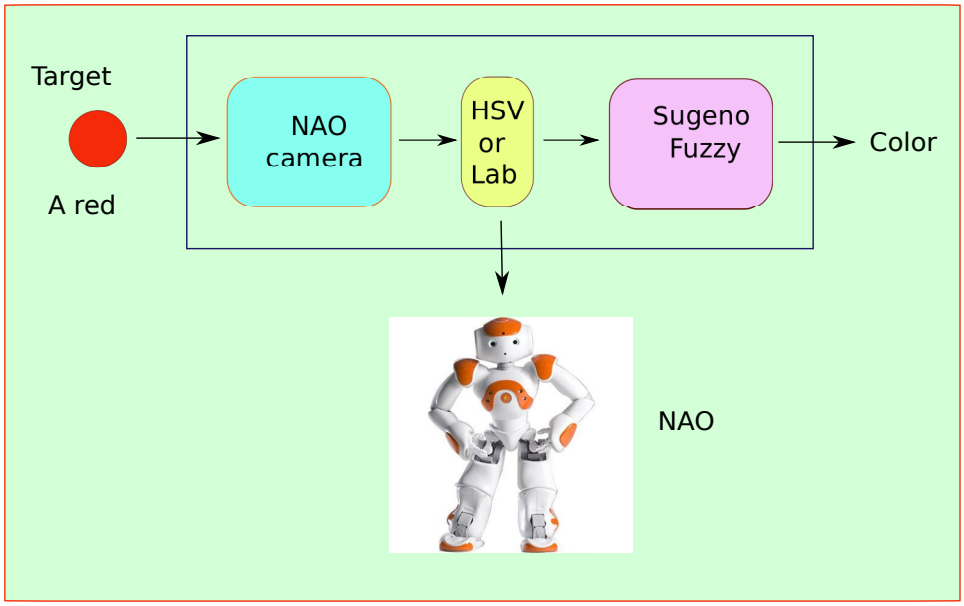

FIGURE 2.12: The Sugeno Fuzzy system for color detection with NAO robot.

nonlinear MIN/MAX operators are employed in the inference mechanism, and we do not find any advantage of using it in our research context. Moreover, the use of these operators introduces the nonlinearity that may affect to the performance of detection system in real time. On another hand, the Sugeno type presents structural properties ([25]) that allow exact piecewise multi-linear representation which permits us to integrate this fuzzy system design in some adaptive or learning strategies to specify the rule base parameters. For instance, adaptive and learning fuzzy strategy can be used in the future to adjust some parameters so that the detection performance can be improved. Additionally, the Sugeno type can bring simplicity in the calculation. Under those circumstances, in this work we employ the Sugeno Fuzzy system for the color detection.

2.4 The Sugeno Fuzzy System in the Color Detection

2.4.1 Overall Process

Fig. 2.12 depicts the overview of the Sugeno Fuzzy system applied in a NAO robot for the color detection. First, the NAO robot is requested to find an object with a specific color (e.g red), it then walks around to look for the target. For the sake of simplicity, in our work, the demanded objects will be colored balls, and Hough transformation ([18]) is applied to detect balls' shape in images. Thereafter, the pixels of the detected ball are extracted and their average values in HSV (or Lab) are calculated to use as inputs of the Fuzzy system.

In the Sugeno Fuzzy system, the inputs HSV (or Lab) are fuzzified by associated membership functions. Then a set of inference rules are applied to determine the output

color by applying the Sugeno zero-order formula. After that, the NAO robot responds the output color to its demanding person.

2.4.2 Membership Functions

For the purpose of taking into account human perception in the detection, and due to the difficulty of obtaining an exact mathematical model for the relation between the triplet of $\{H, S, V\}$ or $\{L, a, b\}$ and the output colors, the construction of the Fuzzy system is based on human observation. Indeed, an expert distinguishes several zones by observing the input-output behaviours of color detection, from that a partitioning of the universes of discourse in Fuzzy subsets are provided. This partitioning allows describing the crisp input values by linguistic labels.

Due to the nature of the Lab color space as explained in Section 2.2, the component a represents colors' position between red/magenta and green, so we divide its range into 5 linguistic labels: Green, Greenish, Middle of a, Reddish, and Red, as illustrated by the membership functions in Fig. 2.13. Similarly, the component b composes of 5 labels: Blue, Bluish, Middle of b, Yellowish, and Yellow. Because the component L represents the lightness of color, so we simply call its five labels as: Very Low, Low, Normal, High, and Very High, to describe the degree of lightness.

On the other hand, the component H of HSV color space is described by seven linguistic labels based on human perception of colors' tones: Red, Orange, Yellow, Green, Cyan, Blue, and Purple. Interestingly, a special point of this component is that it can be translated as a colors circle in which the value of 0 and 360 both specify a red color. Therefore, the membership functions of the component H can be represented as in Fig. 2.14. For the component S, because it describes the saturation of color, so we simply use three labels: Pale, Normal, and Clear. For the component V, Dark, Normal, and Light are used to describe the degree of lightness (value) of color.

2.4.3 Sugeno Inference for Ouput Colors

After constructing the membership functions for each color component, we have to determine the output color for each combination of $\{H, S, V\}$ (or $\{L, a, b\}$). As already mentioned, the Sugeno zero-order inference is applied for the reasoning process because it is difficult to define a mathematical relationship between each combination and the output color. With this in mind, we assign a numeric constant for each output color as shown in Table 2.1. The assignment is done in an appropriate way: the output colors are arranged in an order that is intuitive to human perception.

Each inference rule is then in the following form:

$$R^{(i_1,i_2,i_3)}: \text{ if } c_1 \text{ is } c_1^{i_1} \text{ and } c_2 \text{ is } c_2^{i_2} \text{ and } c_3 \text{ is } c_3^{i_3} \text{ then } C = C_i \tag{2.1}$$

where c_1, c_2, c_3 respectively represent for the three components H,S,V (with HSV space) or L,a,b (with Lab space), and $c_1^{i_1}, c_2^{i_2}, c_3^{i_3}$ are the linguistic values in the set of predefined labels for the input c_1, c_2, c_3, respectively. C_i is a numeric constant assigned to the output C of the rule indexed by (i_1, i_2, i_3), each of these numbers represents a color name in Table

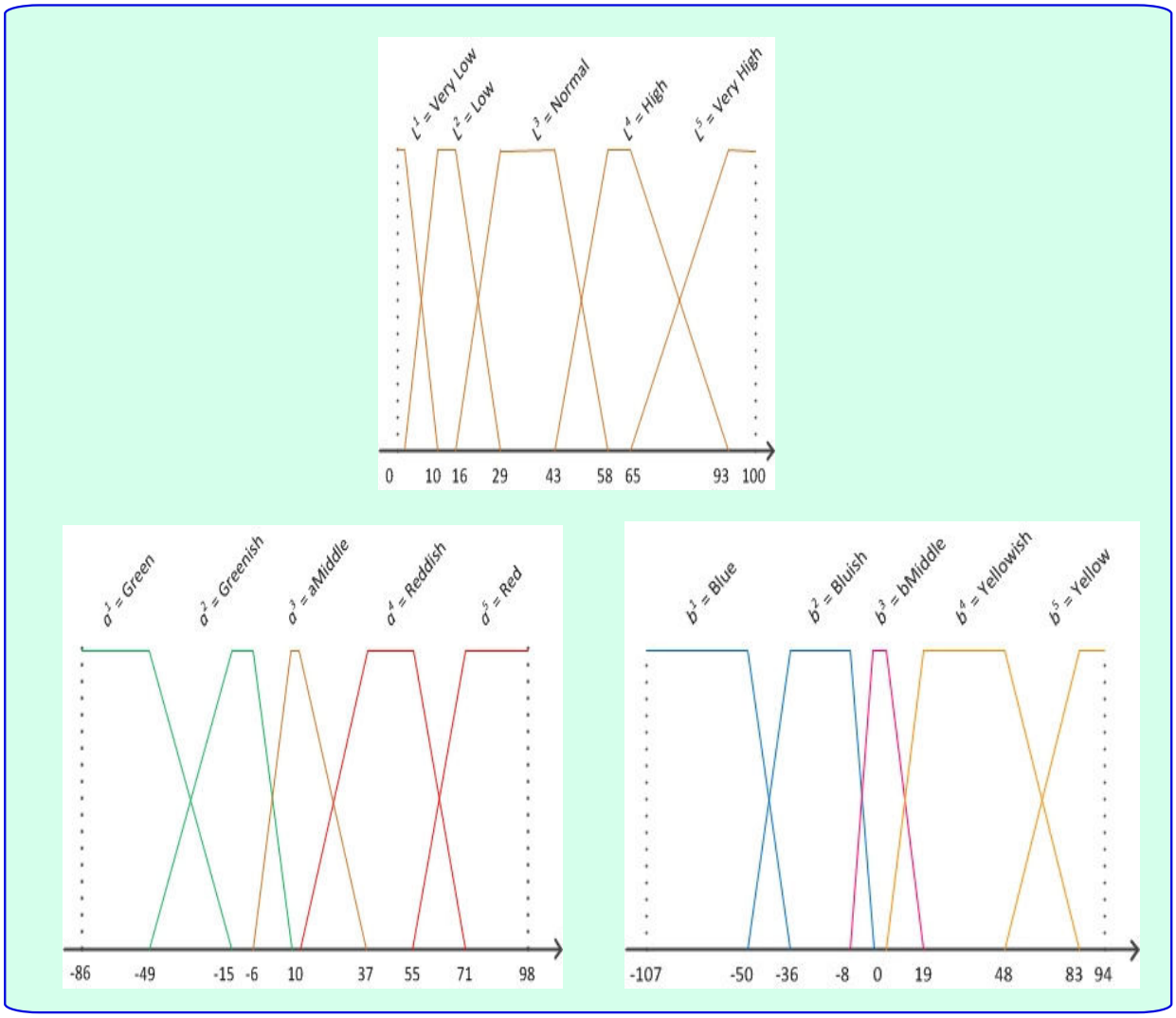

FIGURE 2.13: The membership functions for L,a,b components in the Lab color space. They are chosen after being experimented with the colored balls.

Color C_i	**Color Number**
Blue	1
Purple	2
Pink	3
Red	4
Brown	5
Orange	6
Yellow	7
Green	8
Cyan	9

TABLE 2.1: The numbers assigned for colors.

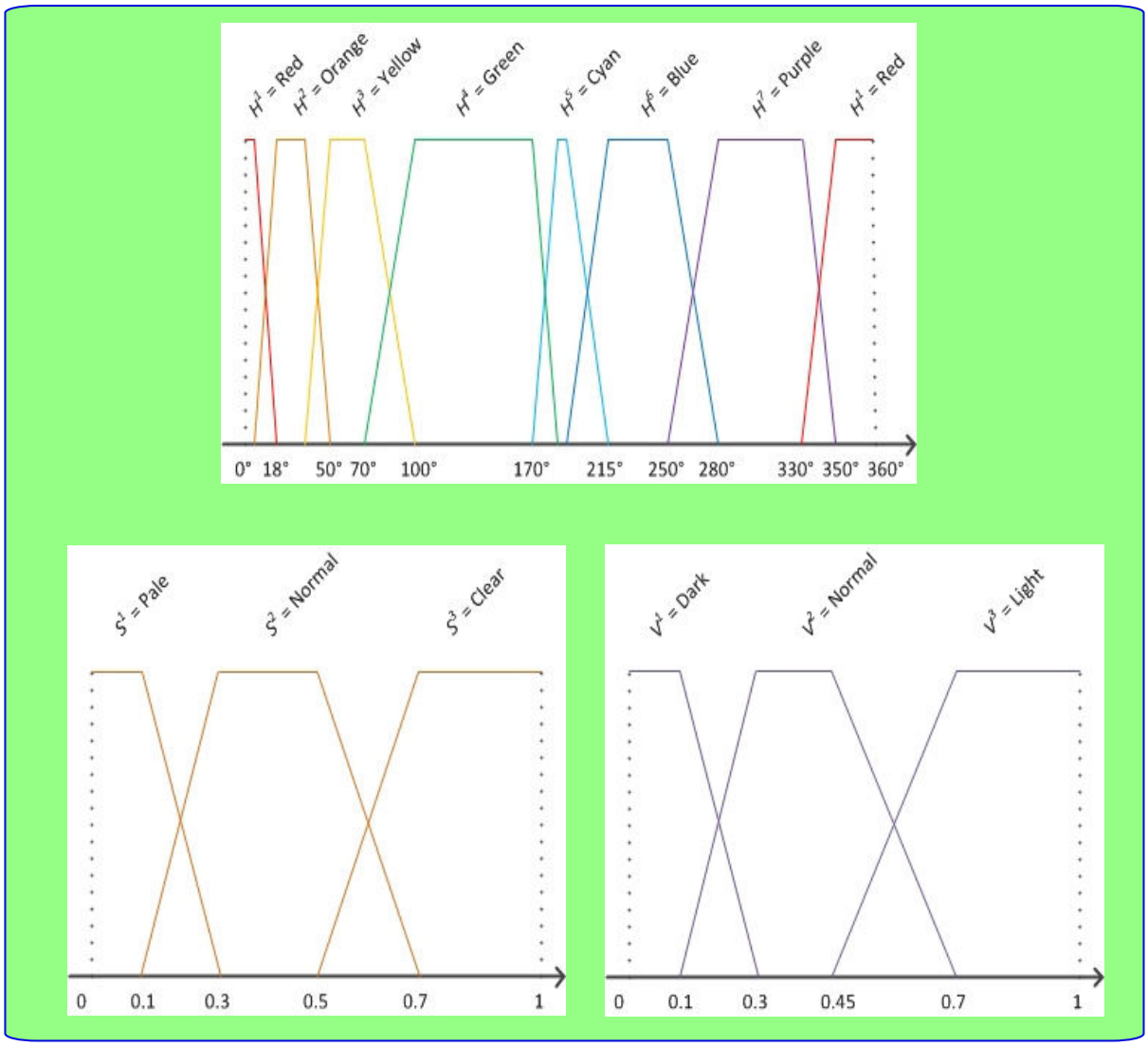

FIGURE 2.14: The embership functions for H,S,V components in the HSV color space. They are chosen after being experimented with the colored balls.

2.1, i.e. $\{C_i\}_{i=1,2,...,9} = \{C_1, C_2, ..., C_9\}$
= {Blue, Purple, Pink, Red, Brown, Orange, Yellow, Green, Cyan}.

Finally, the output which decides the detected color is generated by the following equation:

$$C = \frac{\sum_{(i_1,i_2,i_3)\in I} \zeta^{(i_1,i_2,i_3)}(c_1, c_2, c_3).C_i}{\sum_{(i_1,i_2,i_3)\in I} \zeta^{(i_1,i_2,i_3)}(c_1, c_2, c_3)} \tag{2.2}$$

where $\zeta^{(i_1,i_2,i_3)}$ represents the truth value of the premises of the rule and $I = I_1 \times I_2 \times I_3$ indicates the set of labels representing the rule base, with:

- $i_1 \in I_1 = \{1, 2, ..., |I_1|\}$
- $i_2 \in I_2 = \{1, 2, ..., |I_2|\}$
- $i_3 \in I_3 = \{1, 2, ..., |I_3|\}$

Therefore, with the HSV color space, we have $7 \times 3 \times 3 = 63$ rules, and with the Lab color space, we have $5 \times 5 \times 5 = 125$ rules for the complete Fuzzy system. Table 2.2 shows the inference rules for the Sugeno Fuzzy system in the HSV color space. The term ANY means that the corresponding input variable can be substituted by any linguistic value of its set of terms, while $NOT\ X$ means all the terms except X. For the Lab color space, because it has too many rules (125), so we just put some example rules that generate the blue color in Table 2.3, with the term OR specifying the possibility of using any term in the operands.

2.4.4 A Practical Consideration

It is clear that when applying a Fuzzy system to make decisions and using some defuzzification methods like weighted average, we should consider the order of the output decisions, e.g. the output speed control of a vehicle should be $Low, Average, Fast$ but cannot be $Low, Fast, Average$ or any other different order, if not so we will obtain unexpected results. This is also the reason why in this work, we consider the seven output colors: Blue, Purple, Red, Orange, Yellow, Green, Cyan as the colors for the balls, because from the human perception, these colors are arranged in an intuitive order (like in the Hue circle). Furthermore, we add two other colored balls: Pink and Brown such that the tested pink color lies between Purple and Red, and the tested brown color lies between Red and Orange, as perceptionally tested. From that, we gain 9 colors having an intuitive order and assign them constant numbers as mentioned in Table 2.1.

By this arrangement and assignment, we guarantee the right behaviour of the Fuzzy system because it avoids unexpected colors as outputs after activating some inference rules that are not related. For example, the combination of only red and orange cannot be a green one, so the green color should not be assigned a number between the ones of red and orange.

In addition, the first and the last colors after the number assignment are blue and cyan, and as human perceives, they are the two colors that are next together (like in the

Hue	**Saturation**	**Value**	Color C_i
Blue	ANY	ANY	1
Green	ANY	ANY	8
Cyan	Normal	Normal	8
Cyan	Clear	ANY	9
Cyan	Pale	ANY	9
Cyan	Normal	NOT Normal	9
Orange	Clear	Normal	5
Orange	Pale	ANY	6
Orange	Normal	ANY	6
Orange	Clear	NOT Normal	6
Yellow	Pale	ANY	7
Yellow	Normal	Normal	8
Yellow	Normal	NOT Normal	7
Yellow	Clear	Normal	8
Yellow	Clear	NOT Normal	7
Red	Pale	Light	3
Red	Pale	NOT Light	4
Red	Normal	Dark	4
Red	Normal	Normal	5
Red	Normal	Light	3
Red	Clear	Normal	5
Red	Clear	NOT Normal	4
Purple	Pale	Light	3
Purple	Pale	NOT Light	2
Purple	Normal	Light	3
Purple	Normal	NOT Light	2
Purple	Clear	ANY	2

TABLE 2.2: Inference rules for the HSV color space.

L	**a**	**b**	Color C_i
Very Low	ANY	Blue	1
Low OR Normal	aMiddle	Blue OR Bluish	1
Very Low	Greenish	Bluish	1
Low OR Normal	Reddish	Blue	1
Low	Green OR Greenish	Blue	1
Very Low	aMiddle	bMiddle	1

TABLE 2.3: Inference rules for the Lab color space to generate the blue color (constant number = 1).

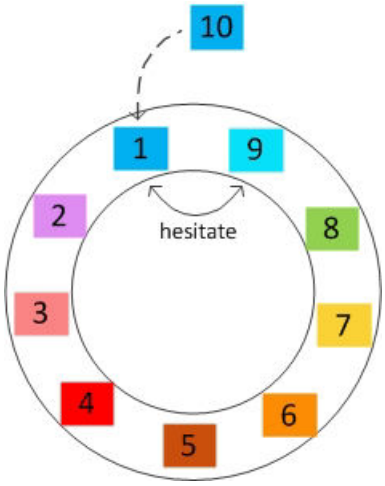

FIGURE 2.15: Illustration for the problem of the first and the last color.

Hue circle). In the normal process of the Sugeno Fuzzy system, if there are only conflicts between these two colors (with their number are 1 and 9, respectively), the system will result a number between 1 and 9, e.g. 4, 5... and it is absolutely not an expected result. In order to solve this problem, we consider the list of output colors as a circle manner in Fig. 2.15. We add a virtual number 10 representing for blue if there are conflicts between it and cyan, so the output of the Sugeno Fuzzy system will be between these two colors only. More virtual numbers can also be added if there are conflicts between the first two colors and the last two colors by using the same principle, however for the rule bases in our work, only one virtual number is enough. By this way, we still guarantee the right input-output behaviour of the system according to human evaluation, which was not considered in many other works applying Fuzzy system for color detection.

2.5 The Reliability of the Proposed Detection Method

2.5.1 The Influence of the Detection Threshold

The proposed method of color detection is clearly described in a visual processing context using the Sugeno Fuzzy system which gives constant conclusions. This allows avoiding the difficulties when using the symbolical systems of Mamdani (conjunctive or implicative form) and the defuzzification method that has to be used.

However, from the numerical nature of the exploited Fuzzy system, it is clear that a detection threshold ϵ should be introduced in the real-time implementation of the proposed detection strategy (see Fig. 2.16). When demanded to find a color ball having constant number C_i, the result is thought to be correct if the Sugeno Fuzzy system gives an output C in the interval $[C_i - \epsilon, C_i + \epsilon]$.

The choice of the threshold ϵ induces the phenomenon of uncertainties and imprecisions in the proposed detection system. In fact, the value of the threshold affects the detection rate (precision) and the conflictual situations among the colors (uncertainty).

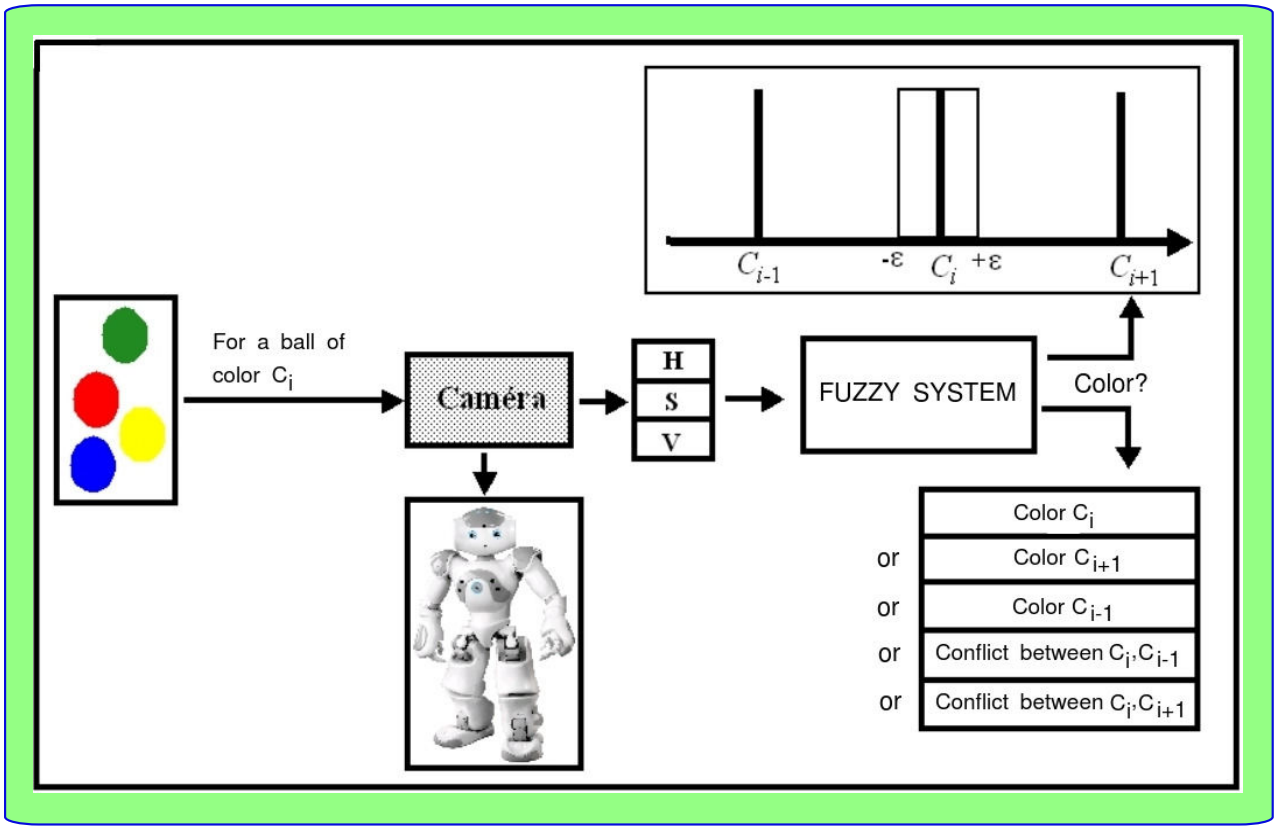

FIGURE 2.16: Threshold value for the output of detection.

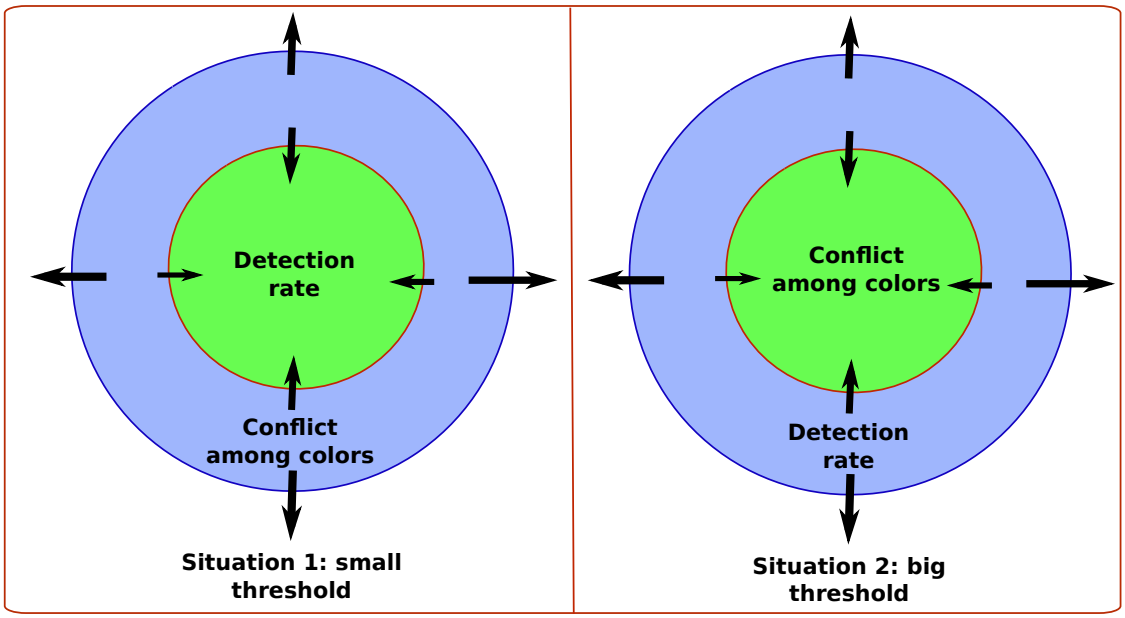

FIGURE 2.17: Threshold value and its compromise.

	Detection Rate	Imprecision in detection	Conflict among colors	Uncertainty in detection
Situation 1: $\epsilon \downarrow$	$\downarrow$	$\downarrow$	$\uparrow$	$\uparrow$
Situation 2: $\epsilon \uparrow$	$\uparrow$	$\uparrow$	$\downarrow$	$\downarrow$

TABLE 2.4: The compromise of the threshold.

It is important to report here that the design of the Fuzzy system necessarily leads to results belonging to the set of possible situations:

$\{C_i, C_{i-1}, C_{i+1}, \text{conflict between } C_i \text{ and } C_{i-1}, \text{conflict between } C_i \text{ and } C_{i+1}\}$

In other words, the conflictual situations are not failed or discarded but they are considered as degenerate outputs of the Fuzzy system.

Generally, a small value of ϵ decreases the detection rate (decreases the imprecision in the detection) and increases the conflictual situations among colors (see Fig. 2.17 and Table 2.4). In this case, the reliability and the precision of the detection are better but the uncertainty in the detection is big. Indeed, the degree of certainty such that a value is in the interval $[C_i - \epsilon, C_i + \epsilon]$ is lower. In the opposite case, as illustrated in Fig. 2.17, a bigger detection threshold improves the detection strategy and weaken the conflictual cases (uncertainty) among the colors. Actually, the precision of the detection is low but the certainty associated with this decision is high. In this situation, the interval $[C_i - \epsilon, C_i + \epsilon]$ is considered as being surer and the degree of imprecision is high. This well-known paradox of imprecision and uncertainty imposes us, in the implementation of our method, a strategy of compromise in the choice of the threshold.

Because the numerical constants assigned for the colors are in a sequence of increasing numbers by 1, so ϵ should be in $[0, 0.5[$, as illustrated in Fig. 2.18.

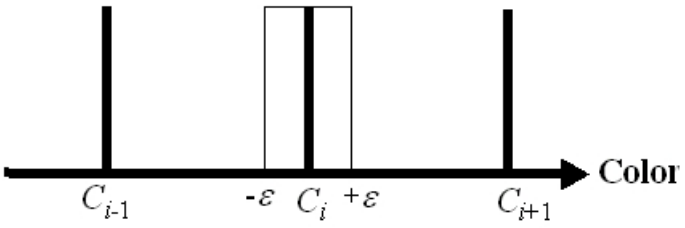

FIGURE 2.18: The interval of the threshold.

Having the threshold ϵ in mind, we formulate its influence to the decision C:

$$\begin{aligned} If\ C \in [C_i - \epsilon, C_i + \epsilon]: &\ \text{C mentions color } C_i \\ If\ C \in [C_{i-1} + \epsilon, C_i - \epsilon]: &\ \text{C hesitates between } C_{i-1} \text{ and } C_i \\ If\ C \in [C_i + \epsilon, C_{i+1} - \epsilon]: &\ \text{C hesitates between } C_{i+1} \text{ and } C_i \end{aligned} \tag{2.3}$$

In the choice of an appropriate value for the threshold ϵ, the two extreme situations $\epsilon = 0$ and $\epsilon = 0.5$ are not considered because they are non-representative in the function of the decision system. In fact, $\epsilon = 0$ leads to a detection rate mostly null and there are always conflicts. Conversely, $\epsilon = 0.5$ gives a total imprecision: not interesting.

Actually, the threshold ϵ has a great impact on the detection rate as already mentioned. Fig. 2.19 shows our test about the influence of the threshold on the average detection rate in the HSV color space of the NAO robot. The blue line is the detection rate, i.e. the ratio of the cases that the Sugeno result falls into the interval $[C_i - \epsilon, C_i + \epsilon]$ with C_i the target colored ball, we also call this the Gross Detection Rate (GDR) in Section 2.5.3. It is clear that the GDR is affected by the threshold ϵ: the bigger ϵ is, the higher GDR we have. For example when $\epsilon = 0.05$, there are only 38.33% over all the cases the NAO robot gives correct results in the interval $[C_i - \epsilon, C_i + \epsilon]$, but this values increases to 69.72% when $\epsilon = 0.5$. For that reason we can not rely on the value of GDR to demonstrate the quality of the Fuzzy system, and we need to introduce another concept: Reliable Detection Rate (RDR) which is depicted as the red line. This rate indicates the reliability about the result of the Fuzzy system. When the value ϵ is small, the detection rate decreases but the reliability of the results are bigger. For example when $\epsilon = 0.05$, the reliability is 43.57%, and this value is 0 when $\epsilon = 0.5$. We talk about the relation between GDR and RDR right later.

2.5.2 The Influence of Uncertainties and Imprecisions

In our work of the color detection, we confront the imperfection coming from two sources (see Fig. 2.20). The first one is the imprecise nature (quantitative), induced essentially by the device of measurements applied to capture information (performance and quality of cameras) and/or the techniques used to extract and manipulate measured information, i.e:

- The numerical Fuzzy system and the choice of threshold.

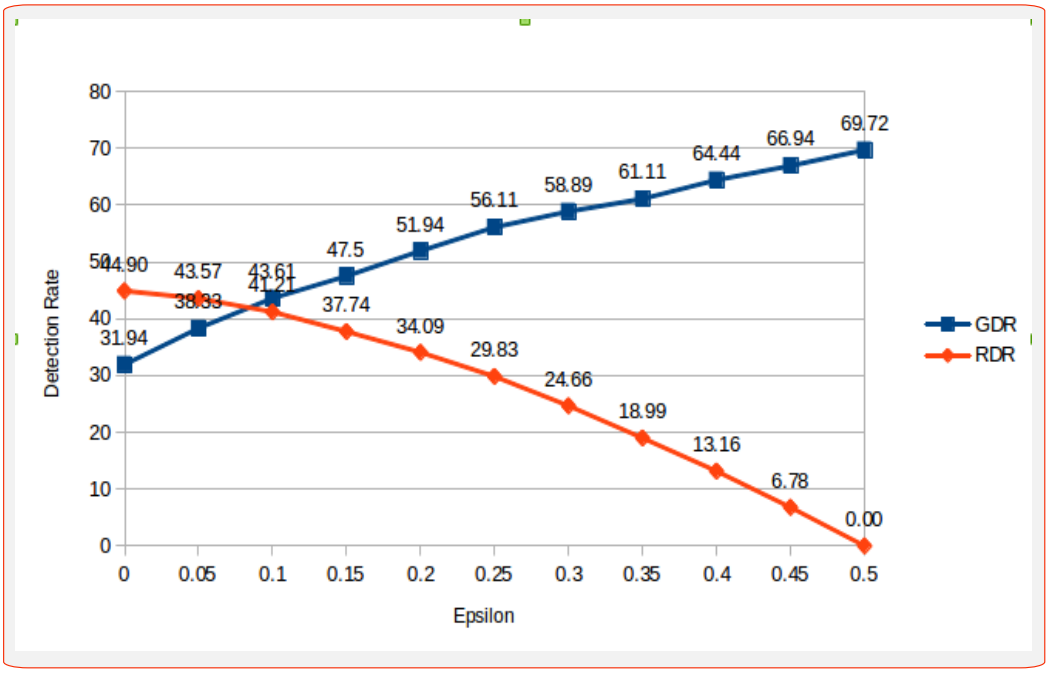

FIGURE 2.19: The influence of threshold in detection rate of the NAO robot with the HSV color space.

- The exploited technique for processing and extraction (for example the Hough transformation used to extract the balls in images).

The second imperfection, described as uncertainty, refers to a qualitative defect and the reliability of the captured information and its suitability (or compliance) with the reality. It is generally induced or imposed by the exploited environment and its uncertainty and the operational conditions of the NAO robot (lighting conditions, viewing angles of the robot while it moves and the overlap between the neighbour colors).

The conflictual situations among colors discussed previously are inherent because of the presence of these imprecisions and uncertainties. Fig. 2.21 shows an example of uncertainty in a conflictual situation where a red ball is captured three times in three different lighting conditions and it gives three different results of the Sugeno Fuzzy system. The color space is Lab and the threshold $\epsilon = 0.1$. According to the figure, the first case (3.66) is considered as a hesitation between pink and red, the second case is certainly considered as red, whereas the last case is an uncertainty between red and brown. Fig. 2.22 shows another example of imprecision in a conflictual situation in the HSV color space with $\epsilon = 0.15$ and 0.10. The two cases give two different results, while the first is considered certainly red, the second is a hesitation between red and brown.

In a general way, one of the main causes for the uncertainty of information derives from its imprecision. In fact, the previously mentioned imprecisions can train or reinforce the uncertainty on the decision (the conflict and the ambiguity among colors). By the same way, the uncertainty on the information exploited can induce the imprecision on the

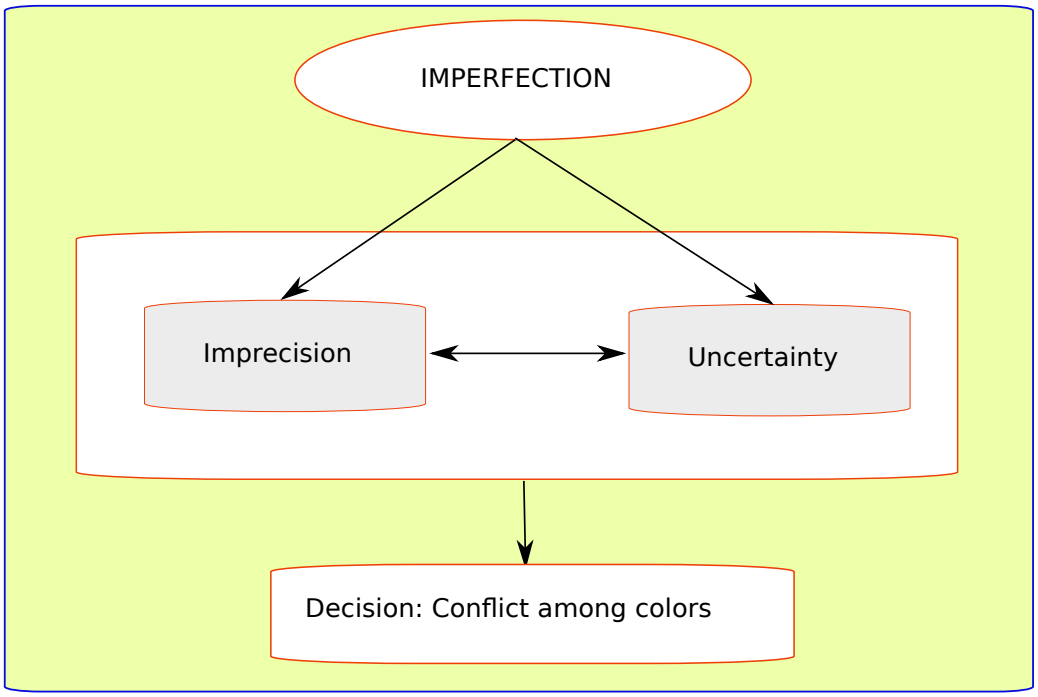

FIGURE 2.20: The relation of Imprecision and Uncertainty.

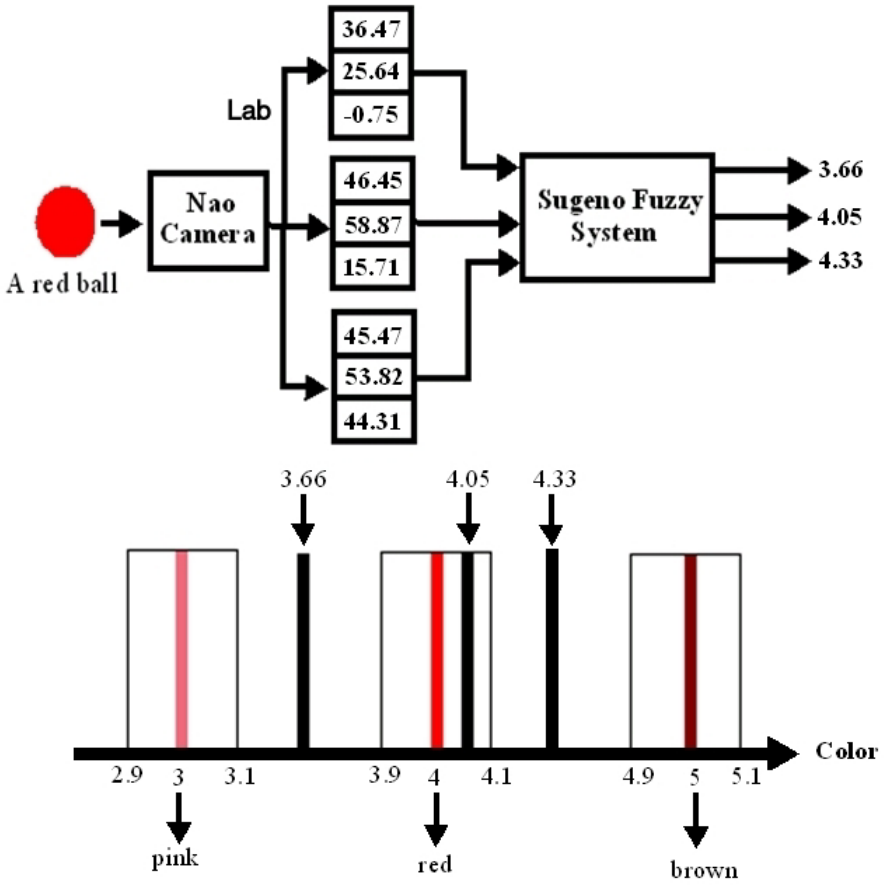

FIGURE 2.21: Conflictual situation caused by changing lighting conditions.

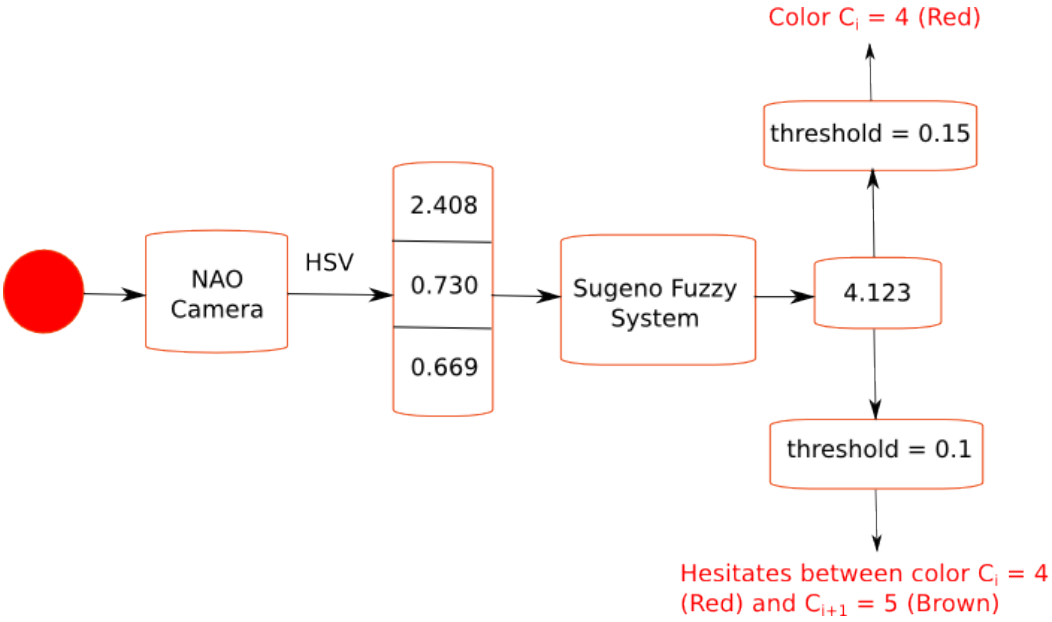

FIGURE 2.22: Conflictual situation caused by different values of the threshold ϵ.

consequences of its processing (the lighting conditions will influence the Hough transformation, for example). Generally, these two imperfections are related: if the imprecision increases then the uncertainty decreases and vice-versa.

2.5.3 The Quantification of Reliability of the Detection System

In our work, the reliability of the system has a strong relation with the detection rate which is in turn affected by the value of the threshold ϵ. Indeed, the reliability of the detection is expressed in the function of threshold:

$$RDR = 2(0.5 - \epsilon).e^{\frac{GDR-1}{\lambda}} \tag{2.4}$$

where RDR and GDR are Reliable Detection Rate and Gross Detection Rate, as already explained in Section 2.5.1. λ is a parameter which helps us justify a good and reasonable relation between RDR and GDR. In this work, we chose $\lambda = 0.85$ which gives us close values between GDR and RDR when $\epsilon = 0$.

Consider the case when $\epsilon = 0$, so $RDR = e^{\frac{GDR-1}{0.85}}$. In this case, if $GDR = 30\%$ then $RDR = 43.88\%$. It means that if we consider only the precise values (color codes) for the decisions (because $\epsilon = 0$), and we gain only 30% as the gross detection rate, we obtain 43.88% as the reliable detection rate. When $\epsilon = 0$ and $GDR = 100\%$, we have $RDR = 100\%$ too. On the other hand, when $\epsilon = 0.5$, we have $RDR = 0$ no matter what value of GDR is, meaning that there is no reliability in that case.

In general, with the same values of GDR, the lower value of ϵ is, the higher value of RDR we have (increasing reliability), and vice versa. The main objective of the work is to increase the reliability, it means that we should choose a small value of ϵ. The value

$\epsilon \in [0.0, 0.1]$ should not be chosen because in this situation we have too much cases of uncertainty. From the experimental study with various lighting conditions, we found that an appropriate value of ϵ is a numerical number in the interval $[0.1, 0.2]$, which allows a balance between imprecision and uncertainty, and guarantee a good reliability.

2.6 Experimental Study

In order to study the efficiency of the Sugeno Fuzzy system in the context of color detection, we tested with real colored balls in an indoor environment. The NAO robot used one camera on its head to capture the surrounding environment. We prepared colored balls in one of the 9 colors: Blue, Purple, Pink, Red, Brown, Orange, Yellow, Green, and Cyan and put each of them in front of the robot, then demanded the robot to capture the ball. For the detail of the implementation, we can refer to Appendices A and B. Testing

information:

- Number of colors: 9
- Number of tests each color: 40

To challenge uncertainties and imprecisions, we provide for each tested color a wide variation of the colored ball. For example, there are many kinds of red color such as "light red", "fire brick", "maroon"... but they all imply "red". Moreover, we also tested with changing lighting conditions to see the effect.

We use the Hough transformation to extract the balls' shapes in the images. The average pixels of balls are used as inputs for the Sugeno Fuzzy system, as indicated in Section 2.4.1. In addition, we tested with both HSV and Lab color spaces whose results are shown in Table 2.5 and Table 2.6, respectively.

In these validations, we tested with two different values of the threshold, and we show here with $\epsilon = 0.1$ and $\epsilon = 0.2$. As discussed in Section 2.5.1, the higher value of ϵ is, the higher detection rates of colors are, and vice versa. However it is opposite when we consider the reliable detection rate.

Table 2.5 show the results in the HSV color space. For each value of ϵ, we present both the Gross Detection Rate and Reliable Detection Rate. We gained only 43.61% of correct results in average when $\epsilon = 0.1$. For $\epsilon = 0.2$, the robot gave 51.94% of correctness.

The detection rates in the Lab color space in Table 2.6 show lower values than the HSV color space in average. It is also clear that the detection rate also increases when the value of ϵ increases.

The rates that you found in the tables are not high but it is reasonable because as already mentioned before, with these small values of the detection threshold ϵ, the rates are small too, however it is interesting for us to resolve these imprecisions and uncertainties.

Color	$\epsilon = 0.1$		$\epsilon = 0.2$	
	GDR	**RDR**	**GDR**	**RDR**
Blue	67.50%	54.58%	70.00%	42.16%
Purple	42.50%	40.67%	42.50%	30.50%
Pink	40.00%	39.49%	62.50%	38.60%
Red	67.50%	54.58%	77.50%	46.05%
Brown	37.50%	38.35%	52.50%	34.31%
Orange	37.50%	38.35%	42.50%	30.50%
Yellow	27.50%	34.09%	35.00%	27.93%
Green	67.50%	54.58%	72.50%	43.42%
Cyan	5.00%	26.26%	12.50%	21.43%
Average	**43.61%**	**41.21%**	**51.94%**	**34.09%**

TABLE 2.5: Color Gross Detection Rate and Reliable Detection Rate in HSV by the NAO robot with different values of threshold.

Color	$\epsilon = 0.1$		$\epsilon = 0.2$	
	GDR	**RDR**	**GDR**	**RDR**
Blue	10.00%	27.75%	25.00%	24.83%
Purple	15.00%	29.43%	22.50%	24.11%
Pink	25.00%	33.10%	30.00%	26.33%
Red	57.50%	48.52%	70.00%	42.16%
Brown	15.00%	29.43%	25.00%	24.83%
Orange	5.00%	26.16%	12.50%	21.43%
Yellow	27.50%	34.09%	40.00%	29.62%
Green	45.00%	41.89%	47.50%	32.35%
Cyan	70.00%	56.21%	75.00%	44.71%
Average	**30.00%**	**35.11%**	**38.61%**	**29.14%**

TABLE 2.6: Color Gross Detection Rate and Reliable Detection Rate in LAB by the NAO robot with different values of threshold.

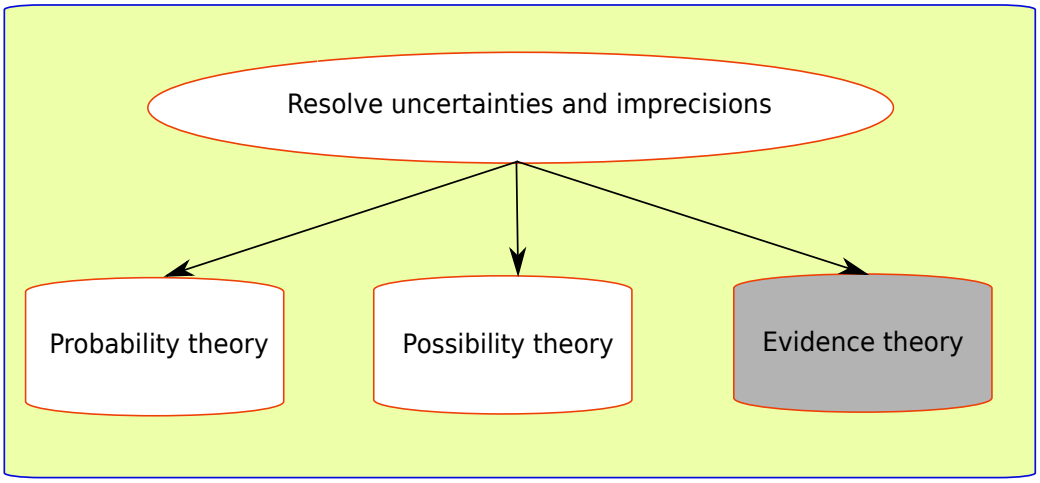

FIGURE 2.23: Well-known models for resolving uncertainties and imprecisions.

2.7 About the Improvement of the Performance for the Detection System

In order to remove ambiguities and conflicts among colors (due to imperfections) and improve the decision system, we propose the introduction of the additional information sources (cameras). In this case, the conflict is naturally reinforced since several imperfect sources join. In fact, combining the information captured by several sources which are often imperfect and sometimes contradictory certainly appears a conflict among these sources. The Evidence theory then offers one of the most suitable formalism to information fusion in which the consideration of the conflict is paramount. Of course, in the literature, many theories have been proposed to manage and manipulate the uncertainties and the imprecisions as well as the inherently conflictual situations (the Possibility theory, the Probability theory, the Evidence theory) (Fig. 2.23).

In the context of our work, the advantages and the conveniences of these theories were analysed. Our objective is not to draw a comparison in performance among these theories but to extract the most suitable tools for our work. In this case, the Evidence theory is taken into account as the most preferable one.

2.8 Conclusion of Chapter

In this chapter, we introduce the NAO robot and consider a scenario in which the robot operates with only a single sensor. We also demonstrate the difficulties of using only one source when dealing with uncertainties and imprecisions.

In the considered scenario, a NAO robot is requested to find a ball with a specific color. There are 9 colors tested in the work and each of them is named linguistically. Some previous works about color recognition and detection have been studied, such as probability-based methods, Neural Network, or Bayesian approaches, but we choose the

Sugeno Fuzzy system due to its emergent advantages. The color space is also taken into account since it can affect the results. The RGB color space is found to be less efficient than the HSV and the Lab color space, so we tested with the latter.

The balls are extracted in images by using the Hough transformation, and the average pixels of balls are taken as inputs for the Sugeno Fuzzy system. Each component in the HSV (or Lab) color space is transferred to linguistics labels, then we infer an output color for each possible combination of the three components H, S, V (or L, a, b). The final step is to use a weighted average calculation to derive the color name.

A very important consideration in this chapter is the value of the threshold ϵ which affects directly to the detection rate. This value is decimal number in the interval $[0, 0.5]$ indicating the level of certainty about a color decision. The higher this value is, the higher detection rate we have, and vice versa. We prefer to use a low value of ϵ to avoid the uncertainties between colors. However, by that way it may decrease the detection rate. In addition, other factors such as the lighting conditions, sensor's quality, the similarity among close colors also lead to uncertainties and imprecisions.

In order to improve the quality of detection, the joining of more information sources is necessary to reduce the mentioned problems. These sources give their own information about the color of the same detected target (ball), then a method of fusion is employed to fuse these information to give a final decision (a better one). The next chapter will be the demonstration of how a fusion method such as Evidence Theory can improve much more the quality of decision, applied in this scenario.

Chapter 3

Fusion of Homogeneous Sensors Data

In the previous chapter, we proposed a scenario in which a NAO robot moves in a smart environment and communicates with human. The robot is requested to find an object whose color is described in human terms. To facilitate the capability of color detection, we applied the Fuzzy Sugeno system in the decision making process of the robot, after having a look at existing works. Although the Lab color space shows a lower detection rate than HSV according to the previous chapter, we still use both of these two color spaces to have another comparative view in the fusion. The system takes the average HSV or Lab values of the detected object as the inputs, then produces a numerical number indicating a color name. That chapter also opens an interesting discussion about the choice of ϵ, the threshold of the detecting decision. However, from the experimental works and analyses, we found that the detection system remains many problems of uncertainties and imprecisions, and a single information source (the NAO camera) cannot resolve those problems.

For that reason, this chapter considers a solution for improving the reliability of the color detection of the NAO robot, by reducing uncertainties and imprecisions. Adding more information sources is considered to be a good choice, so a multi-camera system to detect colored objects will be the focused point of this chapter. Indeed, to process the color information, we only need to work with 2D data, so we consider using multiple 2D cameras (homogeneous sensors). Since we have to integrate the information coming from many sources, finding a good fusion method is mandatory, and we will discuss about the Dempster-Shafer theory as the most appropriate one in this work.

The organisation of the chapter is as follow. Section 3.1 introduces the context of using multiple homogeneous sensors for the work. After that, the principle of the Dempster-Shafer theory is reviewed in Section 3.2, which is a good starting point for Section 3.3 where we show the principle of the detection method as well as illustrative examples. Section 3.4.2 gives experimental results and an application to validate the work. Finally Section 3.5 concludes the chapter.

3.1 The Color Detection Using Multiple Homogeneous Sources

In this chapter, we continue considering the color detection for the NAO robot as introduced in the previous chapter. In fact, the NAO robot finds it difficult to have a high efficiency when detecting the colored targets, even though the Fuzzy system is a very

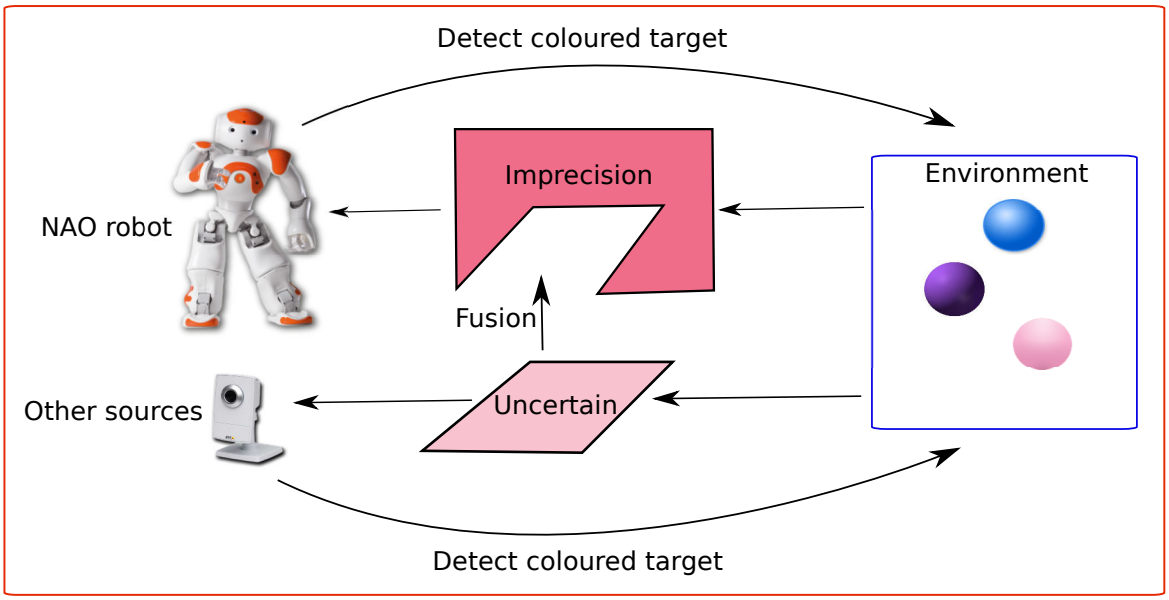

FIGURE 3.1: Multiple sources to detect colored targets.

good candidate for making decisions. There are many factors that affect to the reliability of the system such as:

- Exploited environment: change of lighting conditions, appearances of other objects...
- Sensor's quality.
- The influence of the detection threshold ϵ (a small value brings precisions but introduces uncertainties).

Facing those difficulties, we found that using only one information source (the NAO camera) is not sufficient to satisfy the reliability requirement of the color detection system. From that, we propose adding more information sources to improve the quality of results, by fusing informations coming from these sources (see Fig. 3.1).

As the matter of fact, the fusion of multiple information sources brings so many advantages that a single-source system does not have:

- Robustness and Reliability
- Higher confidence
- Reduce uncertainty
- Increase precision

Beside that, it also remains some drawbacks:

- Resources management: adding more sensors means that we have to manage more resources and it costs the system. Moreover, for real-time systems, the performance (execution time) can be limited because we have to take into account the communication among sources and the time to process more information.

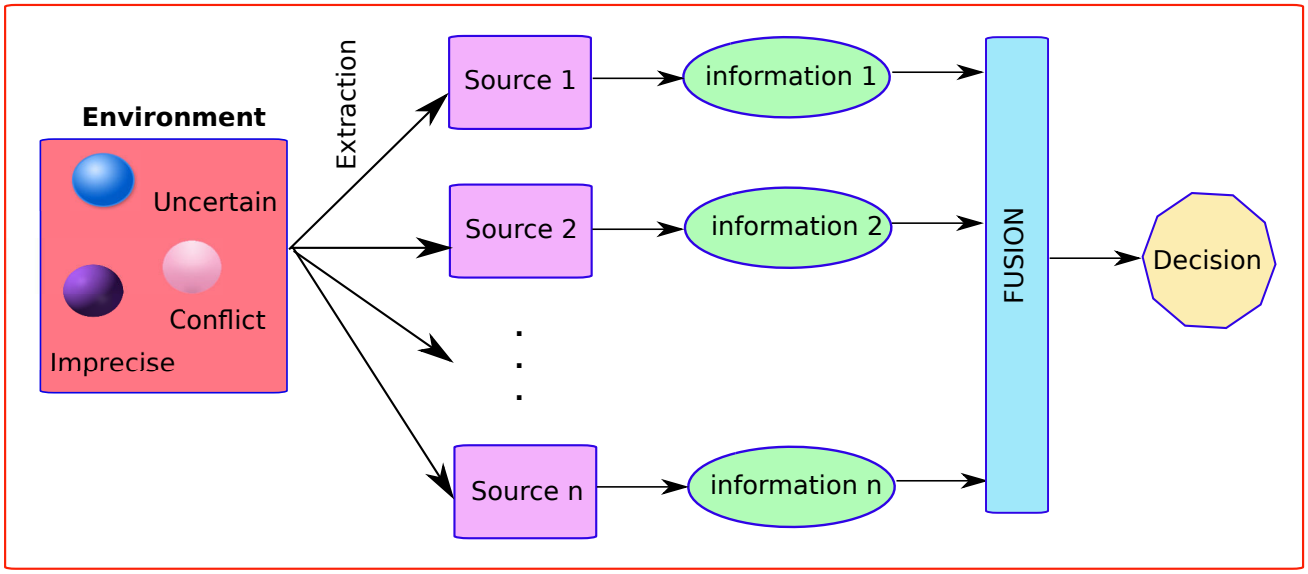

FIGURE 3.2: A generalized multiple-source system.

- Generating conflicts: using multiple sources means dealing with conflicts because at the same moment, each source may give a different result, and our mission is to resolve these conflicts.

Generally, a system can employ any number of information sources to improve its performance. Fig. 3.2 depicts an overview of the context with n sources. To achieve the objective, each source extracts information from the environment (here the cameras capture images) which contains imprecise and uncertain data. After that, each source produces its own processed information, then these informations are combined together (the fusion step) to give the final decision which is expected to be more precise and coherent.

As explained above, when applying multiple information sources, we have to face with conflicts among these sources. Particularly, in the color detection, a colored target may be recognized differently depending on the quality of sensors, the viewing angles of cameras, and the change of light... Fig. 3.3 shows a particular case where the NAO robot and an IP camera recognize the same ball but they give different names for the color (pink and red). If the two camera sensors strongly confirm their own decision, then we have totally conflictual case.

Since each information source can give different results and is conflictual with others, so we have to consider their reliabilities. By taking into account the quality of the detection for each source, we decrease the degree of belief to the output of that source, making a consideration on other results partially.

From the above discussion, we can summarize the problems that we have to face with: uncertainty, imprecision, and conflict. These difficulties affect the quality of a system, especially in our case of the color detection using multiple information sources. To deal

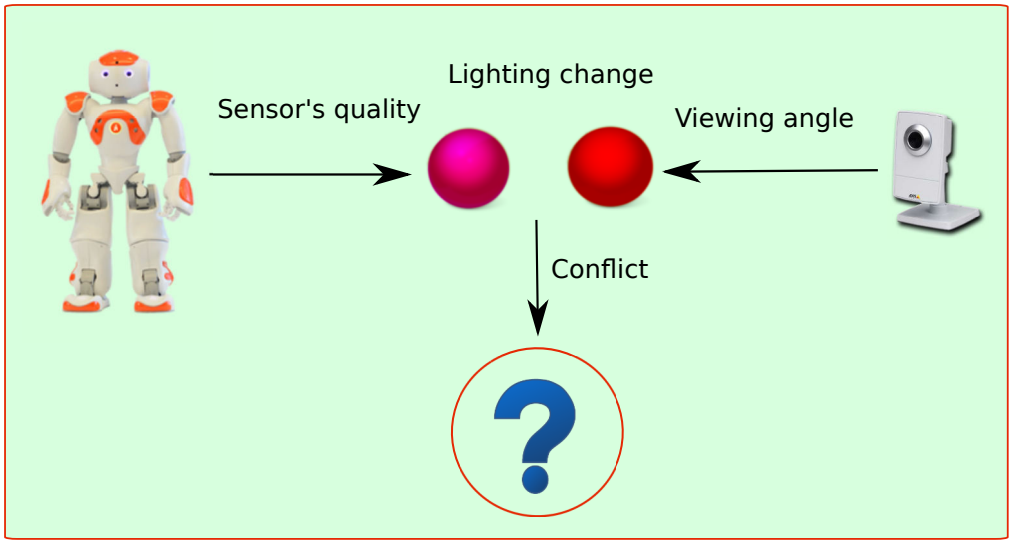

FIGURE 3.3: Fusion of cameras data may cause conflicts on the color detection.

with such problems, several fusion methods have been proposed. The most emergent approach that is feasible for our case is the Dempster-Shafer theory. In fact, the theory is able to express the uncertainty of information in terms of pieces of evidence, then it provides operators to combine these uncertain informations coming from different sources in order to derive a precise decision. Initially, this theory is considered to be appropriate for the combination of reliable sources, however, some discounting techniques which take into account the reliabilities of sources allow the method to combine unreliable sources, from that reduce conflicts and give more coherent results.

The Dempster-Shafer theory has been applied in many existing works, showing its powerful influence in the domain of information and sensor fusion. There are some works using multiple cameras apply the Dempster-Shafer theory to fuse information. For example, in [51], a people tracking system by using multi-camera is introduced, or [54] employes this theory for the improvement of X-rays castings inspection reliability. Actually, in literature, according to our research up to the time of this thesis, there has been no work using multi-camera for the color detection and recognition with the Dempster-Shafer theory. However, since this approach was applied in many works of fusion with great successes, and from the analyses above, it motives us to take the advantages of this theory and apply them in our work. First of all, we have an overview of the Dempster-Shafer theory.

3.2 Background of the Dempster-Shafer Theory

Mentioned from the first publish in [14] and in [64], the Dempster-Shafer Theory (DST or Evidence Theory) has become a very famous framework for information fusion, solving the problems of uncertainties and imprecisions. It is considered as a generalization of the Bayesian theory of subjective probability with belief functions representing degree of belief for one question based on the probabilities of a related question.

When used for sensor fusion, the theory allows various operators/rules which combine multiple information sources, taking into account their degree of belief represented as mass functions (or mass). Probabilities values are assigned to sets of possibilities rather than single events. So if we denote Ω the space of discernment which contains possible decisions:

$$\Omega = \{C_1, C_2, ..., C_N\} \tag{3.1}$$

We then have the power set comprising all possible hypotheses (subsets):

$$2^{\Omega} = \{\emptyset,\ \{C_1\},\ \{C_2\},\ ...,\ \{C_N\},\ \{C_1 \cup C_2\},\ ...,\ \{C_1 \cup C_N\},\ ...,\ \Omega\} \tag{3.2}$$

In the Evidence Theory, we have to determine a mass function which describes the degree of belief for all possible hypotheses in the power set. This function satisfies:

$$\begin{aligned} &m : 2^{\Omega} \rightarrow [0, 1] \\ &\sum_{H \in 2^{\Omega}} m(H) = 1 \end{aligned} \tag{3.3}$$

From the mass function, we are able to determine the belief function:

$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B), \forall A \subseteq \Omega \tag{3.4}$$

where we have $bel(A)$ representing the total share of belief supporting the hypothesis A.

We can also derive the plausibility function:

$$pl(A) = \sum_{\emptyset \neq B \cap A} m(B), \forall A \subseteq \Omega \tag{3.5}$$

where $pl(A)$ describes the maximum share of belief that supports the hypothesis A.

The most interesting point of the Dempster-Shafer theory is that it allows combining several masses derived from multiple sources by applying a combination operator. Initially, the first operator was introduced by Dempster and reprised by Shafer and it is a normalised combination rule. However there have been several rules proposed and we list here some that are interesting for this work.

- Conjunctive combination:

$$m_{Conj}(H) = \sum_{H_1 \cap H_2 \cap ... \cap H_s = H} \prod_{j=1}^{s} m_j(H_j) \tag{3.6}$$

where $m_{Conj}(H)$ denotes the conjunctively combined mass value at the hypothesis $H \in 2^{\Omega}$, and $\{H_j\}$ is the set of hypotheses of the sources that have an agreement on only the hypothesis H, that is the place where the word "conjunctive" comes from. In this rule, we suppose that all the sources must be reliable.

- Disjunctive combination:

$$m_{Disj}(H) = \sum_{H_1 \cup H_2 \cup \ldots \cup H_s = H} \prod_{j=1}^{s} m_j(H_j) \tag{3.7}$$

where $m_{Disj}(H)$ is the disjunctively combined mass value at the hypothesis $H \in 2^{\Omega}$, and $\{H_j\}$ represents the set of hypotheses of the sources whose union forms exactly the hypothesis H. By this rule, we consider that each information source donates one portion about the existence of H, and we assume that at least one of the information sources are reliable.

- Dempster-Shafer combination ([64]):

$$\begin{aligned} m_{DS}(H) = \frac{1}{1 - m_{Conj}(\emptyset)} m_{Conj}(H), H \neq \emptyset \\ m_{DS}(\emptyset) = 0 \end{aligned} \tag{3.8}$$

The conflict $k = m_{Conj}(\emptyset)$ among the sources are considered as the mass value derived by the conjunctive operator on the empty set. Thus it is clear that the lower value of k, the more agreement the sources have together, and vice versa. If $k = 1$, we say that the sources are totally conflictual, and when $k = 0$, the sources totally agree together. The Dempster-Shafer combination is a conjunctively normalised rule. This normalization is interesting only in a closed world (the discernment space) and is applied for sources that are reliable and non-conflictual.

- Yager combination ([75]):

$$\begin{aligned} m_{Yager}(H) = m_{Conj}(H), H \neq \emptyset, H \neq \Omega \\ m_{Yager}(\Omega) = m_{Conj}(\Omega) + m_{Conj}(\emptyset) \\ m_{Yager}(\emptyset) = 0 \end{aligned} \tag{3.9}$$

In this rule, Yager transfers the global conflict into the total ignorance, i.e. to the mass value of Ω in order to stay in the closed world and consider that we know nothing in the case of conflict. This combination rule is better adapted when there are non-reliable sources.

- Florea combination ([24]):

$$\begin{aligned} m_{Florea}(H) &= \beta_1(k).m_{Dis}(H) + \beta_2(k).m_{Conj}(H) \\ \beta_1(k) &= \frac{k}{1-k+k^2}, \beta_2(k) = \frac{1-k}{1-k+k^2} \\ k &= m_{Conj}(\emptyset) : \text{Conflict among sources} \end{aligned} \tag{3.10}$$

  This interesting rule considers the weighted sum of the two rules: conjunctive and disjunctive as the function of the global conflict $k = m_{Conj}(\emptyset)$. By this way, when k goes to 1, the rule focuses on the disjunctive part, and when k downs to 0, the rule considers only the conjunctive part. Remind that by taking into account the disjunctive combination, this rule can be well adapted in the case of non-reliable sources.

- Dubois-Prade combination ([17]):

$$m_{DB}(H) = m_{Conj}(H) + \sum_{\substack{H_1 \cup H_2 \cup \ldots \cup H_s = H \\ H_1 \cap H_2 \cap \ldots \cap H_s = \emptyset}} \prod_{j=1}^{s} m_j(H_j) \tag{3.11}$$

  In fact, this rule is interesting only when the information given by the sources are only on the singletons. This rule proposes a more delicate management of conflict by distributing the partial conflict into the partial ignorances. This rule can also be better adapted with non-reliable sources.

At the final step of fusion by the Dempster-Shafer theory, we have to derive a singleton which is expected as the best output. To do that, we have several decision methods, and the three most well-known are: the maximum of belief, the maximum of plausibility, and the maximum of pignistic probability.

- The maximum of belief:

  The decision is given by the singleton hypothesis that has the maximum degree of belief:

$$D = A \mid bel(A) = max(bel(X)), X \in \Omega \tag{3.12}$$

  where $bel(A)$ is the value of belief function with the hypothesis A. This method of decision is said to be too pessimistic since it concentrates on only each singleton hypothesis without considering their presence in compounding hypotheses.

- The maximum of plausibility:

  The decision is given by the singleton hypothesis that has the maximum degree of plausibility:

$$D = A \mid pl(A) = max(pl(X)), X \in \Omega \tag{3.13}$$

  where $pl(A)$ is the value of plausibility function with the hypothesis A. In contrast with the maximum of belief, this decision method is too optimistic because it takes into account the presence of each singleton hypothesis in all the hypotheses.

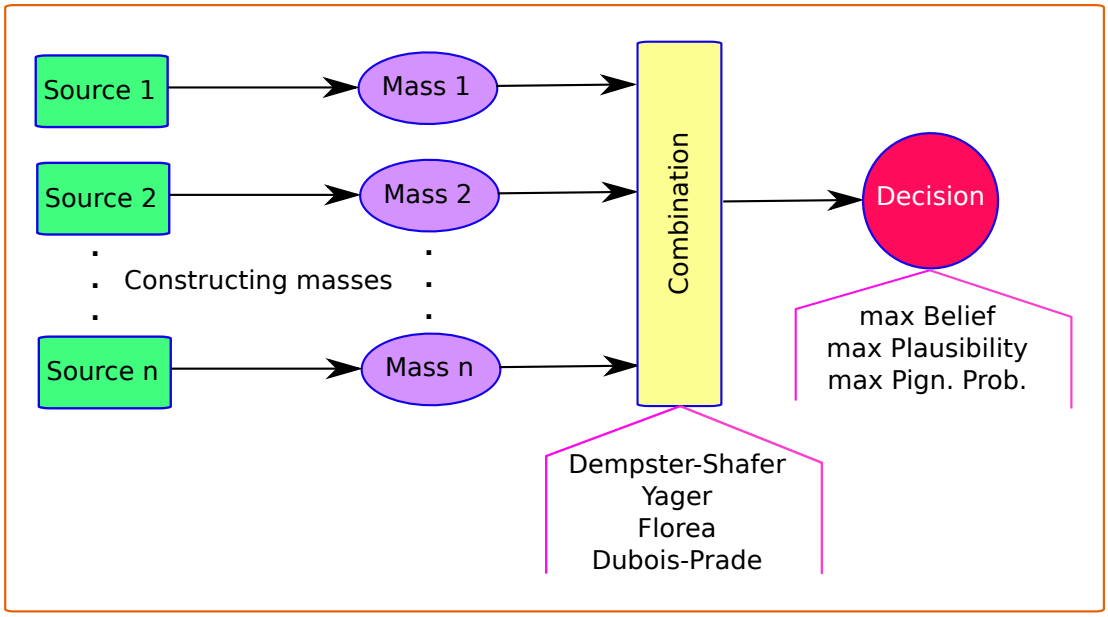

FIGURE 3.4: Applying the Dempster-Shafer theory.

- The maximum of pignistic probability:

  In fact, Smets ([65]) proposed transforming the mass function to a probability function named $BetP$ defined on Ω which is formalized as following:

$$BetP_m(X) = \frac{1}{1 - m(\emptyset)} \sum_{X \in A} \frac{m(A)}{|A|} \tag{3.14}$$

  where $|A|$ represents the cardinality of the subset $A \subseteq \Omega$. By this transformation, the mass $m(A)$ is uniformly distributed over the elements of A. From this probability distribution, we can easily take the decision by applying a classical statistically decision.

Fig. 3.4 describes the overall flow when we apply the Dempster-Shafer theory. First of all, the sources extract and process information, thereafter they construct their own mass values. The masses of the sources are then combined together in order to have only one mass which is used to derive a final decision by considering a decision method.

3.3 The Methodology for the Color Detection Using Multiple Homogeneous Data Sources

3.3.1 The Method's Principle

Overview of the Process

As discussed from Chapter 2, uncertainty can happen at any time due to many impacts from the environment or the sensors' quality, and it affects a lot to the quality of the

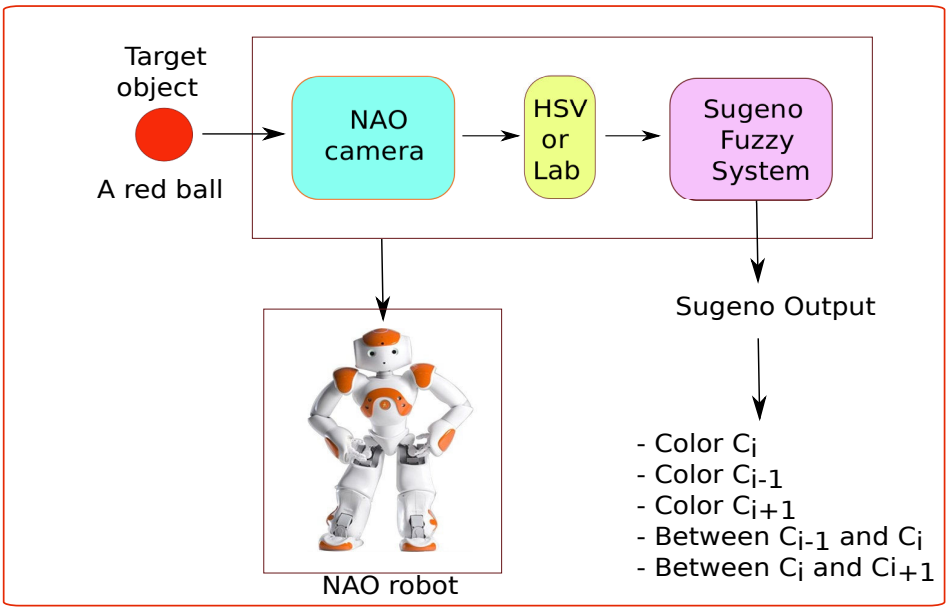

FIGURE 3.5: The cases of uncertainty.

operations. After doing many experiments under different testing conditions, we have found that in most of the cases, the uncertainty usually happens among close colors, more precisely, for a requested color being assigned a constant number C_i, the actual Fuzzy Sugeno result (see Eq. (2.2)) perceived by the camera falls into the set $\{C_{i-1}, C_i, C_{i+1}\}$. Fig. 3.5 shows the possible cases of uncertainty that can be caused. Indeed, due to the natural numbers that we assign to the colors, there are five situations in which the NAO robot may hesitate when it is requested to find a ball number C_i: the Sugeno system output possibly implies the color C_i, C_{i-1}, C_{i+1} or it cannot decide between C_i, C_{i-1} or C_i, C_{i+1}, as seen from Fig. 3.6.

For now, we have determined the possible cases of uncertainty, and we need a method that can well represent these information. As discussed above, the Dempster-Shafer theory emerges as one of the best solutions due to its advantages as explained before. In fact, the number of colors is limited to 9 in this work, however, as indicated before about the uncertain Sugeno output in the interval $\{C_{i-1}, C_i, C_{i+1}\}$, we can propose a space of discernment as the following:

$$\Omega_i = \{C_{i-1}, C_i, C_{i+1}\} \tag{3.15}$$

where Ω_i is the space of discernment associated with a requested color C_i being searched by the robot.

In Fig. 3.7, we show the overall process of the decision system. Starting from a requested color ball having the constant number C_i, the camera of NAO and the additional cameras search for balls in the same scenes. When they detect a ball, they convert the color information into Lab or HSV color space (depending on the user's selection). After that, the Sugeno Fuzzy system is used to determine the Fuzzy output of each camera. According to the above explanation, these outputs may be uncertain and imprecise, so

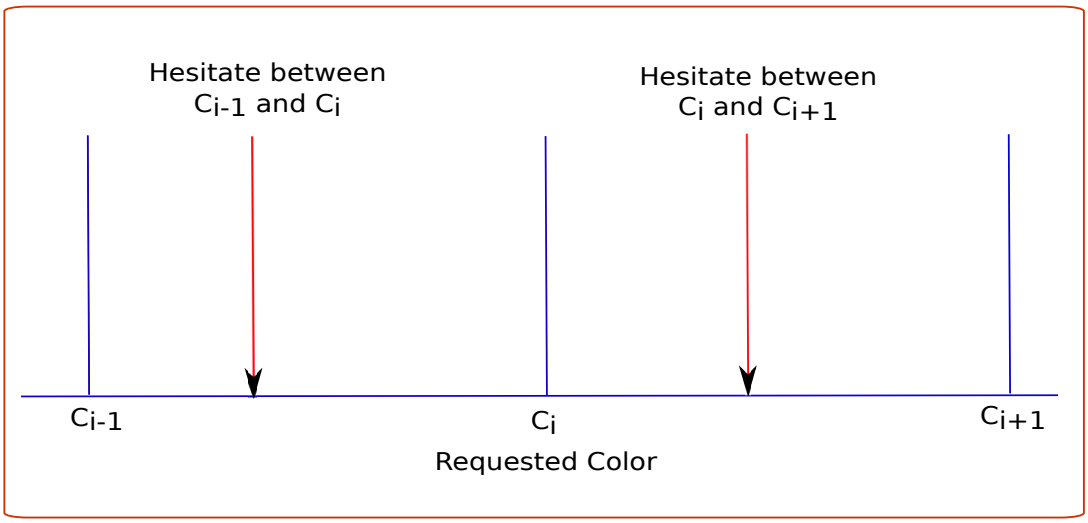

FIGURE 3.6: Uncertain cases when searching for color C_i.

we send them to a mass constructing mechanism in order to represent the uncertainties. After that, the masses are combined into a single one using a predefined combinator, then a decision method is chosen to determine the final output color which will be sent back to the NAO robot.

Constructing Mass Values

In the Dempster-Shafer theory, the difficult part is to define a strategy to construct mass values, which describes the degree of belief on the hypotheses given by the sensors. From the space of discernment defined in 3.15, we have the power set which contains all possible subsets of the space:

$$2^{\Omega_i} = \{\emptyset, \{C_{i+1}\}, \{C_i\}, \{C_{i+1}, C_i\}, \{C_{i-1}\}, \{C_{i-1}, C_{i+1}\}, \{C_i, C_{i+1}\}, \Omega_i\} \tag{3.16}$$

According to Section 2.5.1, we need to define a threshold ϵ which describes the certain interval from which we determine the output color of the Sugeno Fuzzy system. In this section, we take advantage of this threshold value to represent the degree of belief for the singleton hypotheses in the powerset. More precisely, Fig. 3.8 illustrates the use of ϵ to represent the certain interval for each color C_{i-1}, C_i, C_{i+1} as Fuzzy sets (the green lines). If the Sugeno output falls into $[C_i - \epsilon, C_i + \epsilon]$, the mass value of the hypothesis $\{C_i\}$ will be 1, otherwise we obtain a value in $0 \leq C < 1$), as demonstrated by the red lines.

In the same fashion, we can use this Fuzzification-based approach for the cases hesitating between the colors as shown in Fig. 3.9. In this strategy, the membership functions of the hypothesis $\{C_{i-1}, C_i\}$ and $\{C_i, C_{i+1}\}$ are triangles (the green lines) because we consider the value 0.5 to be the mid-point that we have a balance between two colors, and

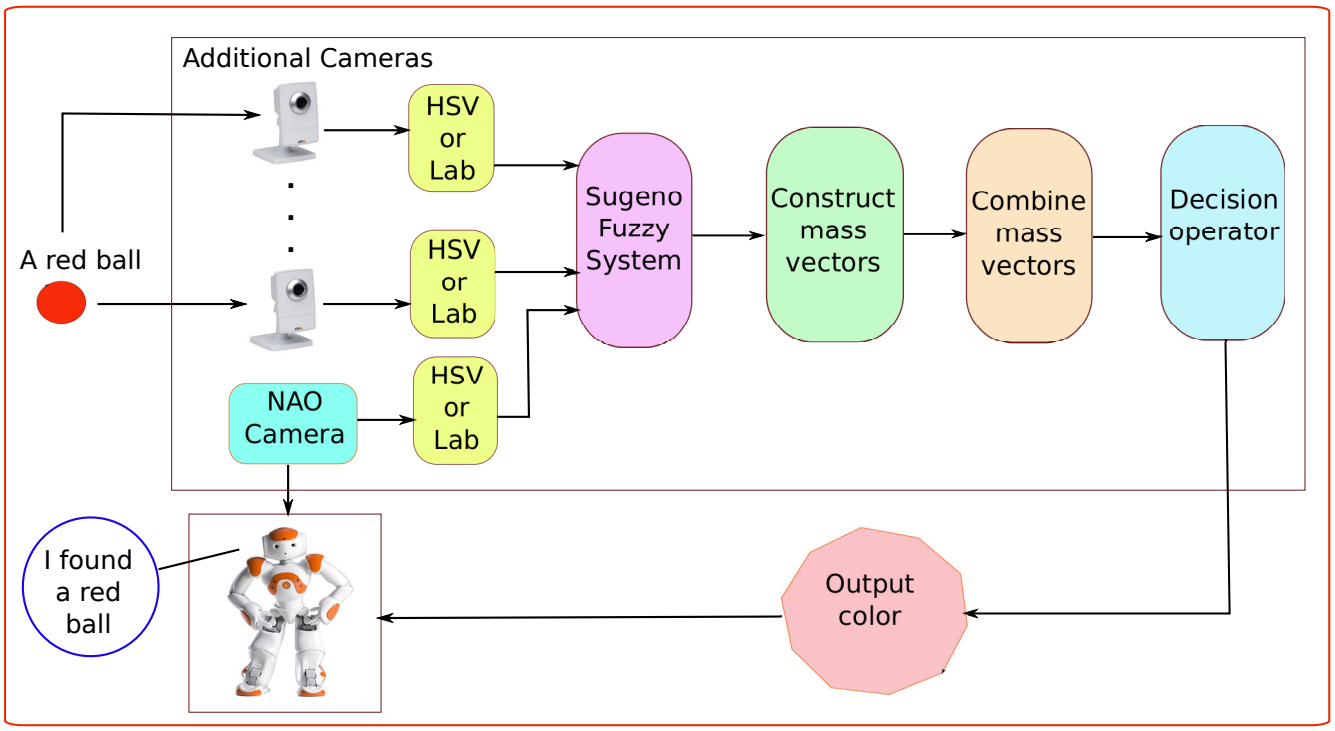

FIGURE 3.7: The overall process of the decision system

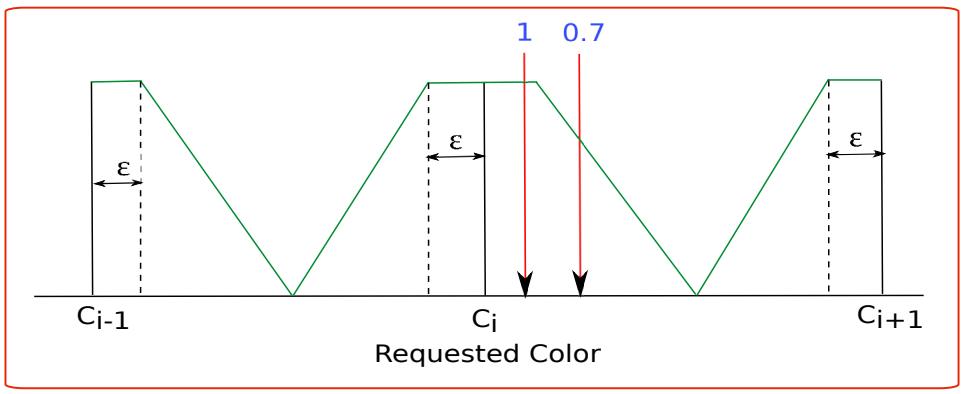

FIGURE 3.8: The Fuzzy sets represent the certainty of colors.

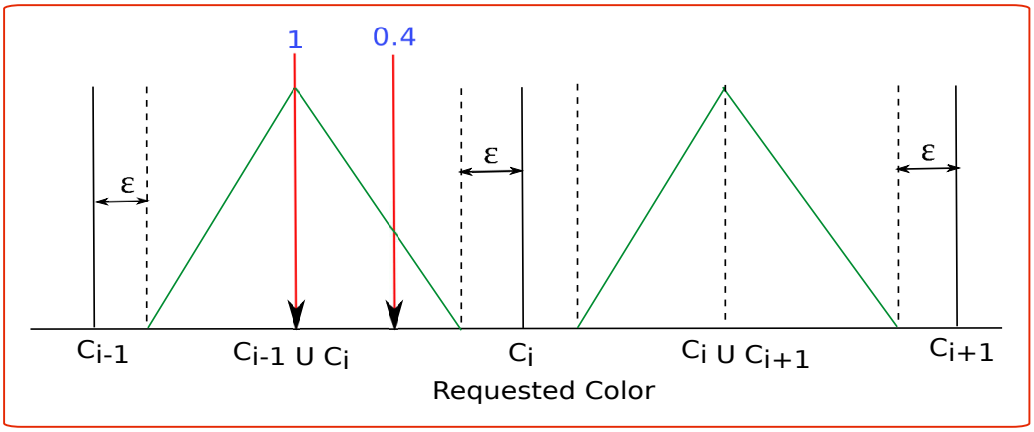

FIGURE 3.9: The Fuzzy sets represent the uncertainty of colors.

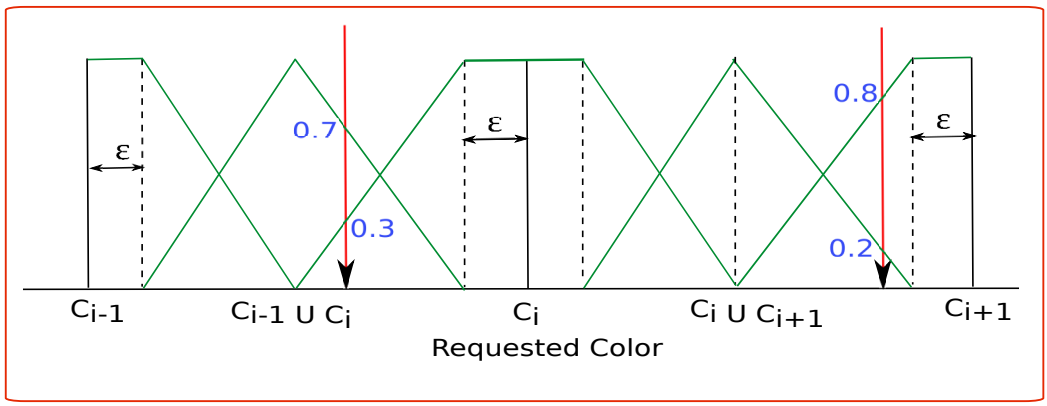

FIGURE 3.10: The mass constructing strategy based on Fuzzification.

every other value is considered to have more bias on a specific color. In the figure, the two red lines show the cases that the hypothesis $\{C_{i-1}, C_i\}$ obtains a mass value of 1 and 0.4, respectively.

With this in mind, we can obtain a strategy of building mass function taking the advantage of Fuzzification as above, by drawing membership function for each hypothesis as shown in Fig. 3.10. Indeed, for the hypotheses $\{C_{i-1}\}$, $\{C_i\}$, and $\{C_{i+1}\}$ we calculate their mass values by fuzzifying the Sugeno output values with the associated trapezoidal membership functions which are constructed based on the value of ϵ. For the hypotheses $\{C_{i-1}, C_i\}$ and $\{C_i, C_{i+1}\}$, we use the triangular functions as explained above to infer the mass values. Due to the nature of the proposed membership functions, we absolutely have: $m_i(C_{i-1}, C_{i+1}) = 0$ and $m_i(\Omega_i) = 0$. Moreover, for the cases that the Sugeno system gives outputs that are outside of the interval $[C_{i-1}, C_{i+1}]$, we do not take them into account and skip these rare situations (already experimentally validated). It means that we consider a closed world in which $m_i(\emptyset) = 0$.

In Fig. 3.10, two examples of the mass calculation are also shown. For the Sugeno output represented by the red line on the left, we have:

$$\begin{aligned} m_i(C_{i-1} \cup C_i) &= 0.7 \\ m_i(C_i) &= 0.3 \\ m_i(H) &= 0.0 \\ H \in 2^{\Omega_i},\ H \neq \{C_{i-1} \cup C_i\},\ H &\neq \{C_i\} \end{aligned} \tag{3.17}$$

And for the second case of Sugeno output (the red line on the right), we have:

$$\begin{aligned} m_i(C_i \cup C_{i+1}) &= 0.2 \\ m_i(C_{i+1}) &= 0.8 \\ m_i(H) &= 0.0 \\ H \in 2^{\Omega_i},\ H \neq \{C_i \cup C_{i+1}\},\ H &\neq \{C_{i+1}\} \end{aligned} \tag{3.18}$$

Actually, due to the above design of the membership functions for the hypotheses, with one Sugeno output, the system hesitates on at most 2 hypotheses, and we always guarantee the condition of the mass function:

$$\sum_{H \in 2^{\Omega_i}} m_i(H) = 1 \tag{3.19}$$

Up to this step, we have determined the mass vector for a Sugeno output. The next step is to combine the masses coming from different sources in order to decide the output.

Combination and Decision

One of the most interesting properties of the Dempster-Shafer theory is that it allows combining mass vectors from different sources into only one vector. The combined vector well represents the degree of belief about the hypotheses proposed from the sources. As mentioned above, Dempster and Shafer proposed the first combination operator (combinator) but there have been several other combinators which have proven their strengths.

According to a high-appreciated work in [62] from Sentz and Ferson about the Dempster-Shafer theory and their combination rules, the Dempster-Shafer rule might be the most appropriate rule to use even in a context of highly conflicting evidence as the conflict is normalized out of the combination. In fact, this rule was initially applied only for reliable sources, however Shafer reprised it and took into account the reliabilities of the sources, from that a discounting step allows the sources to combine effectively despite of conflicts.

For those reasons, in this work we consider the Dempster-Shafer's rule as the combinator to fuse the masses of the sources.

After combining the sources to get only one mass function, the next step is to employ a decision method to derive the final decision. As discussed in Section 3.2, the maximum of belief is too pessimistic and the maximum of plausibility is too optimistic whereas the maximum of pignistic probability is a compromise between the two, so in this work, the maximum of pignistic probability is considered as the method of decision.

Discounting Factor and the Reliability of Sources

Shafer in [64] introduced a tradeoff method to deal with conflicts when combining information sources. Indeed, when we meet a conflictual case among sources, we should decrease the degree of belief for each source based on their reliability first, then combine the resulting mass functions. From that a discounting factor α is introduced, which allows transferring the degree of belief of a source into the set of ignorance (Ω):

$$\begin{aligned} m_j^{\alpha}(H) &= \alpha_j.m_j(H),\ \forall H \in 2^{\Omega} \\ m_j^{\alpha}(\Omega) &= 1 - \alpha_j.(1 - m_j(\Omega)) \end{aligned} \tag{3.20}$$

where $\alpha_j \in 2^{\Omega}$ is the discounting factor (reliability) associated with the source S_j.

In this work of the color detection, α is measured statistically during the test with each camera. More precisely, it is considered as the detection rate when a camera detects the colored balls. For example to measure the discounting factor for the NAO's camera, we tested with 100 images for each color, calculate the detection rate of each color, then the discounting factor is the average value of these rates.

3.3.2 Illustrative Example

This section provides some examples to illustrate the proposed fusion steps described above. As already mentioned, when the robot is requested to find a ball with the color C_i, we suppose a space of discernment $\Omega = \{C_{i-1}, C_i, C_{i+1}\}$ since the experimental validation showed most of this case. We present two examples, one for two cameras, and one for three cameras.

Example 1: Conflict between two cameras

Suppose that the NAO robot is requested to find a color red ball, it employs another IP camera for the detection. Each camera gives its own Sugeno Fuzzy output. However, the results of the two cameras are conflictual, as shown in Fig. 3.11, which make the decision to be difficult. Fig. 3.12 shows two different images when the two cameras capture the same ball at the same time.

Fig. 3.13 shows the construction of mass vectors in this case, by considering the method presented in Section 3.3.1. The threshold ϵ is chosen as 0.05 for example.

It is interpreted from Fig. 3.13 that the first camera's output indicates a hesitation between the color $C_{i-1} = 3$ and the hypothesis $\{C_{i-1}, C_i\}$. By using the Fuzzification based construction of mass, this camera implies a portion of 0.11 on C_{i-1} and 0.89 on

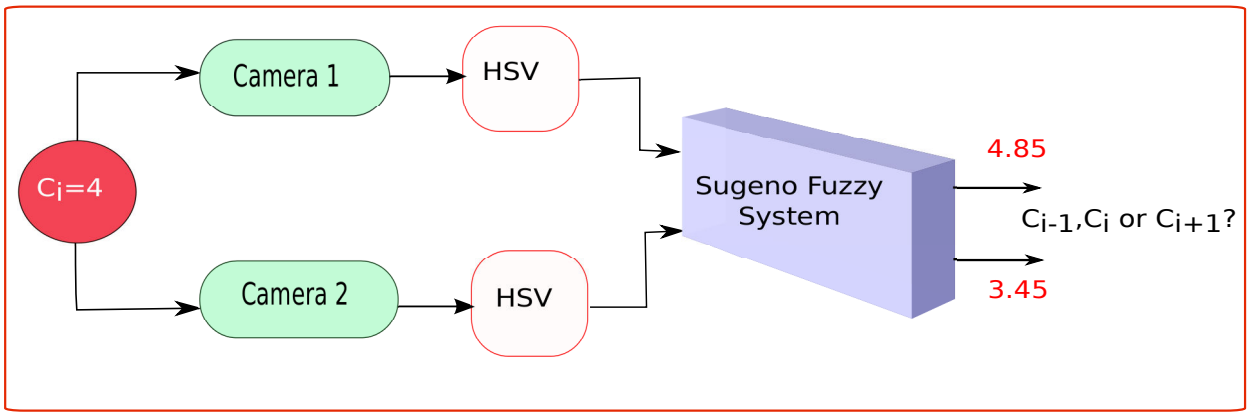

FIGURE 3.11: Conflict between two cameras

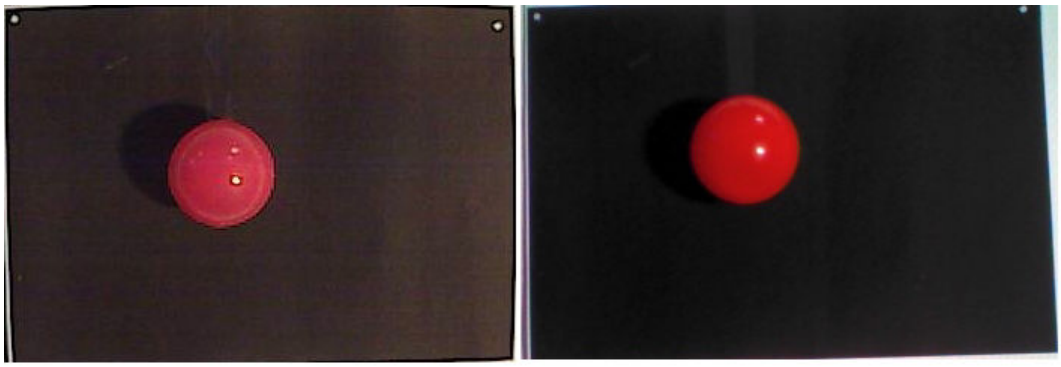

FIGURE 3.12: A ball is captured by the IP camera (left) and the NAO camera (right).

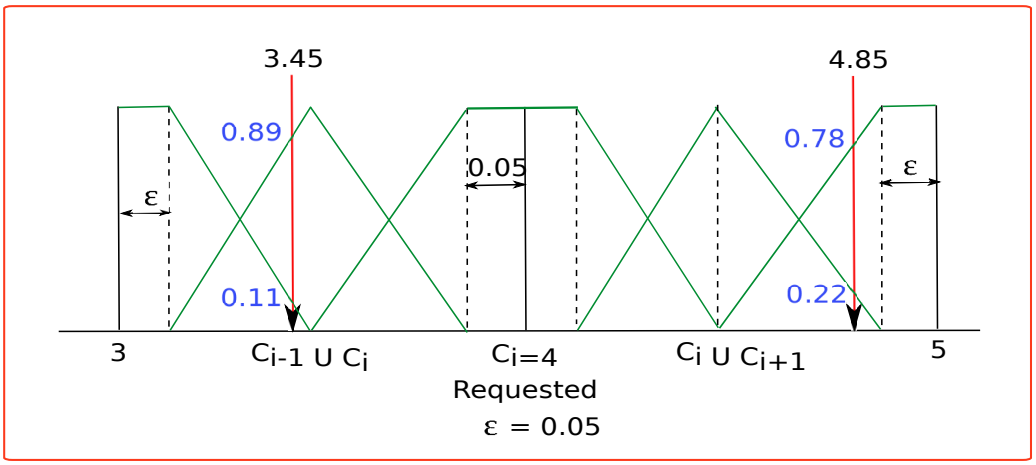

FIGURE 3.13: Construction of mass from Sugeno output.

$\{C_{i-1}, C_i\}$. In the same way, we interpret the Sugeno output of the second camera as 0.78 for $\{C_{i+1}\}$ and 0.22 for $\{C_i, C_{i+1}\}$. From that, we construct all values of masses as shown in Table 3.1.

Mass	$\emptyset$	C_{i-1}	C_i	$C_{i-1} \cup C_i$	C_{i+1}	$C_{i+1} \cup C_{i-1}$	$C_i \cup C_{i+1}$	Ω
m_1	0	0.11	0	0.89	0	0	0	0
m_2	0	0	0	0	0.78	0	0.22	0

TABLE 3.1: The mass values given by the two cameras.

The conflict k between these masses (see Eq. (3.8)) is calculated as 0.8025 and it is considered as a very high conflict.

In this situation, we suppose that the reliabilities of cameras: $\alpha_1 = 0.7$, $\alpha_2 = 0.6$. By applying the discounting factor discussed in Section 3.3.1, the mass values then will be in Table 3.2.

Mass	$\emptyset$	C_{i-1}	C_i	$C_{i-1} \cup C_i$	C_{i+1}	$C_{i+1} \cup C_{i-1}$	$C_i \cup C_{i+1}$	Ω
m_1	0	0.08	0	0.62	0	0	0	0.3
m_2	0	0	0	0	0.47	0	0.13	0.4

TABLE 3.2: The mass values given by the two cameras after discounting by reliabilities.

Next, we have a look at how some combination methods fuse these masses and give decision. It is note that because the combination of Dubois-Prade considers only on singleton hypotheses so it is not interesting in this work, thus we take into account only the combination rules of Dempster-Shafer, Yager, and Florea.

- Using the Dempster-Shafer combination

  The result of applying the Dempster-Shafer rule is shown in Table 3.3. m_{DS} is the combined mass, bel, pl, and $betP$ are the belief function, the plausibility function, and the pignistic probability, respectively. The final column shows the decisions made by these selections.

Mass	$\emptyset$	C_{i-1}	C_i	$C_{i-1} \cup C_i$	C_{i+1}	$C_{i+1} \cup C_{i-1}$	$C_i \cup C_{i+1}$	Ω	Decision
m_1	0	0.08	0	0.62	0	0	0	0.3	
m_2	0	0	0	0	0.47	0	0.13	0.4	
m_{DS}	0	0.05	0.13	0.38	0.21	0	0.06	0.18	
bel	0	0.05	0.13	0.55	0.21	0.26	0.40	1	C_{i+1}
pl	0	0.60	0.74	0.79	0.45	0.87	0.95	1	C_i
$betP$		0.30	0.40		0.30				C_i

TABLE 3.3: Dempster-Shafer combination and decision.

- Using the Yager combination

Mass	$\emptyset$	C_{i-1}	C_i	$C_{i-1} \cup C_i$	C_{i+1}	$C_{i+1} \cup C_{i-1}$	$C_i \cup C_{i+1}$	Ω	Decision
m_1	0	0.08	0	0.62	0	0	0	0.3	
m_2	0	0	0	0	0.47	0	0.13	0.4	
m_{Yager}	0	0.03	0.08	0.25	0.14	0	0.04	0.46	
bel	0	0.03	0.08	0.36	0.14	0.17	0.26	1	C_{i+1}
pl	0	0.74	0.83	0.86	0.64	0.92	0.97	1	C_i
$betP$		0.31	0.38		0.31				C_i

TABLE 3.4: Yager combination and decision.

- Using the Florea combination

Mass	$\emptyset$	C_{i-1}	C_i	$C_{i-1} \cup C_i$	C_{i+1}	$C_{i+1} \cup C_{i-1}$	$C_i \cup C_{i+1}$	Ω	Decision
m_1	0	0.08	0	0.62	0	0	0	0.3	
m_2	0	0	0	0	0.47	0	0.13	0.4	
m_{Florea}	0	0.03	0.07	0.21	0.12	0.02	0.03	0.52	
bel	0	0.03	0.07	0.31	0.12	0.16	0.22	1	C_{i+1}
pl	0	0.78	0.84	0.88	0.69	0.93	0.97	1	C_i
$betP$		0.31	0.37		0.32				C_i

TABLE 3.5: Florea combination and decision.

From the above tables, we saw that the three combination operators give the same decision results. However, the Dempster-Shafer rule looks better because the distinction between the decision C_i and the other decisions is clearer than the other rules. Indeed, using the maximum of pignistic probability, the Dempster-Shafer rule gives 0.3, 0.4, 0.3 for C_i, C_{i-1} and C_{i+1}, respectively, so the difference between the best decision and the second is $0.4 - 0.3 = 0.1$. Whereas this number for the Yager's rule is $0.38 - 0.31 = 0.07$, and the Florea's rule $0.37 - 0.32 = 0.05$.

Example 2: Conflict among three cameras

Suppose that in this case we use three cameras: one of the NAO robot, one IP camera, and we add a Web camera as the third one. The three cameras scan the environment to search the target colored ball, and they give different Sugeno results which are conflictual, as shown in Fig. 3.14. Fig. 3.15 shows a red ball captured by the three cameras at the same time. The NAO camera sees a red ball, however it seems to be a pink ball viewed by the IP camera and a brown ball viewed by the Web camera.

We still use the Fuzzification-based mass construction method described in Section 3.3.1. The threshold ϵ is chosen as 0.2 for example.

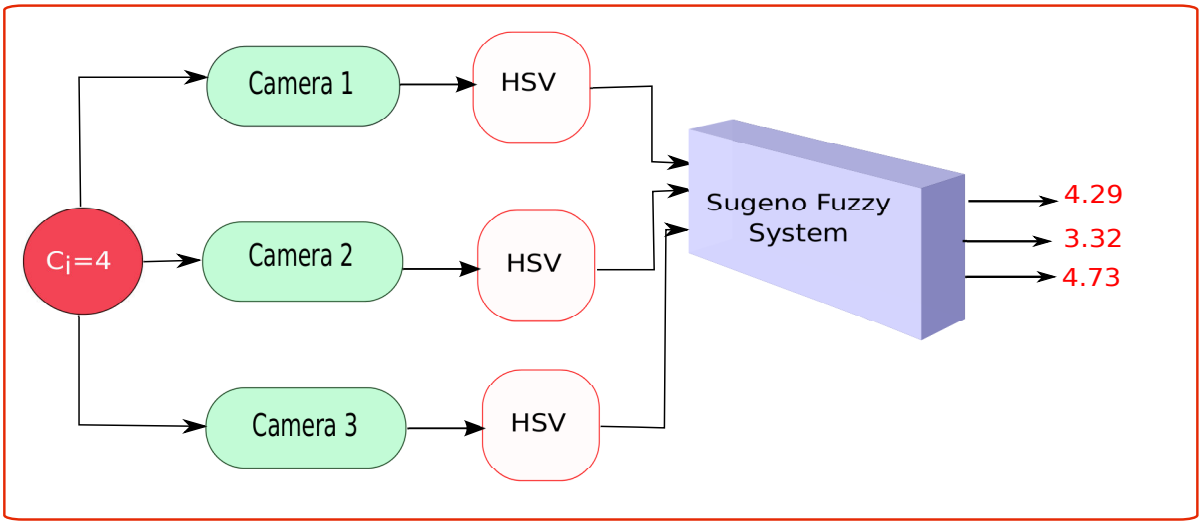

FIGURE 3.14: Conflict among three cameras.

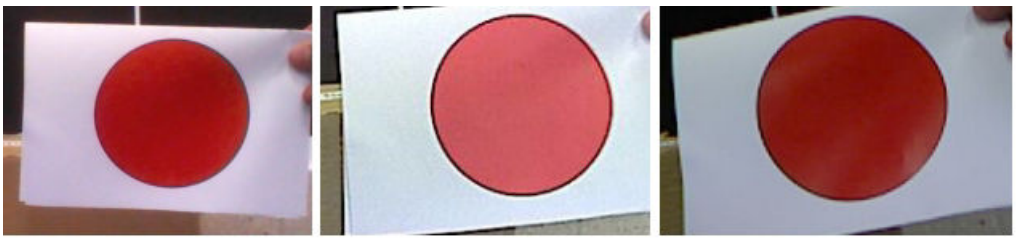

FIGURE 3.15: From left to right: a red ball is captured by the NAO camera, the IP camera, and the Web camera at the same time.

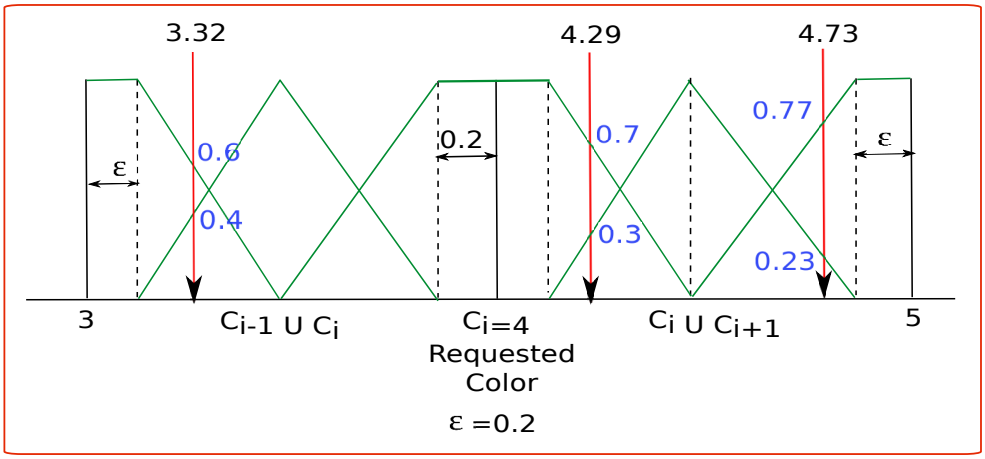

FIGURE 3.16: Construction of mass from Sugeno output.

Mass	$\emptyset$	C_{i-1}	C_i	$C_{i-1} \cup C_i$	C_{i+1}	$C_{i+1} \cup C_{i-1}$	$C_i \cup C_{i+1}$	Ω
m_1	0	0	0.7	0	0	0	0.3	0
m_2	0	0.6	0	0.4	0	0	0	0
m_3	0	0	0	0	0.77	0	0.23	0

TABLE 3.6: The mass values given by the three cameras.

The conflict k between these masses is calculated as 0.907 meaning a very high conflictual case.

In this situation, we suppose that the reliabilities of the three cameras: $\alpha_1 = 0.52$, $\alpha_2 = 0.42$, and $\alpha_3 = 0.48$, respectively. By applying the discounting factor discussed in Section 3.3.1, the mass values then will be in Table 3.7.

Mass	$\emptyset$	C_{i-1}	C_i	$C_{i-1} \cup C_i$	C_{i+1}	$C_{i+1} \cup C_{i-1}$	$C_i \cup C_{i+1}$	Ω
m_1	0	0	0.36	0	0	0	0.16	0.48
m_2	0	0.25	0	0.17	0	0	0	0.58
m_3	0	0	0	0	0.37	0	0.11	0.52

TABLE 3.7: The mass values given by the three cameras after discounting by reliabilities.

After that we show the results of combination by using the rules of Dempster-Shafer, Yager, and Florea, respectively, in Tables 3.8, 3.9, and 3.10.

Mass	$\emptyset$	C_{i-1}	C_i	$C_{i-1} \cup C_i$	C_{i+1}	$C_{i+1} \cup C_{i-1}$	$C_i \cup C_{i+1}$	Ω	Decision
m_1	0	0	0.36	0	0	0	0.16	0.48	
m_2	0	0.25	0	0.17	0	0	0	0.58	
m_3	0	0	0	0	0.37	0	0.11	0.52	
m_{DS}	0	0.1	0.29	0.06	0.20	0	0.13	0.22	
bel	0	0.1	0.29	0.45	0.20	0.30	0.62	1	C_i
pl	0	0.38	0.70	0.80	0.55	0.71	0.90	1	C_i
$betP$		0.2	0.46		0.34				C_i

TABLE 3.8: Dempster-Shafer combination and decision.

Mass	$\emptyset$	C_{i-1}	C_i	$C_{i-1} \cup C_i$	C_{i+1}	$C_{i+1} \cup C_{i-1}$	$C_i \cup C_{i+1}$	Ω	Decision
m_1	0	0	0.36	0	0	0	0.16	0.48	
m_2	0	0.25	0	0.17	0	0	0	0.58	
m_3	0	0	0	0	0.37	0	0.11	0.52	
m_{Yager}	0	0.06	0.2	0.04	0.14	0	0.09	0.47	
bel	0	0.06	0.2	0.30	0.14	0.2	0.42	1	C_i
pl	0	0.58	0.8	0.86	0.7	0.8	0.94	1	C_i
$betP$		0.24	0.42		0.34				C_i

TABLE 3.9: Yager combination and decision.

Mass	$\emptyset$	C_{i-1}	C_i	$C_{i-1} \cup C_i$	C_{i+1}	$C_{i+1} \cup C_{i-1}$	$C_i \cup C_{i+1}$	Ω	Decision
m_1	0	0	0.36	0	0	0	0.16	0.48	
m_2	0	0.25	0	0.17	0	0	0	0.58	
m_3	0	0	0	0	0.37	0	0.11	0.52	
m_{Florea}	0	0.06	0.17	0.04	0.12	0	0.08	0.55	
bel	0	0.06	0.17	0.26	0.12	0.17	0.36	1	C_i
pl	0	0.64	0.83	0.88	0.74	0.83	0.94	1	C_i
$betP$		0.26	0.41		0.34				C_i

TABLE 3.10: Florea combination and decision.

The three combinations give the same results: the colored ball C_i (red in this case). However, the Dempster-Shafer combination still gives a more reasonable result since the difference between C_i and the other decisions is bigger than this value in the combination rules of Yager and Florea. This is an interesting case when we have three sources having a very high conflictual degree ($k = 0.907$) but the correct decision is still made.

3.3.3 On the Choice for the Number of Sources

How many sources are needed for the fusion is still a controversy discussion. Theoretically, the more sources are used, the more information we have and the fusion is expected to be better. However, it should depends on the specificity of the applications and the scenarios where they are used, at least in this work of the color detection for the NAO robot by using multi-camera.

We have tested with the fusion of two and three cameras, and they give good results (see Section 3.4). However, if we add more sources, the conflictual cases among sources would be more complicated and the fusion strategy is not ensured to have better results. Moreover, due to the characteristics of the NAO robot, setting up too many sources for it may require complex procedures.

In addition, adding more sources means that we have to manage more resources, which can affect to the performance of the detection system. Indeed, the NAO robot communicates with other cameras through a local wireless network, so the response time is strongly affected by the number of sources, which should be avoided in a real-time system.

Under those circumstances, in this work we consider using only two or three information sources for the fusion in the color detection system of the NAO robot.

3.4 Application and Validation

3.4.1 The Context of Application

We have implemented the discussed methodology in an application in which the NAO robot is requested to find a color ball. In this application, the robot's microphones and a camera on its head are activated to be used for human voice recognition and images

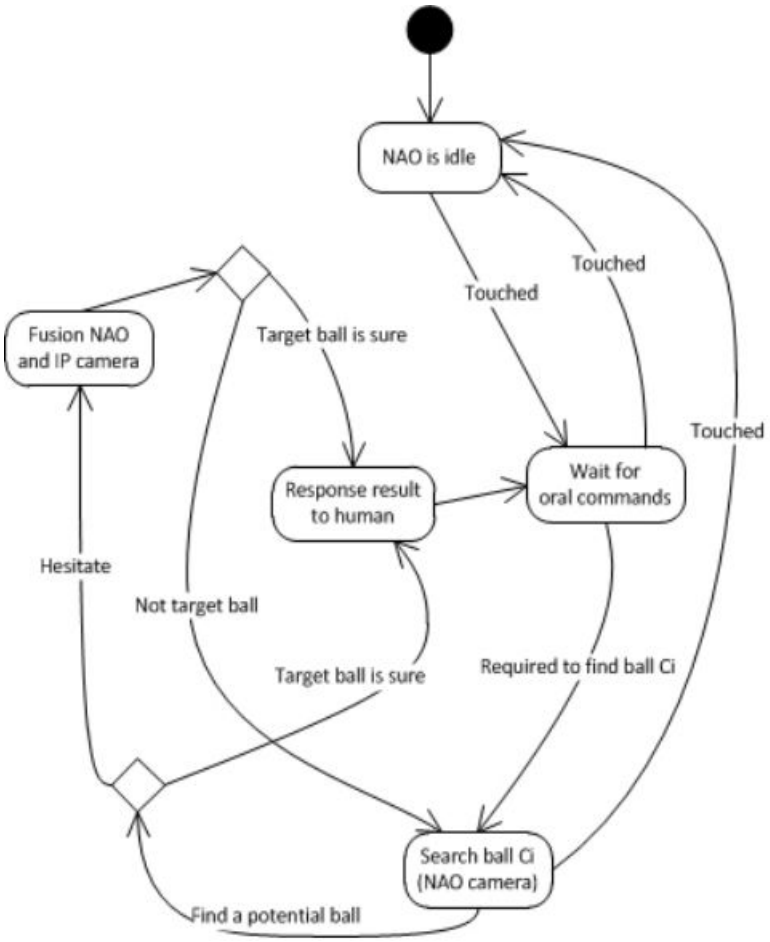

FIGURE 3.17: Processing flow of the color detecting application.

processing, respectively. The tactile sensor on its head and a bumper on its feet are used for humans to start/stop the application. Additionally, 25 degrees of freedom (DOF) help NAO perform many motion tasks, and we use its two legs to move when searching the ball, its hands are used to point to the target ball it has found. Furthermore, we add another IP camera to support for the robot, and the two cameras communicate through a wireless local network. The IP camera is located close to the robot such that they see the same scenes.

In this work, we chose C++ for the development of application in order to gain better performance because it is the native language for NAO robot. For visual processing, OpenCV 2.4.9 was employed to process images captured from the cameras. The speech recognition and the movement of the robot are done using libraries provided by the manufacturer. For more information about the implementation of the scenario, the readers are invited to Appendix B.

Fig. 3.17 describes the scenario of the application. First, the robot comes to the "idle" state when it starts, because we may have several applications installed on the robot, so there is a need to have a way of starting for each application. In our case, we touch the tactile sensor on the robot's head or the bumper on its right foot to start/stop the

application of color detection. When it is touched, it will wait for the oral commands from a human. The commands are in the following form:

"NAO, find the C_i ball"

In this case, C_i is the requested color ball that was discussed in previous sections. When NAO recognizes an oral command from a human, it starts searching for the target by walking around and uses its camera to scan the surrounding environment. Hough transformation is used to detect the balls' shapes from captured images and NAO will decide on the detected ones. When it is sure that a ball has the target color, it will stop moving and says that it has found the ball.

In the case where NAO is uncertain about the color of a detected ball, for example it is not sure that the detected ball is yellow or green, it will call the connected IP camera to make the fusion between these two cameras. After the fusion, if NAO decides that this is the target ball, it will respond to the human, otherwise, it continues its search. During the finding process, we can stop NAO by touching the tactile sensor or its bumper.

3.4.2 Validation of the Detection and Discussion

We did several experiments to validate the results of color detection in both HSV and Lab color spaces. We tested with 9 colors, and with each color, 40 tests were carried under different lighting conditions and we varied of the colors' hues in order to challenge uncertainties and imprecisions. For each test, the cameras capture the same ball, then each one gives their detection result using the Fuzzy Sugeno system, then we use the Dempster-Shafer combinator to fuse cameras' decisions to have better outputs.

Fusion of Two Cameras

In this experiment, two cameras are taken into account: the NAO camera and another IP camera. Table 3.11 shows the experimental results in the HSV color space. According to Table 2.19, a small value of ϵ is preferred in order to balance between the reliability and the uncertainty of the system. The threshold value ϵ used in this validation is 0.15. According to the results it can be stated that the color detection performance has been increased using the Dempster-Shafer theory. Indeed, the rate of detection has been improved by 24% for NAO and 33% for the IP camera.

Color	Sugeno Fuzzy (NAO)	Sugeno Fuzzy (IP)	Fusion
Blue	67.50%	50.00%	90.00%
Purple	42.50 %	47.50%	67.50%
Pink	47.50%	32.50%	85.00%
Red	72.50%	10.00%	92.50%
Brown	42.50%	20.00%	70.00%
Orange	42.50%	40.00%	57.50%
Yellow	30.00%	55.00 %	70.00%
Green	70.00 %	80.00%	80.00%
Cyan	12.50 %	12.50%	35.00%
Average	**47.50%**	**38.61%**	**71.94%**

TABLE 3.11: Color detection performance in HSV with two cameras.

Table 3.12 shows the results in the Lab color space. The detection rate in this space is not as good as in the HSV color space. The reason, as explained in Chapter 2, is due to the fact that the HSV color space gives us more eases to construct the fuzzy rule base, although we cannot conclude that which one is better in general. The detection threshold ϵ is still 0.15. In this experiment, the fusion of two cameras still improves dramatically the detection rate comparing to each individual camera.

Color	Sugeno Fuzzy (NAO)	Sugeno Fuzzy (IP)	Fusion
Blue	15.00%	20.00%	55.00%
Purple	20.00 %	52.50%	70.00%
Pink	27.50%	20.00%	42.50%
Red	65.00%	20.00%	65.00%
Brown	17.50%	12.50%	35.00%
Orange	5.00%	5.00%	22.50%
Yellow	40.00%	57.50 %	72.50%
Green	47.50 %	32.50%	65.00%
Cyan	70.00 %	92.50%	92.50%
Average	**34.17%**	**34.72%**	**57.78%**

TABLE 3.12: Color detection performance in Lab with two cameras.

Fusion of Three Cameras

In this experiment, we add a Web camera as the third information source, to see the effect of the fusion. The threshold ϵ is still chosen as 0.15. We also test with both HSV and Lab color spaces, as shown in Table 3.13 and 3.14.

It is clear to see that when combining three cameras, the results are better than two cameras. Indeed, in the test with the HSV color space, the fusion of two cameras gives us a detection rate of 71.94% whereas this number for three cameras is 75.56%. On the

other hand, the test with three cameras in the Lab color space also give better results than two cameras (57.78% and 67.22%).

It is also worth noting that the detection rate for each camera cannot be high because the chosen ϵ is too small, and the colors tested are varied so much in their hues for challenging uncertainties and imprecisions. However, it is also the interesting point of this work because we improve it by using fusion strategy.

Color	**Sugeno Fuzzy (NAO)**	**Sugeno Fuzzy (IP)**	**Sugeno Fuzzy (Webcam)**	**Fusion**
Blue	67.50%	50.00%	70.00%	92.50%
Purple	42.50 %	47.50%	35.00%	67.50%
Pink	47.50%	32.50%	32.50%	82.50%
Red	72.50%	10.00%	10.00%	87.50%
Brown	42.50%	20.00%	60.00%	80.00%
Orange	42.50%	40.00%	25.00%	67.50%
Yellow	30.00%	55.00 %	65.00%	77.50%
Green	70.00 %	80.00%	80.00%	87.50%
Cyan	12.50 %	12.50%	10.00%	37.50%
Average	**47.50%**	**38.61%**	**43.06%**	**75.56%**

TABLE 3.13: Color detection performance in HSV with three cameras.

Color	**Sugeno Fuzzy (NAO)**	**Sugeno Fuzzy (IP)**	**Sugeno Fuzzy (Webcam)**	**Fusion**
Blue	15.00%	20.00%	30.00%	57.50%
Purple	20.00 %	52.50%	17.50%	62.50%
Pink	27.50%	20.00%	20.00%	45.00%
Red	65.00%	20.00%	40.00%	92.50%
Brown	17.50%	12.50%	45.00%	57.50%
Orange	5.00%	5.00%	27.50%	40.00%
Yellow	40.00%	57.50 %	77.50%	82.50%
Green	47.50 %	32.50%	45.00%	70.00%
Cyan	70.00 %	92.50%	85.00%	97.50%
Average	**34.17%**	**34.72%**	**43.06%**	**67.22%**

TABLE 3.14: Color detections performance in Lab with three cameras.

3.5 Conclusion

In this chapter, we remind the problem of uncertainties and imprecisions that are discussed at the end of Chapter 2. In the proposed context, the NAO robot is requested to find an object whose color is linguistically labelled. Actually, by using the Fuzzy Sugeno system, the robot can well detect the target object in ideal conditions, however, due to impacts

from the exploited environment such as lighting conditions, or quality sensors, or the overlap among close colors, the performance of the detection system may be limited.

For those circumstances, this chapter focuses on how we can improve the color detection of the NAO robot, and the application of multi-camera is proposed. In fact, more cameras are put aside the NAO robot and these cameras capture the same scenes. First, each camera uses the Fuzzy Sugeno system to recognize the color of the detected object (a ball), then the Dempster-Shafer theory is employed to fuse their decisions.

Based on the introduction of the threshold value ϵ in Chapter 2, we propose a Fuzzification-based method for constructing mass values. Additionally, by considering the reliabilities of cameras, we avoid the cases of total conflict when combining sources by applying the discounting method proposed by Shafer.

In this chapter, several operators for combining mass vectors are reviewed with illustrative examples. The combinators were proposed by Dempster-Shafer, Yager, Florea, and Dubois-Prade. Each one has its strengths and weaknesses. For this work, we consider a closed world and the reliabilities of cameras, so the Dempster-Shafer combinator is the most appropriate. For the decision method, because the maximum of plausibility is too optimistic, and the maximum of belief is too pessimistic, so we choose the maximum of pignistic probability which is considered to be a compromise between the other two.

Finally, we implemented the proposed method in an application in which the NAO robot is requested to find a color ball. With the support from a second camera (IP), the NAO robot reduces the cases where it hesitates on the decisions. In addition, we validated the proposed methodology with the fusion of three cameras, and we did 40 tests for each color. The experimental results show a dramatical improvement in the detection rate by using the multi-camera fusion in comparison with each individual camera.

At the end of this chapter, we clearly see the advantage of homogeneous sensor fusion. It is applied in a specific scenario in which the NAO robot is requested to find a colored object. In the next chapter, we still focus on the fusion of multi-camera data, however the employed sensors are heterogeneous, and the objective is to recognize more complicated objects for the NAO robot.

Chapter 4

Fusion of Heterogeneous Sensors Data

In the previous chapter, we already discussed about the fusion of homogeneous sensors (2D cameras) for the color detection. The NAO robot is requested to find a ball with a specific color. Only one NAO's camera is not sufficient to accomplish the task due to many impacts from environment such as lighting conditions, noises, or the quality of sensor, so extra cameras are added to improve the performance of the detecting process. Each camera employs the Fuzzy Sugeno system to produce its own output, then the Dempster-Shafer theory is used to fuse those results.

In this chapter, we continue the discussion about the advantage of fusion. However, this focuses on how we integrate heterogeneous sensors to improve the efficiencies of the NAO robot. In the considered scenario, the robot is requested to recognize an object in front of it. These objects are previously learned during the training phase, then in real-time, the robot has to give the right name of the object it detects frontwards.

The work described in this chapter is considered as a generalized idea of Chapter 3 where we use a vision system to recognize colors. The main difference is that the objects tested in Chapter 3 are simple balls with a uniform color for each ball, but in this scenario, the objects can be in arbitrary forms with more complex structures. Our objective in the future works is trying to add more eases in the object recognition by considering the combination of the color detection described in the previous chapter and the object recognition of this chapter.

For the test of heterogeneous sensors in this work, we continue using a camera of NAO and an IP camera, then we add the third source which is an Axus Xtion Pro camera (3D) (see Fig. 4.1). The first two were already described in the previous chapter. For the third camera, it provides depth images which describe the captured scenes in the 3D space by clouds of points. Table 4.1 shows the specification of the 3D camera. Indeed, Xtion Pro uses infrared sensor and it comes with a set of developer tools as well as an easy plug and play USB design which allow ease of use. Additionally, when compared to another 3D camera like Kinect, Axus Xtion is more compact, lighter weight and does not require power supply except USB. It also provides better quality of RGB images than Kinect.

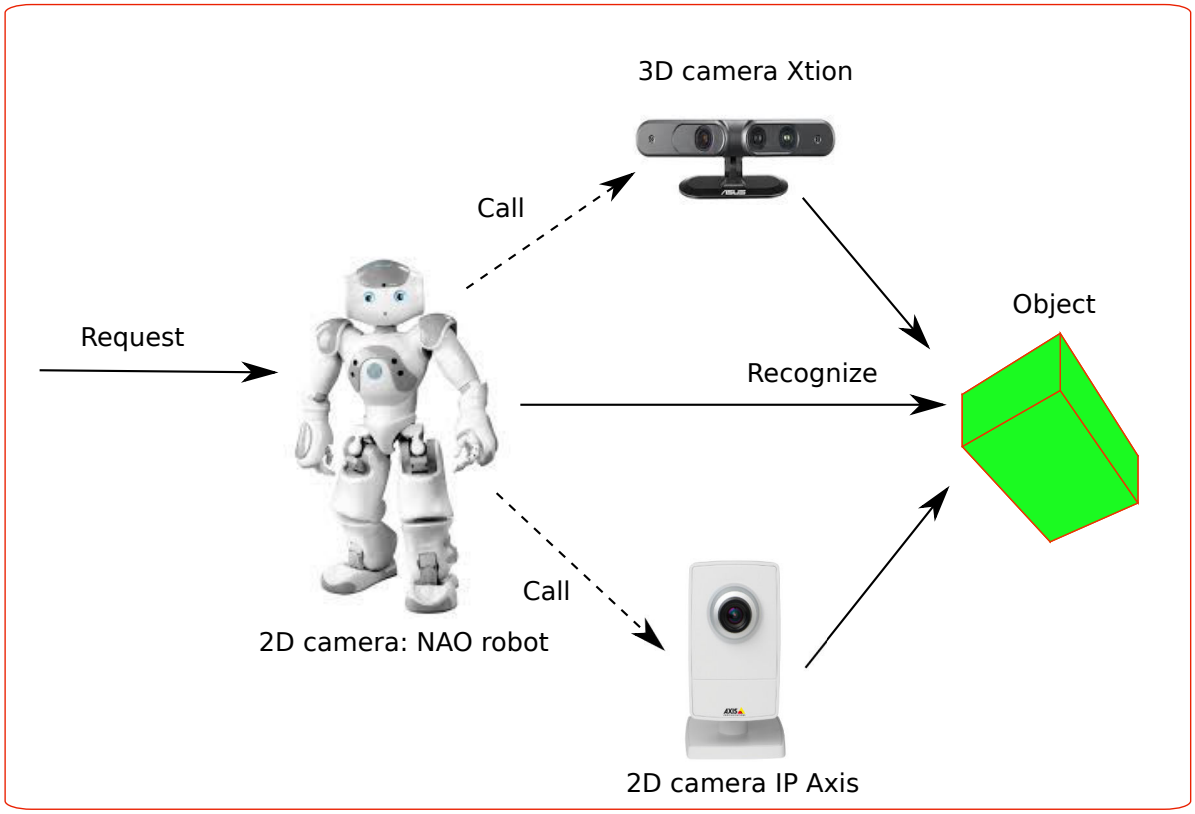

FIGURE 4.1: Object recognition context by using multiple heterogeneous cameras.

Field of view	58° H, 45° V, 70° D
Distance of use	Between 0.8m and 3.5m
Interface	USB 2.0
Software	1 Xtion Portal 3 Motion Games: BeatBooster, MayaFit, DanceWall
Dimensions	18 × 3.5 × 5 (L × W × H)

TABLE 4.1: Specification of the Axus Xtion camera.

In the scenario, we prepare a set of complex objects that have many similarities (in order to challenge uncertainties and imprecisions). After that, we train the NAO robot and the cameras with these objects. During runtime, the NAO robot is asked to recognize the object which is in front of it, and it uses the three cameras for improving the recognition quality.

In fact, the NAO's camera and the IP camera give 2D images which allow to describe the colors, textures of the detected objects. Beside that, the Axus camera provides 3D images which describe the objects as clouds of 3D points. Each type of camera gives advantages and disadvantages, and the idea of this work is to combine these heterogeneous sensors. The fusion is expected to give better recognition results comparing to the recognition rate of each individual camera. Taking advantage of the Dempster-Shafer theory

FIGURE 4.2: Axus Xtion Pro

as discussed in the previous chapter, we continue using this approach to do the fusion of cameras.

The organization of the chapter is as follow. First, Section 4.1 introduces the problem of the object recognition and our choice for the solution. Second, Section 4.2 describes the recognition system in a global view, and the application of the Dempster-Shafer theory in this scenario. Section 4.3 then gives an example applying the proposed method, and Section 4.4 shows the experimental results. Finally, Section 4.5 concludes the chapter.

4.1 The Context of Object Recognition

This chapter focuses on how the NAO robot recognizes an object that is previously learned, so we firstly have a look at how the other works have dealt with object recognition. In fact the problem has been addressed for several decades. The number of methodologies is huge up to now; each of them tried to prove their strengths and overcame the weaknesses of the preceding solutions. To be more coherent to this thesis, we present here some existing works that employ single camera and multiple cameras for the object recognition.

4.1.1 Object Recognition by Single Camera

The advantage of using only one camera is that it does not require to set up extra ones, and the performance in terms of processing time sounds better. Therefore, to recognize objects by visual sensors, ones traditionally employ a single camera for the recognition. , both 2D and 3D data.

For instance, [5] proposed a technique for recognizing objects by using shape matching between images. The feature description of the samples and the query image are calculated by Geometric Blur descriptor, then for each feature point in a sample image, they find the best matching feature point in the query image based on normalized correlations of the descriptor. The mean of these best correlations is the similarity of the exemplar to the query. They also proposed an algorithm to calculate the correspondence from each sample image to the query image, finally choose the class of the sample that has the best correspondence to the query image, which implies a lower cost.

Besides that, [44] introduced the term 'inner-distance' which is considered as the length of the shortest path between landmark points within the shape silhouette. The distance is used to build better shape representation. First, they build articulation invariant signatures for 2D shapes by combining the inner-distance and multidimensional scaling. After that, the shape context is extended with the inner-distance to form a new descriptor. They also define a new dynamic programming-based method for shape matching and comparison.

For some texture-based approaches, [57] proposed a texture descriptor based on Random Sets and the experiment shows that this method outperforms the co-occurrence matrix descriptor. Decision tree induction is used in that work to learn the classifier. Another example can be found in [3] where color and texture information are both used in an agricultural scenario to recognize fruits.

On the other hand, some context-based methods like [26], [52], and [74] consider contextual information surrounding the target objects. These information come from the interaction among objects in the scene and they help to disambiguate appearance inputs in recognition tasks. For instance, Galleguillos et al. introduce a method for object categorization that incorporates two types of context: co-occurrence and relative position with local appearance-based features. The approach uses a conditional random field to maximize object label agreement according to both semantic and spatial relevance. They model relative location between objects using simple pairwise features, then by vector quantizing this feature space, they learn a small set of prototypical spatial relationships directly from the data.

Comparatively, the methods based on local feature description like SIFT ([46]) and SURF ([30]) have received many positive evaluations and have been widely applied ([1], [67], [60], [49]). SIFT extracts keypoints from object to build feature vectors. They then calculate the matching (using Euclidean distance) between an input object and the ones in database to find the best candidate class. After that, the agreement on the object and its location, scale, and orientation are determined by using a hash table implementation of the Generalized Hough Transform. In a different manner, SURF uses a blob detector based on the Hessian matrix to find interest points, then it calculates the descriptor by using the sum of Haar wavelet responses. Finally, by comparing the descriptors obtained from different images, the matching pairs can be found.

The above discussed works concentrate on 2D objects recognition. In contrast, for the purpose of collecting spatial information about the detected objects, and avoiding imprecision of 2D images under non-ideal lighting conditions like outdoor environment,

FIGURE 4.3: Example of occlusion during object recognition.

some works focus on 3D object recognition. For instances, [36] proposed an extended version of the Generalized Hough Transform in 3D scenes. The new version follows the same principle with the normal Hough-based method but the main difference is that the gradient vector is replaced with a surface normal vector. Each point in the input cloud votes for a spatial object's reference point and the accumulating bin with the maximum votes indicates an instance of the object in the scene. In [23] and [38], the 3D extensions of SIFT and SURF descriptor also gave positive recognition results.

In addition, [77] introduced a new 3D shape descriptor called Intrinsic Shape Signature to characterize a local/semi-local region of a point cloud. This descriptor uses a view-independent representation of the 3D shape to match shape patches from different views directly, and a view-dependent transform encoding the viewing geometry to facilitate fast pose estimation. In a different manner, [16] and [56] consider the use of point pairs for the description and the feature matching is then done by implementing a hash table.

Recently, the SHOT descriptor [71] has emerged as an efficient tool for 3D object recognition ([72], [59]). Indeed, the descriptor encodes histograms of basic first-order differential entities (i.e. the normals of the points within the support), which are more representative than plain 3D coordinates about the local structure of the surface. After defining an unique and robust 3D local reference frame, it is possible to enhance the discriminative power of the descriptor by concerning the location of the points within the support, from that describing a signature.

It is clear that the above mentioned approaches have experimentally shown good results in the object recognition. Nevertheless, many of them did not focus on the problem of uncertainties and imprecisions which might come from the quality of data and sensors, the lighting conditions, and especially, the viewing angles of cameras to the objects, and the similarity among confusing objects. Fig. 4.3 shows an example of having difficulty during the object recognition when the target object (the bottle) is hidden partially by other objects in the scene.

Under those circumstances, some works consider adding more cameras to the recognition system since only a single camera cannot give enough accurate results. The key point is trying to solve the difficulty when we have only

a single view to the object, and to prevent the influence of quality-limited cameras to the performance.

4.1.2 Objects Recognition by Multiple Cameras

Actually, until the time of this thesis, there are not many works applying multi-camera for the object recognition in the literature. The main difference of such systems to traditional systems is that in stead of using one camera, two ore more cameras are used for the recognition process.

In [12], a method for using multiple cameras to simultaneously view an object from multiple angles is proposed. First, each camera detects the object by using its own base classifier which is learned from an extremely large training set. After that, the outputs of these single-image object detectors are combined to obtain improved detection performance. The correspondence between pairs of detection are determined, i.e. which detections in both cameras actually correspond to the same object in the scene. Then, for each pair of corresponding detections, they compute the posterior probability of the class label. This method sounds good, however, it requires so much time for the training process, and the analysis on the used sensors is quite simple: only two 2D cameras are tested. Additionally, posterior probability-based method remains some drawbacks such as it does not tell how to select a prior and often brings high computational cost.

In [73], an object recognition using multi-view imaging is introduced to solve the problems of noise or low resolutions generated when using single camera. The detected target is captured from multiple views then SIFT features are extracted. After that, they combine all the nearest and second nearest neighbour matches from all multi-view images and filter them. Finally, a two-stage Hough histogram clustering is applied to filter the matches. The entry in the dictionary with the highest Normalized Cross-Correlation is chosen as the recognition result. Indeed, this method would pose a question about the performance since it requires many calculation steps. Moreover, the test is carried with multiple-view but not really multiple-camera, so the problem of uncertainties related to the quality of sensor may not be resolved.

In addition [68] introduces an approach combining multiple color cameras and multiple depth sensors (Kinect) to recognize objects. For color images, the DPM (deformable part model), and the VFH descriptor (viewpoint features histogram) is applied for point clouds generated from depth cameras. For the fusion of these cameras, they use the center point of each object and transform all hypotheses from different cameras into a single 3D coordinate system. Each hypothesis votes for all its neighbours, which are closer than a threshold t with 50% of its own score. Finally, they perform 3D non-maximum suppression with the distance threshold t in 3D space to yield the final set of hypotheses. Actually, the idea of this method is novel and interesting since it takes into account the advantages of 2D and 3D data. However, the fusing step is too simple to have a good decision because the voting strategy is not a real good candidate for doing the fusion.

There are also other works related to using multi-camera or multi-view for the recognition which we can find in the literature. For example [31] with a 3D human action

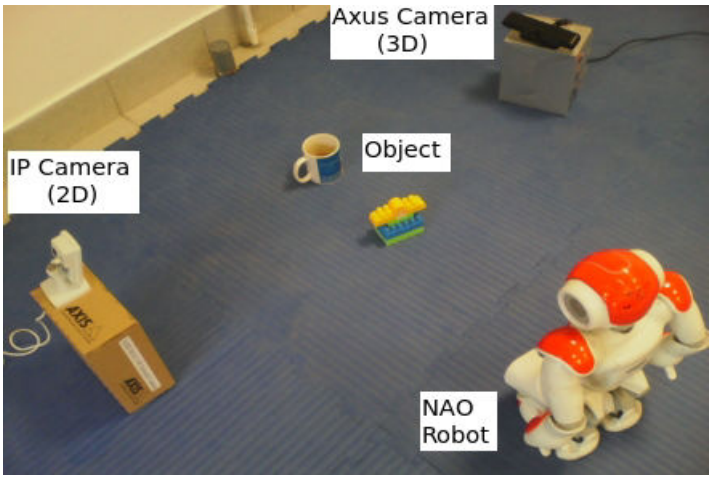

FIGURE 4.4: The multi-camera system helps NAO recognize objects.

recognition for multi-view camera system, or [22] with multiple cameras to simultaneously acquire images from different view angles of an unknown, randomly occluded object belonging to a set of a priori known objects. Other examples can also be found in [61] and [58].

4.1.3 The Choice for Our Solution

According to the lack of reliability when using only one single camera, and to good performances shown by existing works using multiple cameras, we will improve the object recognition for the NAO robot by adding external cameras to its vision system. In this work, we use heterogeneous camera sensors with two 2D cameras: the NAO's and the IP Axis which are described in Chapter 3, and we add one 3D camera: Axus Xtion Pro, as shown in Fig. 4.4.

In fact, the combination of the both types of cameras (2D and 3D) brings some benefits. 2D images provide information about the characteristics on the surface of objects such as colors, contrast, intensity... However, when we want to differentiate two objects like in Fig. 4.5, it is difficult for the 2D cameras because the objects' surfaces do not have many interesting features except a uniform color (yellow). This is the place where the 3D camera becomes helpful since it can provide depth information about the object's shapes.

On the contrary, for the objects that look like in Fig. 4.6, it is difficult if we have only depth information because the two objects look the same in their shapes. However, it still remains interesting for the 2D cameras because they can well differentiate the two objects based on the features of their surfaces.

The choice of methods for processing image data is also important. First, for 2D processing, we use the SURF descriptor (discussed in Section 4.1.1). This approach allows extracting feature points with being invariant in rotations. There are some works studying the comparison between SURF and SIFT, and it is not false to say that SURF shows a better performance than SIFT (see [27]). For 3D processing, we use the SHOT

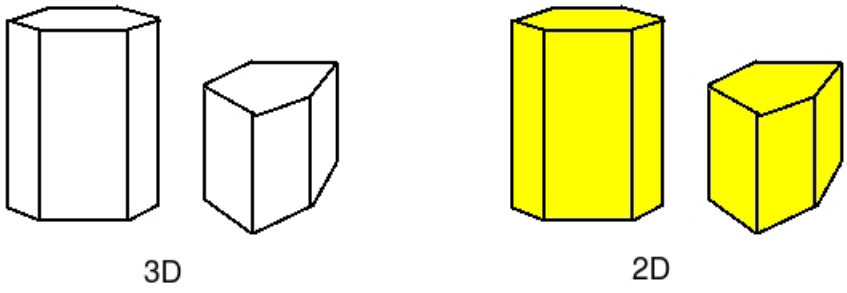

FIGURE 4.5: The difficulty of 2D cameras.

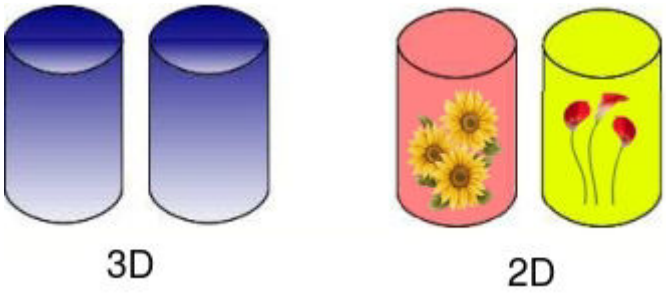

FIGURE 4.6: The difficulty of 3D cameras.

descriptor (see Section 4.1.1), which achieves a better balance between descriptiveness and robustness. A study about the performance of this descriptor can also be found in [4] which shows that SHOT'results are very good.

Finally, in order to fuse the information coming from different sources (the cameras in this case), we employ the Dempster-Shafer theory due to its strong advantages that are already discussed in Chapter 3. Each camera gives its own information about the matching between the detected object and the trained models, we then construct mass functions based on these informations, and the Dempster-Shafer theory is used to combine the masses and give the final decision. The detail of the proposed approach will be presented in the next section.

4.2 Methodology of the Object Recognition System By Multi-camera

4.2.1 System Overview

In the proposed scenario, the NAO robot is requested to recognize the object appearing in front of it. The human voices are processed actually in the same way as in the scenario of the color detection described in Chapter 2. When the robot receives humans' commands to recognize the object, it uses multiple cameras for the recognition.

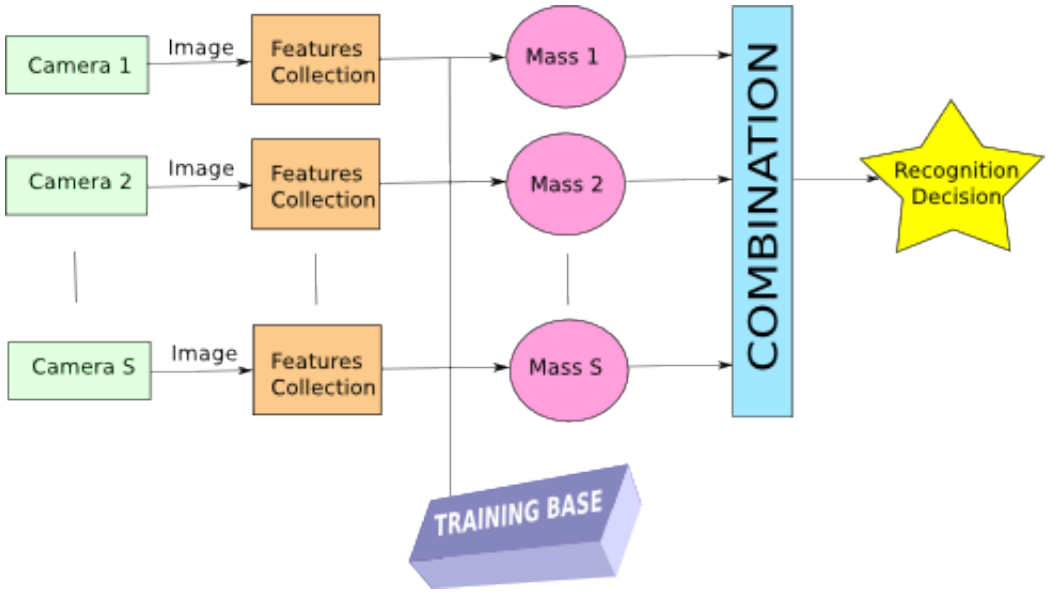

FIGURE 4.7: Overview of the multi-camera-based recognition system.

Fig. 4.7 depicts the overview of the recognition system. First, the cameras capture the scene containing the object. The system is generic so that we can use any number of cameras, as long as they all look at the target scene. No matter which type of camera used, each one extracts the information of the object and represents them in a form of features collection. From the features extracted by a camera, we build a mass function, then the Dempster-Shafer theory is employed to combine those masses in order to provide a decision at the final stage. To construct the mass functions, a training base is taken into account. Indeed, this training base contains the images that are used to train at the preprocessing step, which will be explained later.

Fig. 4.8 describes the detail process of each camera in the object recognition system. This process is applied for both 2D and 3D cameras. From the scene image captured, we extract interesting points of the object. These points are then modelled by a description technique, which allows us to construct mass values. Once we construct a mass vector for each image captured from a camera, we choose the decision that has the maximum of pignistic probability. In the case of multiple cameras, before the decision step, we combine the mass vectors by using combination operators of the Evidence theory.

4.2.2 Data Extraction and Preprocessing

In the first place, the cameras have to extract information about the scenes and the objects to be recognized. This extraction is very important because it has direct effects to the recognition results. For example we want to recognize an object that is put in a frontward scene, supposing that we already approximate its position. If the extraction technique is not good enough, we may obtain much noises from the surrounding scene, or the description of the object does not give a good model as the object represents itself to human perception.

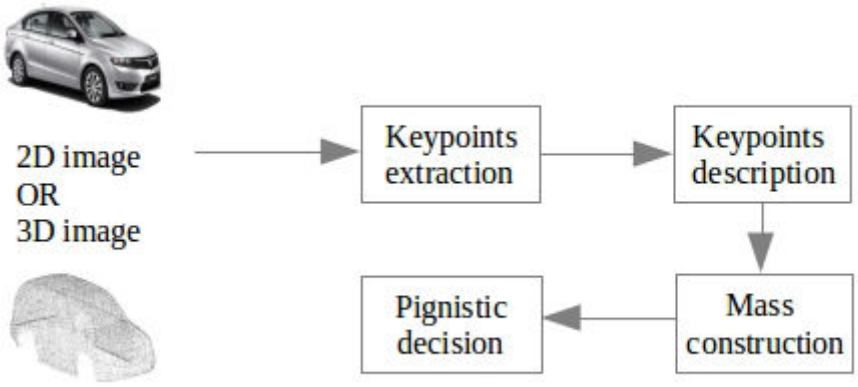

FIGURE 4.8: Process of each camera in object recognition.

There are several ways to represent objects in scenes such as color histograms, interesting points extraction, contextual modelling, or other methods. As discussed above, in this work we decide to extract interesting points of the objects due to its advantages in the context of object recognition challenging uncertainties and imprecisions.

First, interesting points (or key points) of the object in the scene are extracted. In an image, an interesting point can be described as a point that has rich information about the local image structure around it, and these points characterize well the patterns in the image. After that, we use methods of descriptor to build a feature vector for each interesting point. The methods of descriptors used in this work are SURF ([30]) for 2D data and SHOT ([71] for 3D data. The interesting points are taken into account by a descriptor are so called feature points.

In fact, to detect interesting points, SURF uses an integer approximation of the determinant of Hessian blob detector, which can be easily accomplished by using a precomputed integral image. After that, the description of features is done based on the sum of the Haar wavelet response around the interesting point. The approximation using box filter is illustrated in Fig. 4.9. Fig. 4.10 shows example of using SURF to detect and describe feature points in an image.

In contrast with 2D data, the 3D data representing an object is a cloud of points in 3D space. For 3D data processing, SHOT encodes information about the topology within a special support structure. The sphere is divided into number of bins from which a one-dimensional local histogram is computed for each one. After all local histograms have been computed, they are combined together to form the final descriptor. SHOT uses local reference frame that helps it be invariant in rotation. Fig. 4.11 shows an example of points cloud from a captured object.

For more information about SURF and SHOT descriptors, the reader is invited to Appendix C where we represent the principle and the use of these descriptors in the scenarios of the NAO robot described in this work.

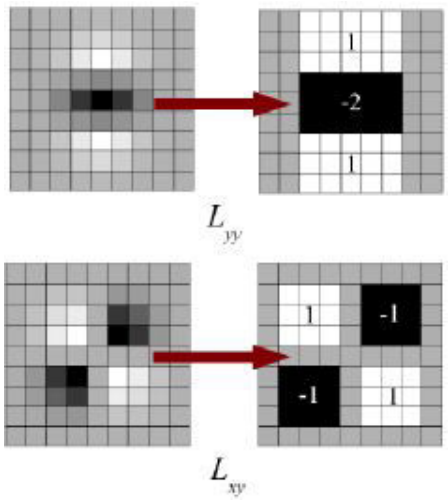

FIGURE 4.9: LoG approximation with Box Filter in SURF.

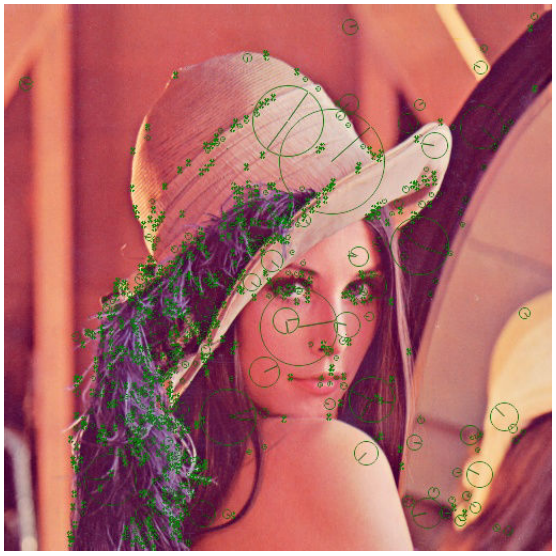

FIGURE 4.10: Example of SURF features descriptor.

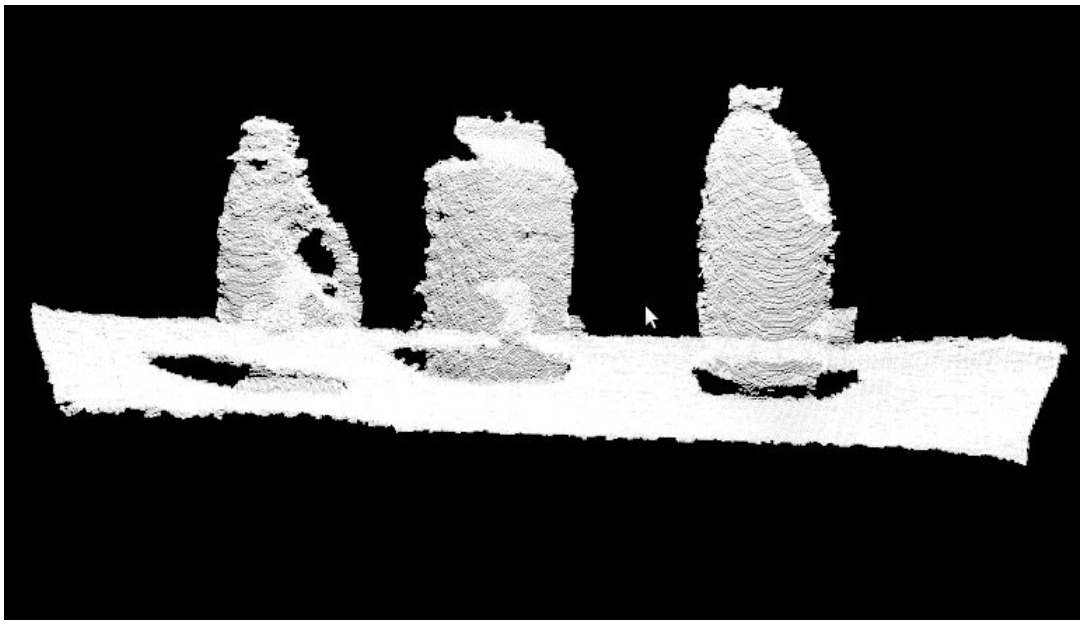

FIGURE 4.11: Example of points cloud.

In fact, the extracted feature points are the main data to be processed for the recognition. Fig. 4.12 shows the principle idea. We first have extracted the feature points of trained objects, they are considered as models. When an object is detected and its feature points are extracted, we will decide the class of the object based on the matching between it and the models in the training base. Actually, the number of feature points are varied depending on the objects and scenes, also the moment of the capture. In average, for both types of cameras, the number of feature points are normally between 100 and 200. In this work, the Evidence theory is taken into account to do this job and its principle application is described in more detail in the next section.

4.2.3 The Dempster-Shafer Theory in the Scenario

In the previous section, we show that the key of this work is how we can model the matching between the captured object and the trained ones based on their extracted feature points. This section presents the steps of applying the Dempster-Shafer for solving the problem. First of all, we will look at the idea of how this theory can do that.

In fact, the NAO robot recognizes the name of an object that was already trained in the preprocessing step. That is, the objects to be recognized are always in a predefined set. Suppose that we have N classes of objects, so the space of discernment is defined as:

$$\Omega = \{O_1, O_2, ..., O_N\} \tag{4.1}$$

Then we have the power set which contains all the possible hypotheses H:

$$2^{\Omega} = \{\emptyset,\ \{O_1\},\ \{O_2\},\ ...,\ \{O_N\},\ \{O_1 \cup O_2\},\ ...,\ \{O_1 \cup O_N\},\ ...,\ \Omega\} \tag{4.2}$$

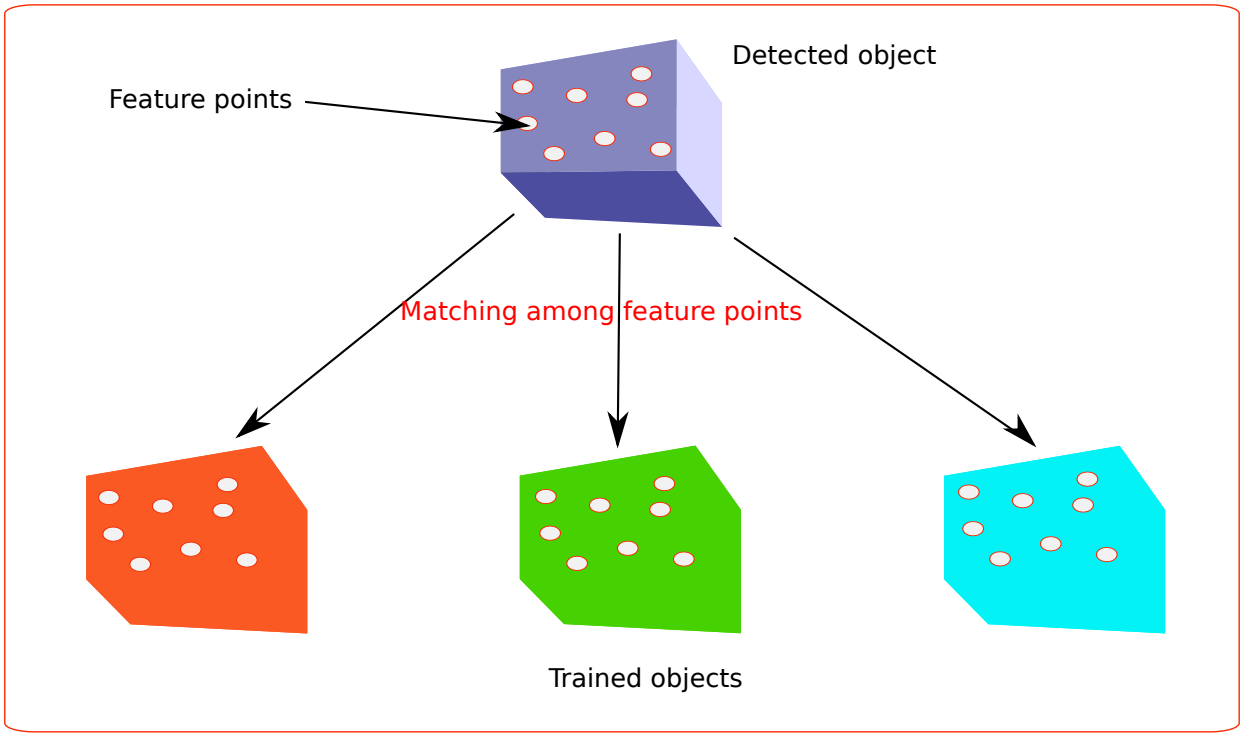

FIGURE 4.12: The detected feature points need to be compared with trained feature points.

As already explained in the previous chapter, we have to construct a mass function which describes the degree of belief for all possible hypotheses in the power set. This function satisfies:

$$\begin{aligned} m &: 2^{\Omega} \rightarrow [0,1] \\ \sum_{H \in 2^{\Omega}} & m(H) = 1 \end{aligned} \tag{4.3}$$

As the matter of fact, constructing mass values is the most difficult task when applying the Dempster-Shafer theory. The advantage of the mass function is that it allows expressing the uncertainties under the form of sub-sets. This feature really contributes to the proposed scenario since we test with the objects that have many similarities. As discussed above, the cameras extract feature points of the object to make decision, and each of feature point can be similar among confusing objects, which makes the problem become complicated if applying other normal methods.

We have already seen the advantages of the Dempster-Shafer theory especially in dealing with uncertainties and imprecisions, whose mass function models different decisions and their combinations. A hypothesis $H = O_{i_1}, O_{i_2}, ..., O_{i_x}$ can be considered as a hesitation among the single decisions. If we take this idea into account, imagine that a feature point of the detected object is matched with not only a single class but also multiple classes, and a set of feature points can quantitatively construct mass functions.

To illustrate the proposed idea, we consider a simple case in Fig. 4.13 where we suppose that there are only three classes of objects: O_1, O_2, and O_3. For the sake of explanation, we assume that we have only one training image for each class. With an input image (captured by a camera) which contains a set X of feature points of object, our mission is to decide the appropriate class for X. The basic idea is that each feature point $p_i \in X$ will vote for a hypothesis $H \in 2^{\Omega}$ based on its matching to the training images. In that figure, the feature point p_1 matches both images of class O_1 and O_2, so we accumulate one vote for the hypothesis $H = \{O_1 \cup O_2\}$. Similarly, the feature point p_2 votes for $H = \{O_3\}$. By doing the same principle for all the feature points of X, we can construct all elements of the mass function after doing a normalization step. Due to the need of clear explanation in a scientific work, the step of defining the matching and constructing mass function will be mathematically described thereafter.

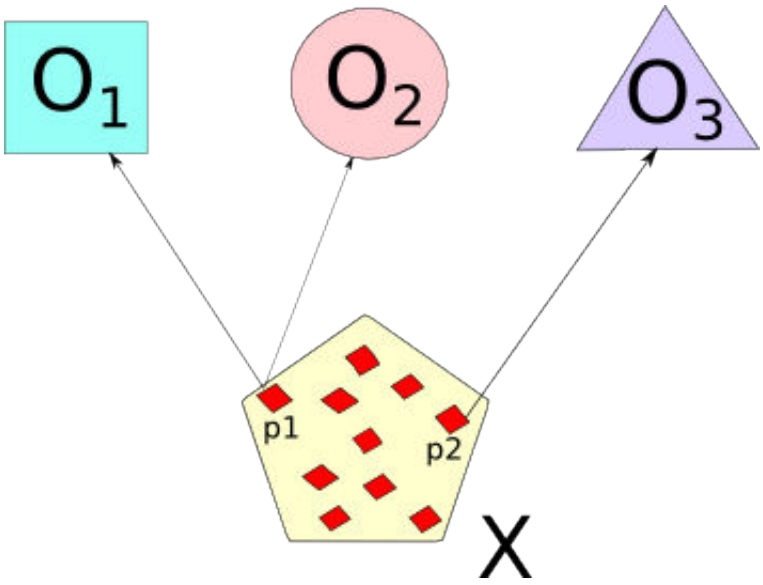

FIGURE 4.13: The idea of applying the Dempster-Shafer theory in this scenario.

4.2.4 Construction of Mass Values

As discussed above, we construct mass values based on extracted feature points of the object. The general idea is to compare the detected feature points with the ones stored in the training base. First, let us denote $\Delta(p_i, p_j)$ the normalized distance between two feature points p_i and p_j; the shorter the distance is, the more similar the two feature points are.

$$\Delta(p_i, p_j) \in [0, 1] \tag{4.4}$$

We call X the set of extracted feature points of the target object in an input image and consider each feature point $p_i^X \in X$. Suppose that we want to evaluate the matching between a feature point p_i^X and a training image M whose class is previously known as $O_j \in \Omega$, we consider the idea in the work of [45]. Indeed, we will find the two nearest

	p_1^X	p_2^X	p_3^X	...	$p_{\|X\|}^X$
O_1	■			...	
O_2		■		...	■
O_3	■		■	...	■
...	...	...	...	...	...
O_N		■		■	

TABLE 4.2: Matching between the feature points of input image X and the classes

neighbours of p_i^X in M, i.e. the two feature points of M that are the most similar to p_i^X, called $p_{i_1}^M$ and $p_{i_2}^M$. The feature points in M are absolutely extracted in the training phase before that. These two nearest neighbours then allow us to determine a solid matching between the target and the trained object. We suppose that $p_{i_1}^M$ is closer to p_i^X than $p_{i_2}^M$ i.e. $\Delta(p_i^X, p_{i_1}^M) \leq \Delta(p_i^X, p_{i_2}^M)$. After that, we define a matching function between the feature point p_i^X of an input image X and the model M:

$$\delta(p_i^X, M) = \begin{cases} 1, & \text{if } \Delta(p_i^X, p_{i_1}^M) \leq \alpha \text{ and } \frac{\Delta(p_i^X, p_{i_1}^M)}{\Delta(p_i^X, p_{i_2}^M)} \leq \beta \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

where α and β are two user-defined parameters such that $0 \leq \alpha, \beta \leq 1$. The former guarantees that the distance between p_i^X and its most similar feature point found in M is small enough whereas the latter helps to avoid false matching. In this work, we choose $\beta = 0.8$ as suggested in [45], and we add $\alpha = 0.25$ in order to reduce noises. Indeed, these two parameters help us to find a strong and distinctive matching between the feature point p_i^X and its closest feature point in M. If $\delta(p_i^X, M) = 1$, we then say that p_i^X matches to the training image M, meaning that it matches to the class $O_j \in \Omega$ of M. If $\delta(p_i^X, M) = 0$, we say that p_i^X does not match to O_j. By doing the same way, we can find all the matchings of the feature points in the input image X to the training image M.

Now we know how to define the matching between an input feature point and a training image, this is the basis to find the matching between an input image X and a class O_j. In order to do that, we consider the matching between each feature points $p_i^X \in X$ and the class O_j. In the case that O_j has several training images M_k, we choose the training image M_{max} that has the maximum number of matchings to X, according to Eq. (4.5):

$$\delta^{max}(p_i^X, O_j) = \delta(p_i^X, M_{max}) \tag{4.6}$$

Table 4.2 shows an example illustrating the matchings between input feature points and the output classes. A cell $c(p_i^X, O_j)$ implies the matching between the feature point p_i^X of X and the class O_j, $i = 1, 2, ...|X|$ - number of feature points in X, $j = 1, 2, ...N$ - number of classes. If the cell is red, it means that the feature point p_i^X matches to the class O_j (i.e. $\delta^{max}(p_i^X, O_j) = 1$), otherwise it does not match.

After we determine the matching between the input feature points and the output classes, we can construct the mass function as follow. Each feature point p_i^X will vote for

a hypothesis in the power set such that the hypothesis is composed of the classes that match to p_i^X. Mathematically, let's define a hypothesis-voted function that calculates the accumulated votes for each hypothesis:

$$accVote(X, H) = \sum_{p_i^X \in X} \phi(p_i^X, H), \quad H \in 2^\Omega \tag{4.7}$$

where $\phi(p_i^X, H)$ is a function indicating the matching between the feature point p_i^X and every element class in H:

$$\phi(p_i^X, H) = \begin{cases} 1, & \text{if } H \equiv \{O_j\},\ \delta^{max}(p_i^X, O_j) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

where $\delta^{max}(p_i^X, O_j)$ was already explained above. Indeed, $\phi(p_i^X, H)$ indicates whether a feature point p_i^X only matches to every element class in the hypothesis H or not, and $accVote(X, H)$ calculates the number of such feature points in X. After that, we calculate the mass function based on the hypothesis-voted function:

$$m^X(H) = \frac{accVote(X, H)}{G^X} \tag{4.9}$$

where G^X is the normalization factor that guaranties the condition in Eq. (4.3):

$$G^X = \sum_{H \in 2^\Omega, H \neq \emptyset} accVote(X, H) \tag{4.10}$$

It is worth noting that, in this work we assume that the class of object in the input image X is only in Ω, so we always put $m^X(\emptyset) = 0$.

Actually, the number of feature points can give different results in mass construction. However, we should not take too many of feature points because it may decrease the performance during runtime, there are also some feature points that are not useful and they make imprecision. In the application, we configure a parameter which allows deciding the quantity of feature points, which can found in Appendices.

Now we are able to construct a mass vector from a set of object's feature points of an input image captured by a camera. In the next section, we show how we combine these masses and make decision on the label of the object.

4.2.5 Combination and Decision

In Chapter 3, we already listed some interesting combinators that allows integrating information from multiple sources, and the Dempster-Shafer combinator is the selected one due to its strong properties and rationality in the work of color recognition. Similarly in this scenario, we find that this combinator is also the most appropriate choice because we consider a closed world in which the objects to be recognized are always in a predefined set. Moreover, we take into account the reliability of each camera (by testing individual

recognition test for each one), so we can avoid the problem of total conflict when combining the sources.

To remind, the Dempster-Shafer rule of combination was proposed in [64] under the following formula:

$$m_{DS}(H) = \frac{1}{1-k} \sum_{H_1 \cap H_2 \cap \ldots \cap H_S = H} \prod_{j=1}^{S} m_j(H_j), \ H \in 2^{\Omega}, \neq \emptyset$$
$$m_{DS}(\emptyset) = 0 \tag{4.11}$$

where $k = m_{Conj}(\emptyset)$ measures the conflict among S sources, as presented in Eq. (3.8).

Once we combine the masses given by the cameras into only one mass, the next step is to make decision on the class of object. As already discussed in Chapter 3, the maximum of belief is too pessimistic, and the maximum of plausibility is too optimistic, so we consider the maximum of pignistic probability ([65]):

$$betP_m(H) = \frac{1}{1 - m(\emptyset)} \sum_{H \in A} \frac{m(A)}{|A|} \tag{4.12}$$

Finally, the class of the detected object will be the singleton element of mass function that has the maximum value of pignistic probability. This is also the final decision of the NAO robot.

4.3 Illustrative Example

In this section, we provide an example to illustrate the proposed method. Suppose that the NAO robot is trained to recognize three classes of object, so we have the space of discernment:

$$\Omega = \{O_1, O_2, O_3\} \tag{4.13}$$

So there are 8 possible hypotheses in the power set:

$$2^{\Omega} = \{\emptyset, \{O_3\}, \{O_2\}, \{O_2 \cup O_3\}, \{O_1\}, \{O_1 \cup O_3\}, \{O_1 \cup O_2\}, \Omega\} \tag{4.14}$$

Suppose that the NAO robot captures the scene. It firstly extracts the feature points of the object, and we assume that there are 10 feature points $p_i \in X$, for simplicity. After that, we find the matching from each of these feature points to the training images as explained in Section 4.2.4. Each feature point can correspond to one or several object classes, and we give an example of these matchings by Table 4.3. In that table, each cell describes the matching between a feature point and a class; if $\delta^{max}(p_i, O_j) = 1$, the cell is red, otherwise it is left blank. The last row of the table indicates the hypotheses voted by the associating feature points. After this step, we obtain the degree of belief of each hypothesis in the powerset. For example, p_1 only matches to class O_1, so it votes for

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
O_1	■		■	■			■		■	
O_2		■			■	■			■	■
O_3				■			■	■		■
Vote for:	O_1	O_2	O_1	$O_1 \cup O_3$	O_2	O_2	$O_1 \cup O_3$	O_3	$O_1 \cup O_2$	$O_2 \cup O_3$

TABLE 4.3: Matching between the input feature points and the classes

$H \in 2^\Omega$	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	accVote	Mass value
$\emptyset$	0	0	0	0	0	0	0	0	0	0	0	0.00
O_3	0	0	0	0	0	0	0	**1**	0	0	1	1/10
O_2	0	**1**	0	0	**1**	**1**	0	0	0	0	3	3/10
$O_2 \cup O_3$	0	0	0	0	0	0	0	0	0	**1**	1	1/10
O_1	**1**	0	**1**	0	0	0	0	0	0	0	2	2/10
$O_1 \cup O_3$	0	0	0	**1**	0	0	**1**	0	0	0	2	2/10
$O_1 \cup O_2$	0	0	0	0	0	0	0	0	**1**	0	1	1/10
Ω	0	0	0	0	0	0	0	0	0	0	0	0/10

TABLE 4.4: Accumulated vote for each hypothesis.

$H = \{O_1\}$, meanwhile p_4 matches to both classes O_1 and O_3, so it votes for $H = \{O_1 \cup O_3\}$.

In the next step, we have to calculate the strength of each hypothesis in the power set, i.e. how many feature points vote for a specific hypothesis. To see that, we use Table 4.4 in which the first column shows all the hypotheses. Each cell from the second column to the eleventh column is the value of $\phi(p_i^X, H), H \in 2^\Omega$ (see Eq. (4.8)). Remind that if $\phi(p_i, H) = 1$, it means that the feature point p_i votes for the hypothesis H. The column $accVote$ indicates the number of votes by the feature points (see Eq. (4.7)), and the last column shows the value of mass associated to a hypothesis calculate by Eq. (4.9) and Eq. (4.10). Note that we have $G = \sum accVote = 1 + 3 + 1 + 2 + 2 + 1 + 0 = 10$

Now we already determine the mass values of the NAO robot in this example. We suppose to add an IP camera (2D) and an Axus Xtion camera (3D) in order to improve the quality of recognition. These cameras capture the same object and obtain different values. Fig. 4.14 shows an example of how three cameras see an object from their positions.

These two cameras also extract feature points of the object, then based on their matchings to the training base, we are able to obtain the two new mass vectors. We assume that the values of these masses (three cameras) are shown in Table 4.5. When looking at the table, the NAO robot gives more belief on the class O_2, the IP camera relies more on the class O_1, and the Axus camera mainly hesitates between O_1 and O_2, which makes the decision more interesting analyze. By using the Dempster-Shafer rule of combination, we calculate the combined mass which is shown in column m_{comb}. After that, column $BetP$ shows the pignistic probability of each singleton hypothesis, which

FIGURE 4.14: From left to right: an object is captured by the Axus camera, the NAO camera, and the IP camera.

Hypothesis	m_{NAO}	m_{IP}	m_{Axus}	m_{comb}	$BetP$	Decision
$\emptyset$	0.00	0.00	0.00	0.00		
O_3	0.10	0.23	0.21	0.22	0.23	
O_2	0.30	0.17	0.12	0.26	0.27	
$O_2 \cup O_3$	0.10	0.08	0.00	0.00		
O_1	**0.20**	**0.32**	**0.09**	**0.49**	**0.50**	O_1
$O_1 \cup O_3$	0.20	0.13	0.13	0.02		
$O_1 \cup O_2$	0.10	0.00	0.39	0.01		
Ω	0.00	0.07	0.06	0.00		

TABLE 4.5: Mass values from the sensors.

tell us that the decision should be the class O_1 due to its strong emergence compared to the others.

4.4 Experimental Results

4.4.1 Testing Strategy

We applied the proposed method in a scenario where a NAO robot is requested to recognize an object frontwards. First, the robot receives oral commands from human by using a voice recognition library implemented before. After it understands that it has to recognize the object, it employs two other cameras: an IP camera and an Axus camera for the recognition process. After having the result, the robot responds to human the name of the detected object through its loud speakers.

As mentioned before, the main concentration of this work is to demonstrate how we resolve the problems of uncertainties and imprecisions during the object recognition process. For that reason, we challenge the cameras by testing with confusing objects. Actually, we did three experiments, each of them contains a set of objects to be recognized, as shown in Fig. 4.15. In the first set, there are 4 cups which may cause uncertainty in their spatial structure so it is difficult if we have only 3D data of them. On the other hand, the second experiment contains 4 boxes that have similar branch information on their surface, which may limit the recognition of the 2D cameras. Finally in the third

set, we tested with 4 Lego bricks which are considered to be difficult for both 2D and 3D cameras to recognize.

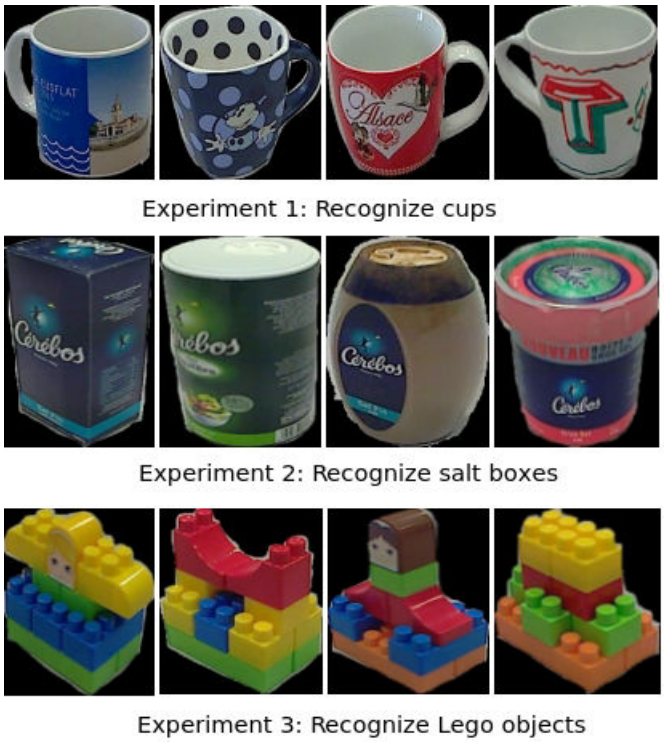

FIGURE 4.15: The tested objects for the recognition system.

In the first place, the training base has to be constructed. For each camera, we train two images of each object in different view points. The number of training images can affect to the performance of the system, and we experimentally find that having only two images is reasonable. We then manually remove the background in these images in order to have only the model objects. The feature points of the objects are also pre-extracted to save calculations during runtime.

For the test phase, each object is put in front of the NAO robot. The IP camera and the Axus camera are on the two sides of the robot to help it improve the recognition. These cameras capture the same scene at the same time whenever the robot wants to recognize the object in the scene. To focus on the work of recognition, the image region containing the object is restricted in order to avoid ambiguity in the scene. For each of the three experiments, we did 32 recognition tests with different objects of 4 classes (so 8 tests for each object). The tested objects are turned around and put in different angles to the cameras in each test in order to challenge the uncertainty. For example, when recognizing a cup 8 times, the NAO camera may find that at some views, the cup looks totally different if compared to the training images of the robot. However, at these difficult cases, the other cameras may get better view such that the cup matches well to their training base. This is interesting to consider since we combine different types of cameras in different view points to recognize an object.

Camera	NAO (2D)	IP (2D)	Dempster-Shafer Fusion
Experiment 1	78%	88%	**97%**
Experiment 2	72%	72%	**81%**
Experiment 3	59%	59%	**75%**
Average	69.67%	73%	**84.33%**

TABLE 4.6: Object recognition using two cameras 2D: NAO and IP camera.

4.4.2 Results and Analyses

In Chapter 3, we present the experimental results by using two and three homogeneous sensors (2D cameras), and for this case of the object recognition, we do the same thing. The idea is to demonstrate how fusion strongly influences the recognition results comparing to each individual camera and their combinations.

Two 2D cameras

Table 4.6 shows the results by using only two 2D cameras: NAO and IP camera, without the contribution of the Axus camera. As mentioned above, we did three experiments with three sets of objects. The second and the third column are the results by using only the NAO camera, and the IP camera, respectively. Notably, using one camera means that we apply the same method of recognition described in Section 4.2.3 but there is no combination with other cameras, and the decision method is still chosen by the maximum of pignistic probability. The final colum shows the result using the fusion of these two cameras.

From the Table 4.6 we see that the highest recognition rate belongs to the experiment 1. This is reasonable according to what is already explained: the cups are totally different in their surfaces, which is well recognized by 2D processing. However, the second and the third experiment do not give high recognition rates because these objects are difficult for 2D cameras to recognize (see Fig. 4.15). In average, the result by using fusion (84.33%) is much better than using individual cameras (69.67% and 73%).

One 2D camera and one 3D camera

Table 4.7 shows the results of using one 2D camera and one 3D camera: IP and Axus. The reason for choosing the IP camera instead of the NAO's because the IP camera shows better individual results (see Table 4.6).

In this test, the recognition rate of the first experiment remains the same, but the rate of the second experiment increases much more from the previous test with two 2D cameras (100% and 81%). These numbers are coherent because the second experiment, as indicated before, contains the objects that are difficult for 2D processing but much easier for 3D processing (boxes of salt in different shapes). The average of recognition by

Camera	IP (2D)	Axus (3D)	Dempster-Shafer Fusion
Experiment 1	88%	75%	**97%**
Experiment 2	72%	91%	**100%**
Experiment 3	59%	69%	**75%**
Average	73%	78%	**90.67**%

TABLE 4.7: Object recognition using one camera 2D and one camera 3D: IP and Axus camera.

Camera	NAO (2D)	IP (2D)	Axus (3D)	Dempster-Shafer Fusion
Experiment 1	78%	88%	75%	**97%**
Experiment 2	72%	72%	91%	**97%**
Experiment 3	59%	59%	69%	**84%**
Average	69.67%	73%	78%	**92.67%**

TABLE 4.8: Objec recognition using two 2D cameras and one 3D camera: NAO, IP, and Axus camera.

using fusion of these two cameras (90.67%) is better than the use of two 2D cameras in the previous test (84.33%).

Three cameras

Now we test with the fusion of the three heterogeneous cameras: NAO camera, IP camera, and Axus camera. In the first experiment, the Axus camera does not contribute to the recognition rate, but it is really emerge in the second experiment. The third experiment shows lower recognition rate for three cameras because the tested objects in this case are difficult for both 2D and 3D processing. In average, the results of using these three cameras give better recognition rate (92.67%) than the results of two cameras that have been shown above (84.33% and 90.67%). This interesting outcome allows us to give a strong conclusion that in this case of the object recognition, the fusion of heterogeneous sensors increases dramatically the recognition rate.

4.5 Conclusion of Chapter

In the previous chapter, we propose to use multi-camera system to recognize coloured ball for a NAO robot. The sensors are considered to be homogeneous since they are all 2D cameras. This chapter considers a more general fusing scenario where we apply heterogeneous sensors fusion to help the NAO robot recognize an object, which allows reducing uncertainties and imprecisions caused by many factors such as lighting conditions, viewing angles, and similarities among confusing objects.

The NAO robot uses one camera on its head and another IP camera. These two 2D cameras are combined with a 3D camera, Axus Xtion Pro. The combination of these camera types bring advantages because 2D cameras can recognize well the characteristics on the surface of the objects such as colors, intensity, contrast... meanwhile 3D cameras can handle well the spatial information about the structure of the objects.

In the scenario, when the NAO robot is requested to recognize an object frontwards, it also calls the IP and the Axus camera to form a multi-camera recognition system. These cameras capture the same scene containing the object, then they build mass functions and the Dempster-Shafer theory is used to combine these masses. Finally, the decision on the object class is made by using the maximum of pignistic probability. The robot then says the recognition result to human.

Actually, each camera extracts feature points of the object. The feature points carry rich information about the local image structure around them, and they characterize well the patterns in the image. From the extracted feature points, we compare them to the training database. Each feature point may correspond to one or several object classes, so they vote for a hypothesis in the powerset of the Dempster-Shafer theory. From that, we are able to construct mass values for the cameras.

We applied the proposed method in the NAO robot and validated it by three experiments. For each one, the objects are turned around and the multi-camera system recognizes it, reports the results to human. We also compare the fusion results by the Dempster-Shafer combination with the single results when using only NAO camera, or IP camera, or Axus camera individually. In average, the fusion results given by the Dempters-Shafer theory improves dramatically the recognition rate.

Chapter 5

Conclusion and Perspectives

5.1 Thesis Review

As the matter of fact, the domain of robotics attracts a lot of researches due to its important roles in our modern life. However, during the operation of a robot, it may find difficult to make decisions due to the effect of uncertainties and imprecisions coming from the quality of sensors or from the exploited environment. For that reason, the fusion of multi-sensor is taken into account and it is considered as the most appropriate approach to deal with such problems.

In this thesis, we concentrate on the multi-sensor fusion to improve the reliability of a humanoid NAO robot. In the first place, the robot is requested to find an object whose color is described in human terms. In the second scenario, it is requested to recognize an object in front of it. The two scenarios require the robot to process with visual information, and we use a camera on its head to do the task. In addition, we add more external cameras to from a multi-camera system for the detection and recognition.

In the scenario of the color detection, we test with colored balls under different lighting conditions and variation of color hue. The color space RGB is not considered as a good choice as the CIE-L*a*b and the HSV. The NAO robot recognizes human command then walks around to find the target. To detect balls' shapes in images, we apply the Hough transformation. The average pixel values of the detect ball are then used as the inputs for a Fuzzy system. We choose the type Sugeno for the Fuzzy system due to its light calculation and structural properties.

Indeed, the construction of the Fuzzy system is based on the perceptual evaluation i.e. we define the linguistic labels for each component and give a color for each possible combination. Each color is assigned a constant number and the output of the Fuzzy system is a numerical value which specifies the color detected. This leads to the introduction of a threshold value ϵ which defines for each constant number a the certain range in which the fallen Fuzzy output specifies the color of that constant. This threshold gives a compromise between uncertainty and reliability of the Fuzzy system. If it is big, the uncertainty may be decreased but the imprecision can arise, and vice versa. Moreover, the uncertainty and imprecision may also come from other factors such as lighting conditions or the quality of sensor. For those reasons, the use of additional sources is necessary, and we create a homogeneous multi-camera system composed of the NAO camera and two 2D external cameras.

The Dempster-Shafer theory is chosen to for the fusion of multi-camera. Indeed, from the Sugeno output of each camera, we construct a mass function based on the threshold value ϵ. We also review some of well-known operators to combine the masses and the one proposed by Dempster-Shafer is selected. To derive the final output, we choose the class that has the maximum value of pignistic probability. We show the experimental results with three cameras and their fusion is seen as better results comparing to each single camera.

For the case of object recognition, the NAO robot is requested to recognize an object in front of it. The objects are previously trained and stored in the learning base. Our idea is that we extract the feature points of the detected object and compare them to the ones in the training base. In this scenario, we employ heterogeneous camera sensors to deal with the recognition. Two 2D cameras (the NAO robot and an IP camera) and one 3D camera (Axus Xtion Pro) are used to form the multi-camera system. To extract feature points with the 2D cameras, the SURF technique is used, and with the 3D camera, the SHOT descriptor is employed. The Dempster-Shafer theory is again chosen for the fusion of these cameras.

Based on the correspondences between the feature points of the detected object and the feature points stored in the learning base, we construct mass values for each camera. The Dempster-Shafer operator is then used to combine the masses, and the maximum of pignistic probability is used to make the final decision. The combination of these heterogeneous sensors brings many advantages since the 2D data gives us information about the characteristics of the object's surface while the 3D data describes geometrical information of the object's form. In the test, we choose the objects that have many similarities, and they are turned around to challenge uncertainties and imprecisions. The experimental results show that the fusion of multi-camera is better than the recognition result of each single camera.

5.2 Thesis Conclusion

The questions proposed in the thesis are already presented in Chapter 1, and we remind them here with the answers. The first question concerns that can the NAO robot accomplish well its tasks against uncertainties and imprecisions using only one camera sensor? And the second question focuses on the improvement with the use of multi-camera comparing to a single camera.

Ideally, the NAO robot should operate autonomously in its tasks, that is, there is no need to add external information sources. However, in reality, we cannot avoid the problem of uncertainties and imprecisions that may come from the robot itself or from the exploited environment, and they often lead to decreasing of system reliability. Indeed, Chapter 2 emphasizes this confirmation. The NAO robot uses only one single camera (on its head) to detect a requested colored ball by using the Sugeno Fuzzy system, but the results are too low: 43.61% and 51.94% (in HSV space) when ϵ is 0.1 and 0.2, respectively. Despite the low values of threshold ϵ, these detection rates are not acceptable in a real-time system.

In Chapter 3 with the use of homogeneous sensor fusion to resolve the problem of uncertainties and imprecisions, we test with the NAO camera, an IP camera, and a webcam. The experimental results demonstrate that the fusion of these three cameras gives a better detection rate (75.56% in HSV space) comparing to the single results of cameras (47.50%, 38.61%, and 43.06%). It is not to say that in every case, the fusion is better, but in average it should be. That is the reason why we test with many images for each colored ball (40×9), and the result confirms the hypothesis. Additionally, in Chapter 4 that we show the object recognition of the NAO robot, the fusion of heterogeneous camera sensors also gives better recognition rate than each single camera. After three sub-experiments, the average recognition rates for the cameras are 69.67%, 73%, and 78%, respectively, whereas the fusion of these three cameras gives a rate of 92.67% i.e. an improvement of approximately 20% in recognition rate.

In conclusion, we say that the fusion of multi-sensor helps the NAO robot improves the reliability in his operations against the difficulties of uncertainties and imprecisions it deals with when using only a single information source. Actually, to validate the efficiencies of fusion, we need to test it in many domains with a huge number of scenarios, and that requires the collaboration of so many researches. However, this thesis contributes the confirmation at least in the experiments with the NAO robot with the color and object recognition. We have tried to validate with various number of sources (1, 2, and 3 cameras) and different types of fusion (homogeneous and heterogeneous), also different scenarios (color detection and object recognition). For that reason, we strongly believe in the contribution of this thesis to the research community in multi-sensor fusion for decision-making robotics.

5.3 Discussion

In the results of the color detection by using multi-camera fusion (Chapter 3), we also present the detection rate by using two cameras: the NAO camera and the IP camera, before showing the fusion result of three cameras (add another webcam). Theoretically, when we have more sources, we gain more information and the fusion should work better. However, this does not always happen. For example the detection rate for the color pink and red when using two cameras (85% and 92.50%) are better than using three cameras (82.50% and 87.50%). Anyway, the fusion of three cameras gives a better average result for all colors than the fusion of two cameras (75.56% and 71.94%). For this, we can say that the third webcam contributes to improve the detection rate.

It is also important to note that the results of single cameras are not good because the threshold values chosen are low ($\epsilon = 0.1$ and 0.2 in this thesis). However, as analysed in Section 2.5.1, a big value of threshold may decrease the reliability of the detection system, but a small value leads to many cases of uncertainty. Our objective here is not to try to show an excellent detection rate but we focus on how to resolve uncertainties and improve the reliability of the detection system. We emphasize on the hypothesis that in average, the fusion of multi-camera improves dramatically the reliability of the color detection system when using only a camera individually.

On the other hand, we also present the fusion results with a combination of each two cameras and three cameras in the work of the object recognition (Chapter 4). In average,

the fusion of three cameras gives better result (92.67%) than using two cameras (84.33% and 94.67%), except the case of the experiment 2 where the NAO camera decreases the recognition rate of the two other cameras (100% down to 97%). This can be explained that in the experiment 2, the tested objects are difficult for 2D cameras because they have many similar visual information on surface. Actually, we already consider the discounting factor in the mass function so that the added camera should improve the recognition rate, however in this experiment, we believe that the number of test must be huge. It explains the reason that for only experiment 2, the fusion of two cameras is better than the fusion of three cameras, but in average for all these three experiments (three sets of objects), the fusion of three cameras are better.

It is also worth noting that in the case of the object recognition by the heterogeneous sensors, we have to set up some parameters, especially with the Axus camera. For instance, to capture 3D images, we need to define the sampling radius of the captured image, as well as the radius for the SHOT descriptor. Change of these parameters lead to the modification in the number and characteristics of extracted feature points, and may change the recognition results. However, in literature there is no specific method for choosing these parameters, because they really depend on the application. In this scenario, we have tested several times to choose the appropriate parameters that allow us to well detect feature points of the objects.

5.4 Limitations of the Research

The thesis well concludes the advantage of multi-sensor data fusion by experimental results on the NAO robot. However, it remains some limitations that we may take into account for the next works.

First, the NAO robot actually has two cameras on its head and theoretically we should use both of them for the fusion. However, they do not share their view, that is, the viewing angles of these cameras are separated, so they cannot capture the same scene at the same time. For that reason, we have to add external cameras (which is not so convenient), but we can also use another robot whose multi-camera is easier for fusion in the next work.

Second, in the work of object recognition, we extract the feature points of the object. However, the object is present in a scene that may be affected by other noises, and the extracted feature points may come from other things instead of the target object. To avoid that problem, the object are put in the scene where we limit as much as possible the presence of other objects so that we extract well the feature points of the target object. In the future work, we are going to consider about the segmentation to deal with such problem.

5.5 Recommendation for Further Researches

The thesis finishes and remains some directions for the future researches which are going to improve the current results or to retake the results of this work for the next, or extend them with more complicated scenarios.

In fact, for both scenarios of color and object recognition, we have to define some parameters to get the system work. For instance, with the Fuzzy system to detect color in Chapter 2, we define the fuzzification and inference rules based on perceptual evaluation. Additionally, in the work of object recognition, the two parameters α , β in Eq. (4.5) are defined manually. In the future works, such parameters may be chosen optimally by using an evolution approach such as the Genetic Algorithm.

In further researches, we may consider more complicated scenarios to do the fusion. The type of sensors will be more diversified, for example we can combine the sonar sensor and the camera of the robot for a recognition scenario. Other recognition techniques can be consulted, for example the Evidential k-nearest neighbors in [79]. The fusion can also be validated with some other approaches such as the Extended Kalman Filter, or the Bayesian Network, along with the Dempster-Shafer theory.

Publications

- Thanh Long Nguyen, Didier Coquin, Reda Boukezzoula. **Recognition of Confusing Objects for NAO Robot.** Carvalho, J.P., Lesot, M.-J., Kaymak, U., Vieira, S., Bouchon-Meunier, B., Yager, R.R. (Eds.). *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Jun 2016, Eindhoven, Netherlands. Springer International Publishing Switzerland 2016, CCIS 610 (PART 1), pp.262-273, 2016, Communications in Computer and Information Science <10.1007/978-3-319-40596-4_23>

- Thanh Long Nguyen, Reda Boukezzoula, Didier Coquin, Stéphane Perrin. **Combination of Sugeno Fuzzy System and Evidence Theory for NAO Robot in Colors Recognition**. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2015)*, Aug 2015, Istanbul, Turkey. IEEE Conference Publication, pp.1-8, 2015, <10.1109/FUZZ-IEEE.2015.7337900>

- Thanh Long Nguyen, Reda Boukezzoula, Didier Coquin, Eric Benoit, Stéphane Perrin. **Interaction between Humans, NAO Robot and Multiple Cameras for Colored Objects Recognition using Information Fusion**. *IEEE, Proceedings of the 8th International Conference on Human System Interaction (HSI 2015)*, Jun 2015, Warsaw, Poland. IEEE Conference Publications, pp.322-328, 2015, <10.1109/HSI.2015.7170687>

- Thanh Long Nguyen, Reda Boukezzoula, Didier Coquin, Stephane Perrin. **Color Recognition for NAO Robot Using Sugeno Fuzzy System and Evidence Theory**. *Atlantis Press. 16th World Congress of the International Fuzzy Systems Association (IFSA)*, Jun 2015, Gijon, Spain. EUSFLAT-15, pp.1176-1183, 2015, <10.2991/ifsa-eusflat-15.2015.166>

Appendix A

Software Platform of the NAO Robot

A.1 NAO Robot as a Platform of the Work

Although the use of multi-camera is not restricted to any kind of vision system, we still apply the proposed method in a NAO robot due to the development of our projects related to robotics. Therefore in this appendix we would like to introduce this interesting platform.

The NAO robot was developed by the Aldebaran-Softbank Robotics company (https://www.ald.softbankrobotics.com/fr) and it is their first humanoid robot, with the height of 58 cm. The NAO robot used in this work is the fourth version and is equipped with several sensors as well as 25 degrees of freedom (DoF) which ease the robot's motion. Figure A.1 shows the description of NAO.

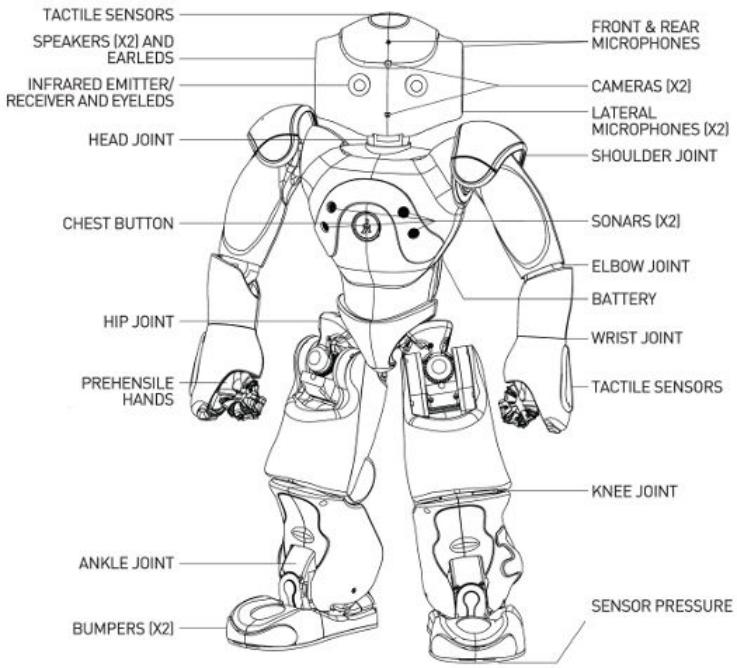

FIGURE A.1: The NAO robot and its components.
Image from: http://doc.aldebaran.com/1-14/family/nao_h25/index_h25.html.

For the vision system, NAO has two HD cameras on its head. They do not have any common in their views: one scans the upper space and the other is mainly responsible for the lower space. For the sound processing system, it is equipped with four microphones which can help it in speech recognition or sound localization. It also has two loud-speakers located at the two sides of the head. In order to detect obstacles during movement, the robot can use the sonar sensors on the chest. There are also tactile sensors attached to its hands and on top of the head. Additionally, the robot has two bumpers on two feet to detect dangerous contacts.

A.2 Software In and Out of the Robot

The NAO robot can come with two types of software (see Figure A.2):

- Embedded Software: Programs running on the motherboard located inside the head of NAO, which allow autonomous behaviours. The OpenNAO is the operating system and it is an embedded GNU/Linux distribution based on Gentoo. NAOqi is the main software that controls the NAO robot and this specific program advertises the modules and the methods of NAO as behaviours.
- Desktop Software: Programs running on our computer, allowing to create behaviours and remote control of the robot. The manufacturer provides Choregraphe, a visual programming language allowing to create animations and behaviours easily. Additionally, the Monitor software is a tool dedicated to give us an elementary feedback from the robot and a simple access to its cameras.

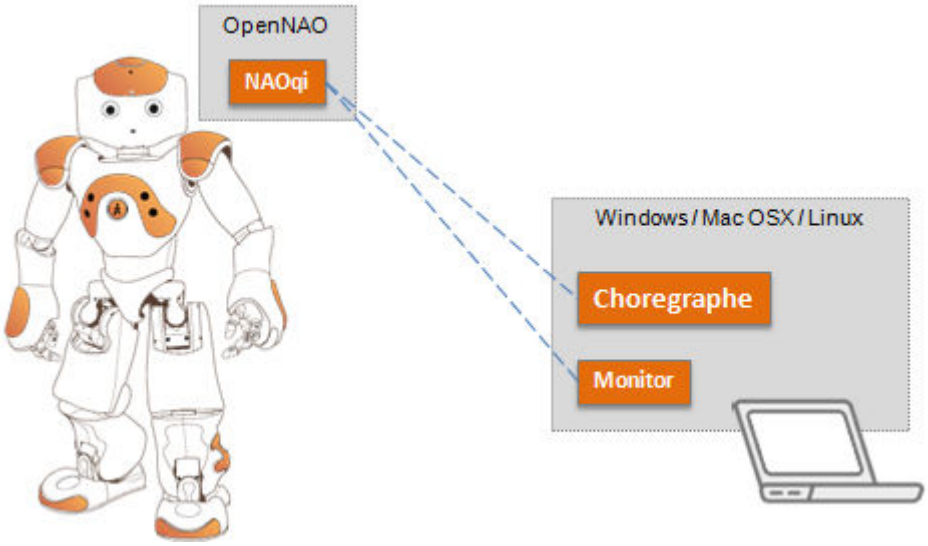

FIGURE A.2: Software in and out of NAO.
Image from:
http://doc.aldebaran.com/1-14/getting_started/software_in_and_out.html.

A.3 Programming Guide

A.3.1 NAOqi Framework

The NAOqi is the framework used to program for NAO, which is responsible for managing parallelism, resources, synchronization, events... It also allows homogeneous communication between different modules, homogeneous programming and homogeneous information sharing. This framework is cross-platform, cross-language, and it provides introspection. The native language for the robot is C++, however, we can develop software by some other languages such as Python, Java, and even Matlab.

The NAOqi process The NAOqi excutable running on the robot works as a broker. In fact, it loads a preference file containing the libraries to be run at initial time. Each library contains one or more modules that use the broker to advertise their methods. We are able to find the advertised methods in tree or through network by using the lookup service provided by the broker. Figure A.3 describes the process in which the NAOqi receives request from network, then it loads modules which then call their associated methods.

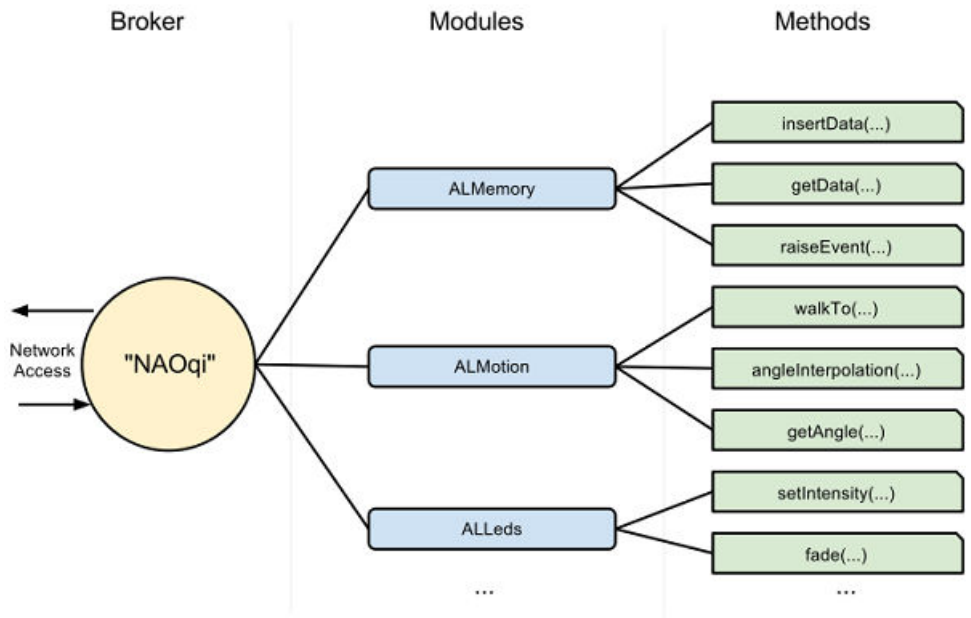

FIGURE A.3: NAOqi process.
Image from: http://doc.aldebaran.com/1-14/dev/naoqi/index.html.

Modules Each module is typically a class within a library, which will be instantiated when the library is loaded from the preference file. The name and the methods of a module can be available for calling from others if it advertises the methods to the broker. There are two types of modules:

- Remote module: Through a network. To communicate with others, a module needs a broker which is responsible for all the networking parts. This kind of module is

compiled as an executable file, and can be run outside the robot. Remote modules can be debugged from an external computer so it is easier to develop. However, in terms of performance, it is much slower than local modules.

- Local module: A local module is compiled as a library and can be used only in side the robot. However, its performance is much better than remote modules. Local modules are launched in the same process, so they can speak to others by using only one broker and they can share variables and directly call others' methods.

A.3.2 Creating a module

In order to start programming a module for NAO, we firstly have to install C++ SDK for the NAO robot development, then install qiBuild, a tool designed to generate cross-platform projects using CMake. The operating system used in this work is a 12.04 LTS Ubuntu distribution installed on an HP Probook core-i5 computer.

For example we create a remote module allowing the robot to say a phrase "Hello World" by using its loud-speakers. We firstly create a source file of the module as shown in the code lines below. A proxy is specialized to transfer texts to speeches is instantiated, then we use it to command the robot to say something.

```
#include <iostream>
#include <alerror/alerror.h>
#include <alproxies/altexttospeechproxy.h>

int main(int argc, char* argv[])
{
  // The phrase to be said.
  const std::string phrase = "Hello world";

  try
  {
    // Create an ALTextToSpeechProxy to use its SAY method.
    // Arguments are: IP address of the robot,
    // and listening port (default 9559)
    AL::ALTextToSpeechProxy text2Speech(argv[1], 9559);

        // Ask the robot to say the desired phrase.
    text2Speech.say(phrase);
  }
  catch (const AL::ALError& e)
  {
    std::cerr << "Exception: " << e.what() << std::endl;
    exit(1);
  }
  exit(0);
}
```

After that we have to prepare a CMakeLists.txt file which is used to build the project:

```
cmake_minimum_required(VERSION 2.6.4 FATAL_ERROR)
# Name of the project.
project(helloworld)

# Enable using the qibuild framework.
find_package(qibuild)

# Create an executable named helloworld with the
# source file : helloworld.cpp
qi_create_bin(helloworld helloworld.cpp)

# Declare that the executable file uses the package ALCOMMON.
qi_use_lib(helloworld ALCOMMON)
```

Before doing that, we have to prepare a toolchain by using the qibuild framework. This toolchain helps us be able to cross-compile the code to run on the robot. In Linux, go to the terminal and type the following command:

```
qitoolchain create toolchain-name \path\to\the\cross-toolchain\
```

After that move to the folder containing the source code and the CMake file shown above, and compile them:

```
qibuild configure -c toolchain-name
qibuild make -c toolchain-name
```

It that step is successful, we are able to find an executable file created which allows to request the robot to say "Hello World". When executing this file in terminal, remember that we have to add two arguments, the first one is the IP address of the NAO robot, and the second one is the port where the robot receives the command, which is normally 9559.

Appendix B

Software Implementation in Color Recognition of NAO robot

This appendix presents some practical information about the software implementation in the scenario of color detection described in Chapter 2 and 3. In general, the NAO robot employs one of its camera in combination with an IP camera to search for a requested coloured ball. Fig. B.1 shows the processing flow to do this. The NAO robot recognizes an oral command from human to find a coloured ball, then it starts moving around for searching the ball. During the robot's movement, the two cameras (NAO's and IP) capture the scenes in front of the NAO, so these two processes are in parallel. Each camera gives a Fuzzy result about the detected ball, then their Dempster-Shafer fusion result will decide the color of the ball. If it is the target ball, the robot stops its motion then responds to human, otherwise it continues the search.

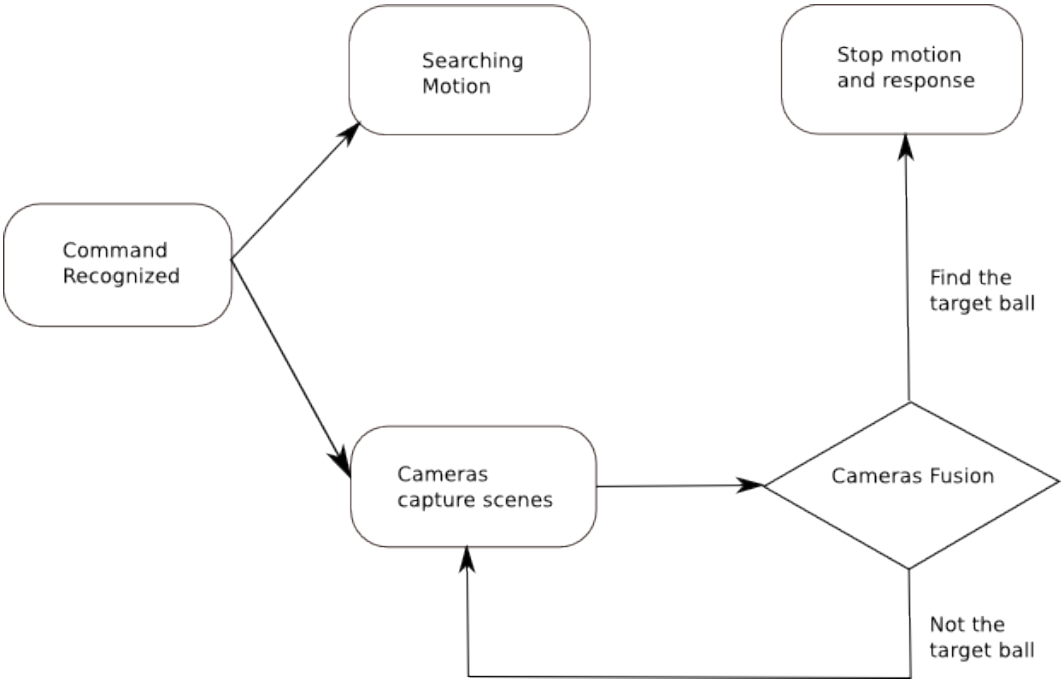

FIGURE B.1: NAO's flow to search target ball.

B.1 NAO's Speech Recognition

In the scenario, human's command to request the robot to find a ball is in the following form: "NAO, find the C_i ball" where C_i is a color in the predefined set discussed in Chapter 2. In order to recognize oral commands, NAO uses 4 microphones located on its head. There is also an implemented module for recognizing speeches in NAO, named ALSpeechRecognition. This modules allows the robot to recognize predefined words or phrases in several languages.

To use the speech recognition module, we firstly have to create a proxy to the ALSpeechRecognition module, so we can use its advertised methods by the broker. After that, we define the list of phrases that the robot will recognize:

```
// Create a proxy to the module.
AL::ALSpeechRecognitionProxy fSpeechRecog;

// Set the speaking language.
fSpeechRecog.setLanguage("English");

// Define the oral commands.
colorSearchingCommands = new std::string[9];
colorSearchingCommands[0] = "NAO, find the Blue ball";
colorSearchingCommands[1] = "NAO, find the Purple ball";
colorSearchingCommands[2] = "NAO, find the Pink ball";
colorSearchingCommands[3] = "NAO, find the Red ball";
colorSearchingCommands[4] = "NAO, find the Brown ball";
colorSearchingCommands[5] = "NAO, find the Orange ball";
colorSearchingCommands[6] = "NAO, find the Yellow ball";
colorSearchingCommands[7] = "NAO, find the Green ball";
colorSearchingCommands[8] = "NAO, find the Cyan ball";

std::vector<std::string> speechCommands;

for (int i = 0; i < 9; i++) {
        speechCommands.push_back(colorSearchingCommands[i]);
}

// Add the oral commands to the predefined list.
fSpeechRecog.setWordListAsVocabulary(speechCommands);
```

Once the ALSpeechRecognition module is started, it places in the memory of robot a key named SpeechDetected. This is a boolean value specifying whether a speaker is detected or not. If a speaker is heard, the phrase in the predefined list that best matches to the caught sentence will be placed in the key WordDetected in the robot's memory. This key is a structure organized as follow:

$$\{phra_1,\ conf_1,\ phra_2,\ conf_2,\ ...,\ phra_n,\ conf_n\} \tag{B.1}$$

where $phra_i$ is one of the predefined phrases and $conf_i$ is an estimate of the probability that this phrase matches to the one pronounced by the speaker. We select the phrase that has the maximum confidence (the first one) for processing.

For the next step, we should define a handler for the event that NAO recognizes a phrase. Indeed, we define the behaviour for the function onSpeechRecognize which is automatically called when a phrase is recognized. We also have to bind this method to the broker so that it can be called from outside.

```
// Bind the method onSpeechRecognized to the broker.
functionName("onSpeechRecognized",
                getName(),
                "This function will response to oral commands.");
BIND_METHOD(NAOIPCamImagesFuzzyFusionAppNAO::onSpeechRecognized);
```

Then we provide the implementation of the function onSpeechRecognized. This function defines the behaviours that the NAO will act each time it recognizes a phrase. In this scenario, the robot will starts moving around to find the ball:

```
void onSpeechRecognized(const std::string& name,
                        const AL::ALValue& val,
                        const std::string& myName);
```

B.2 NAO's Motion to Find Target Ball

Whenever the NAO robot recognizes a command from human that it has to find a ball C_i, it starts moving around to find the ball. In fact, to control the robot's motion, we use the ALMotion module provided in NAO's API. This module allows the robot to make movement and control joints, and to use this module we have to create a proxy to it:

```
AL::ALMotionProxy fMotion;
```

By default, all joints of the robot are in idle state, i.e. there is no energy, so in order to control the motion we firstly have to activate them, for example by the following code:

```
AL::ALValue time = 1.0f;
float stiffness = 1.0f;

fMotion.stiffnessInterpolation("JointActuators", stiffness, time);
```

The above code lines request all the joint actuators of the robot to set up 100% of energy and this is done within 1 second by using an interpolation method. After that we ask the robot to stand up and initialize the movement's parameter. The macro MOTION_POSTURE_START_STANDING_SPEED defines the speed for this action. It is worth noting that in later shown codes we also use some macro definitions like that.

```
// NAO stands up.
fPosture.goToPosture("StandInit",
                     MOTION_POSTURE_START_STANDING_SPEED);
```

```
// Initialize movement.
fMotion.moveInit();
```

Once the robot is in a ready state for the movement, we can request it to walk to a specific location by the following method:

```
fMotion.post.move(MOTION_MOVE_SEARCHING_X_SPEED,
                  MOTION_MOVE_SEARCHING_Y_SPEED,
                  MOTION_MOVE_SEARCHING_Z_SPEED);
```

In this method, the three parameters are velocity in X axis, Y axis, and around Z axis, respectively. For the three axes, we refer to Fig. B.2. Note that in this case we should use the POST method of the proxy fMotion because this allows us to create a bahaviour (move) in a parallel process since we want the robot to capture the scenes during its movement.

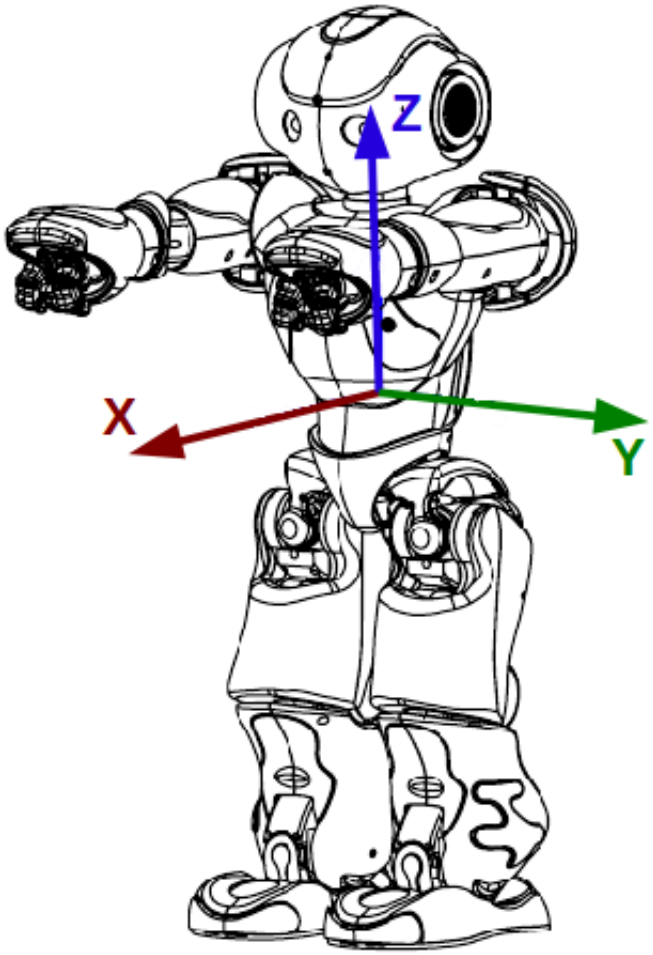

FIGURE B.2: NAO and coordinate system.
Image from: http://doc.aldebaran.com/1-14/naoqi/motion/index.html#almotion.

B.3 Two Cameras to Detect Balls

In order to access the NAO's camera, we have to prepare a proxy to the ALVideoDevice module, then subscribe to the camera. For the method's parameters, we provide the module's name, the index of the desired camera (0 for the upper camera and 1 for the lower camera), as well as the resolution, color space, and desired frame rate.

```
// Create a proxy to this module.
AL::ALVideoDeviceProxy fCam;

// Subscribe to the first camera for capturing images later.
const std::string clientName = fCam.subscribeCamera(
                                        APP_MODULE_NAME,
                                        APP_FIRST_CAMERA_INDEX,
                                        APP_CAPTURE_RESOLUTION,
                                        APP_CAPTURE_COLOR_SPACE,
                                        APP_CAPTURE_FRAMERATE);
```

After that we capture the scene by continuously (in loop) get each image from the camera, using the following method (we use a remote module in this case):

```
AL::ALValue img = fCam.getImageRemote(clientName);
```

Then we process with the structure *img* containing captured image's information.

In order to access the IP's camera, we employs the VideoCapture class of the OpenCV library, which connect to the camera through its IP address:

```
cv::VideoCapture cmrStream(APP_IPCAM_ADDRESS);
```

After that, we can use the variable *cmrStream* to continuously capture the scene, then process with each captured image:

```
cv::Mat src;

cmrStream.read(src);
```

For each image gotten from each camera, we try to detect the ball's shape in the image by employing the Hough transformation which is already implemented in the OpenCV. This library provides the HoughCircles method to find circles in a grayscale image using the Hough transformation. We can refer to this function in http://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=houghcircles#houghcircles. Note that we have to convert the input image to a grayscale one before using. This function returns a set of detected circles with the first one has the highest certainty which is chosen as the detected ball. Indeed, due to the concentration on the work of color detection, we setup the balls such that there is only one ball appearing in front of the NAO at each instance.

```
void HoughCircles(InputArray image, OutputArray circles,
                  int method, double dp, double minDist,
                  double param1=100, double param2=100,
                  int minRadius=0, int maxRadius=0 )
```

After finding the ball's shape, we consider all the pixels of the ball and calculate the average values of them, then we convert these values into HSV or Lab space according to the demand. Fig. B.3 shows the conversion algorithm from RGB to HSV color space. For the conversion from RGB to Lab we normally convert RGB to XYZ then from XYZ to Lab (more information in: http://www.easyrgb.com/?X=MATH).

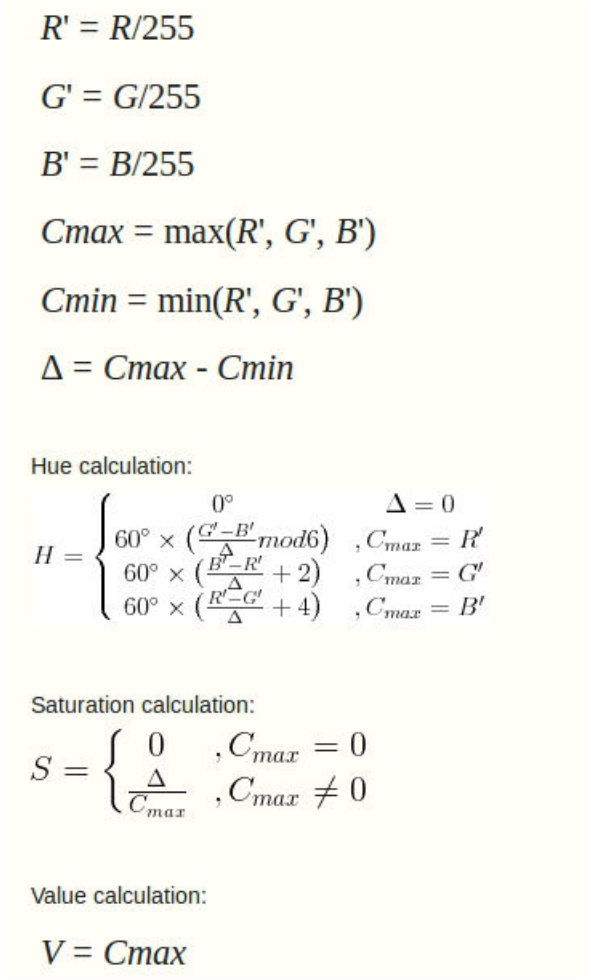

FIGURE B.3: Conversion from RGB to HSV.
Source: http://www.rapidtables.com/convert/color/rgb-to-hsv.htm

After having the average pixel values of the ball, the next step is to use these values as inputs for a Fuzzy Sugeno system described in Chapter 2. We implement this Fuzzy system as a class which stores the configurations of the rule base in an external text file. The output of the system is a numerical number from 1 to 9 which indicates the color number (remind that we use 9 balls in this work) of the ball.

B.4 Fusion of Cameras and Decision

In this application, we use the SOCKET technique for the communication betweet the NAO's camera and the IP camera. A laptop computer controls the IP camera in the network and acts as the server. The NAO camera sends its Fuzzy result to the server where the fusion between the two cameras is handled. We implemented the fusion system as a class and the Evidence Theory combinations as a library.

After having the final result of the ball's color, the NAO robot reacts to the result: if it is the target ball, it stops the movement:

```
fMotion.stopMove();
fMotion.waitUntilMoveIsFinished();
```

Then it notifies the result to human through its loud-speaker. To use this, we have to declare a proxy to the ALTextToSpeech module:

```
AL::ALTextToSpeechProxy fTextSpeech;
```

And send the phrase to be said:

```
fTextSpeech.say(TEXT_SPEECH_I_FOUND_THE_BALL);
```

In the case that the detected ball is not the target one, the robot continues its search and so on.

Appendix C

Software Implementation in Object Recognition of NAO robot

In this appendix, we present some essential information about the software implementation in the scenario of object recognition for the NAO robot. Figure C.1 shows the general flow when the NAO robot is requested to recognize an object from human. It receives oral command or a signal sent from a PC then it tries to recognize an object frontwards. There are three cameras used in this scenario: two 2D cameras (NAO's and an IP camera) and one depth camera (Axus Xtion Pro) which are already mentioned in Chapter 4 as well as the recognition technique. Each camera captures the scene in front of the robot, then constructs mass vectors based on extracted feature points. After that we use the Evidence Theory to combine the masses and give the final decision about the name of the detected object.

In fact, the voice recognition of the NAO robot, the communication between the cameras are already described in Appendix B, so in this appendix we focus on how to use the 2D and 3D cameras to extract feature points of the objects.

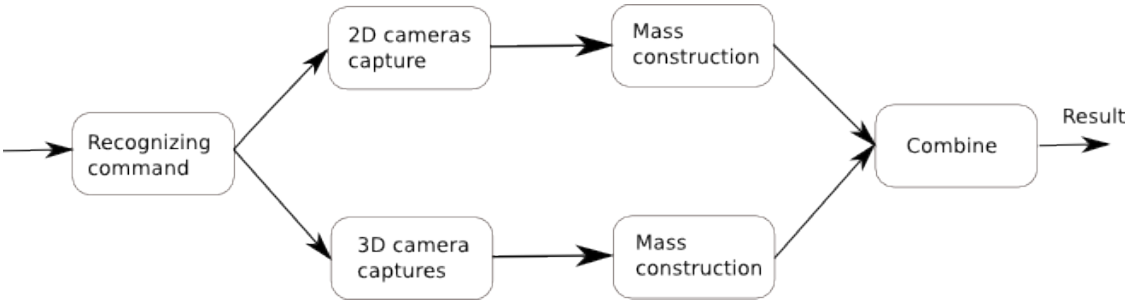

FIGURE C.1: Processing flow of the NAO robot to recognize objects.

C.1 Using 2D Camera

As mentioned in Chapter 4, we use two cameras 2D to extract feature points of the object. The SURF descriptor is applied to obtain interesting points, then these points are described to build a description vector for each one. We present some information about this excellent technique which is taken from [30].

In order to detect interesting points, SURF uses a very basic Hessian matrix approximation, this lends itself to the use of integral images which help reduce computation time. An integral image is precomputed and it allows to calculate the sum of intensities of the original image very fast by considering only 3 simple additions in the integral image. For example if we need to calculate the intensities of the rectangle ABCD of the original image, we need only to calculate: Int(A) - Int(B) - Int(C) + Int(D) (see Fig. C.2). The Hessian matrix is a squared matrix which represents the second order partial derivative of the function. In this case of two variables (x and y coordinates of the image), calculating the $det(matrix)$ can determine the local maximum which is considered as an interesting point. To determine the Hessian matrix of the image which is the result of convolution between a Gaussian filter and the original image, we need to determine the second order partial derivative of the Gaussian function according to x direction, y direction and xy direction (see Fig. C.3). This is done by an integer approximation with a scale sigma = 1.2 (initial scale). After that, to localise interest points in the image and over scales, a non-maximum suppression in a $3 \times 3 \times 3$ neighbourhood is applied.

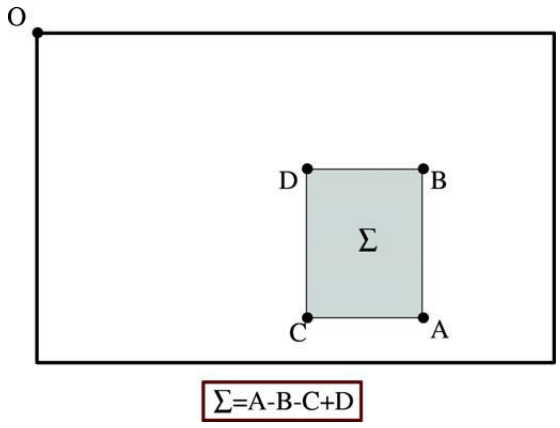

FIGURE C.2: It takes only three additions and four memory accesses to calculate the sum of intensities inside a rectangular region of any size.
Source: [30]

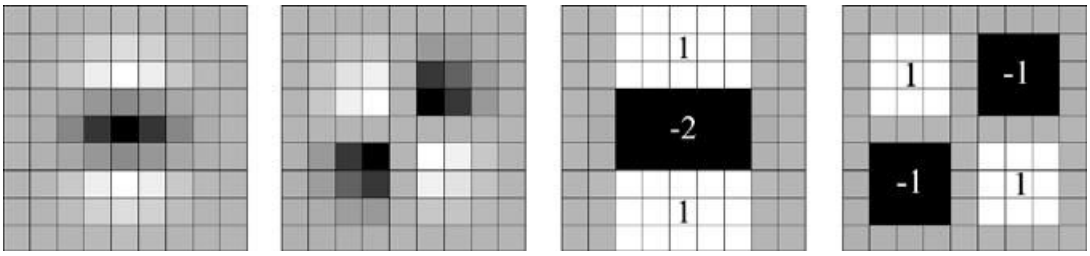

FIGURE C.3: Left to right: The Gaussian second order partial derivative in y- (L_{yy}) and xy-direction (L_{xy}), respectively. The grey region are equal to zero.
[2] Source: [30]

In the next step, interesting points are described. First, we calculate the Haar wavelet reponses in x and y direction within a circular neighborhood of radius $6 \times s$ around the

interest point, with s the scale at which the interest point was detected. We can use integral image in this step to quickly calculate the convolution between the image and the Gaussian approximation created as a Haar wavelet. Then, the vertical (y) and horizontal (x) responses of all points in a sliding window a summed to give a dominant orientation vector (for that sliding window), and we choose the longest vector as the orientation. After that, we construct a square region centered around the interest point and oriented along the orientation selected before. The size of widow is $20 \times s$ (20 times of the scale). The region is split up regularly into smaller $4 \times 4 = 16$ square sub-regions (see Fig. C.4). For each sub-region, we compute Haar wavelet responses. Then, wavelet responses are summed up over each sub-region and form a first set of entries in the feature vector.

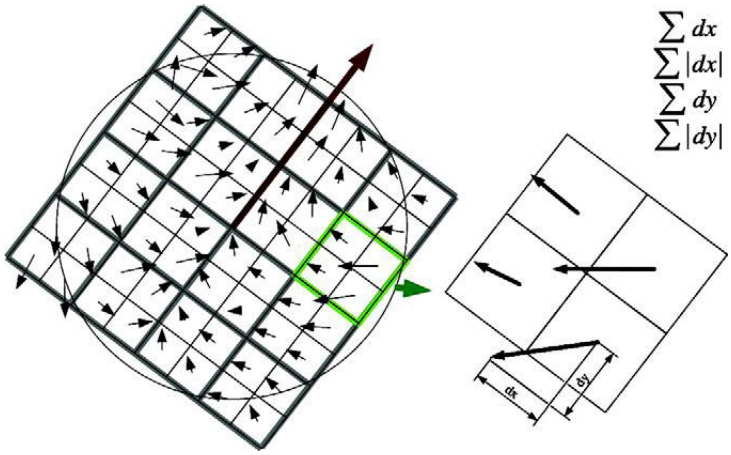

FIGURE C.4: To build the descriptor, an oriented quadratic grid with 4×4 square sub-regions is laid over the interest point (left).
Source: [30]

The SURF descriptor was already implemented in the OpenCV library. The following code declares a detector of interesting points and a descriptor:

```
/// SURF detection of key points.
cv::SurfFeatureDetector detector;

/// SURF extractor for calculating description.
cv::SurfDescriptorExtractor extractor;
```

Then for a captured scene image, we can extract and describe its interesting points. For example we read the scene from an image in gray scale and describe it:

```
// Scene image to recognize model.
cv::Mat scene;

// Load scene image.
scene = cv::imread(sceneFileName, CV_LOAD_IMAGE_GRAYSCALE);

if(!scene.data) {
```

```
        std::cout << "APP-ERROR: Cannot read scene: "
                << sceneFileName << std::endl;
        exit(EXIT_FAILURE);
}

std::vector<cv::KeyPoint> sceneKeypoints;
cv::Mat sceneDescriptors;

// Detect key points for scene.
detector.detect(scene, sceneKeypoints);

if (sceneKeypoints.size() == 0) {
        std::cout << "APP-WARNING: Cannot find any
        keypoints for scene: " << sceneFileName << std::endl;
        return;
}

// Compute descriptors for scene.
extractor.compute(scene, sceneKeypoints, sceneDescriptors);
```

After this step, we can compare the description of the scene with the descriptions of precomputed model images to find the matching. We can use the FLANN library to do that as shown below, with $nbKNN = 2$ indicating that we find the two feature points in the model that best match to each feature point in the scene. The variable *matches* stores the matching information. Based on these information we can construct mass vector as discussed in Chapter 4. Fig. C.5 shows the matching one-by-one of each feature point from a cup captured by a camera and a model image of the cup stored in database.

```
cv::FlannBasedMatcher matcher;

// Find matching between key points of model and scene.
matcher.knnMatch(sceneDescriptors,
                 modelDescriptors[i], matches, nbKNN);
```

C.2 Using 3D Camera

The NAO robot calls another 3D camera named Axus Xtion Pro to capture the depth information of the object. Indeed, the SHOT descriptor ([71]) is used to extract and describe feature points of the object. SHOT stands for Signature of Histograms of Orientations. According to [29], Fig. C.6 shows an overview of the computation steps for each point p in the point cloud. The first three steps are used to compute the local coordinate system at p. The n neighbours p_i of a point p are used to compute a weighted covariance C:

$$C = \frac{1}{n}\sum^{n} r - ||p_i - p||).(p_i - p).(p_i - p)^T \tag{C.1}$$

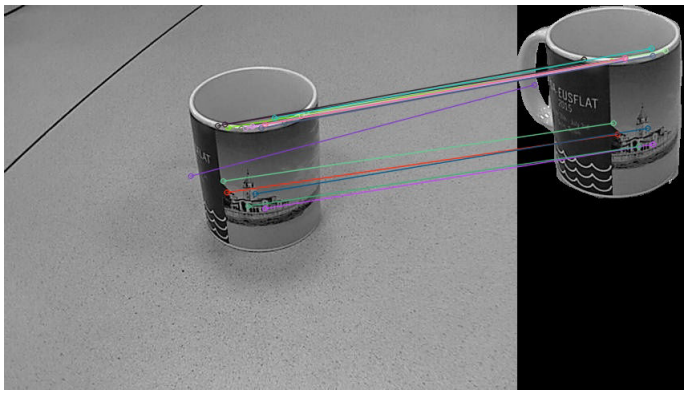

FIGURE C.5: The matching of feature points between a test image and a model image.

where r is the radius of the neighbourhood volume. Then the covariance matrix's eigenvalue decomposition creates three orthogonal eigenvectors composing the local coordinate system at p. This local coordinate system is used to divide the spatial environment of p with an isotropic spherical grid. Then we create histogram for each grid cell, and group cell histograms to each point histogram.

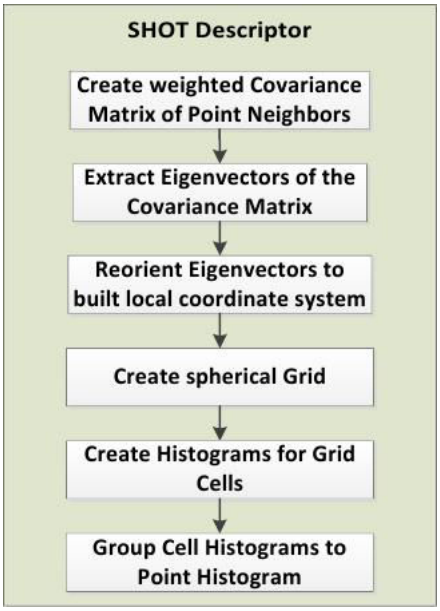

FIGURE C.6: SHOT descriptor computation for one point.
Source: [29]

The SHOT descriptor was implemented in the PCL library ([2]). We firstly have to install the PCL library, then include it in the CMake file:

```
find_package(PCL 1.5 REQUIRED)

include_directories(${PCL_INCLUDE_DIRS} )
```

```
link_directories(${PCL_LIBRARY_DIRS} )

add_definitions(${PCL_DEFINITIONS})
```

Then we should tell the compiler to link this library to the executable file:

```
target_link_libraries(ObjRecogFusion ${PCL_LIBRARIES})
```

Then we can use the following classes provided by the PCL library to extract, describe feature points, and do matching:

```
/// Normal estimator for model and scene.
pcl::NormalEstimationOMP<pcl::PointXYZ, pcl::Normal> normEst;

/// Uniform sapling.
pcl::UniformSampling<pcl::PointXYZ> uniformSampling;

/// SHOT Estimator for description.
pcl::SHOTEstimationOMP<pcl::PointXYZ, pcl::Normal,
                       pcl::SHOT352> descrEst;

/// K-dimension search tree used for matching.
pcl::KdTreeFLANN<pcl::SHOT352> matchSearch;
```

Fig. C.7 shows an example of a 3D image captured by the Axus Xtion Pro camera. After we extract the 3D feature points of the scene and the model object, we can compare and find the matching to construct mass vectors as discussed in Chapter 4.

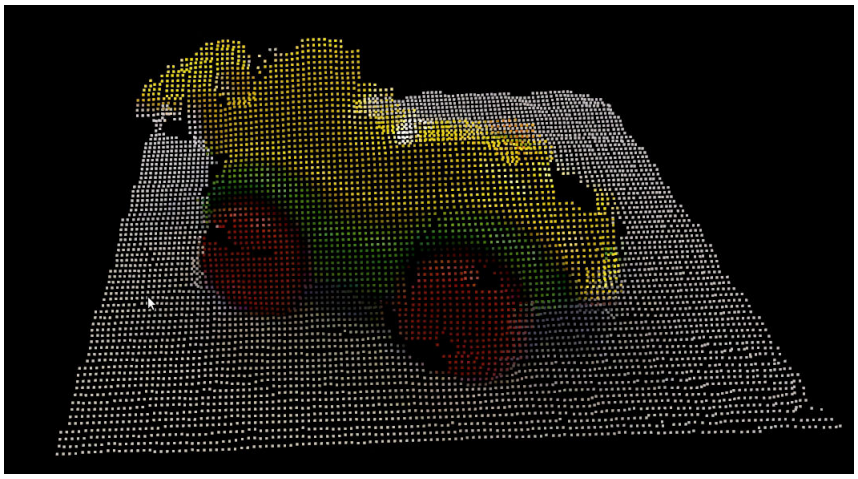

FIGURE C.7: Point cloud captured by the 3D camera.

Bibliography

[1] Alaa E Abdel-Hakim, Aly Farag, et al. "CSIFT: A SIFT descriptor with color invariant characteristics". In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2006, pp. 1978–1983.

[2] Aitor Aldoma et al. "Point cloud library". In: *IEEE Robotics & Automation Magazine* 1070.9932/12 (2012).

[3] S Arivazhagan et al. "Fruit recognition using color and texture features". In: *Journal of Emerging Trends in Computing and Information Sciences* 1.2 (2010), pp. 90–94.

[4] Jens Behley, Volker Steinhage, and Armin B Cremers. "Performance of histogram descriptors for the classification of 3d laser range data in urban environments". In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE. 2012, pp. 4391–4398.

[5] Alexander C Berg, Tamara L Berg, and Jitendra Malik. "Shape matching and object recognition using low distortion correspondences". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 26–33.

[6] Dibya Jyoti Bora, Anil Kumar Gupta, and Fayaz Ahmad Khan. "Comparing the Performance of L* A* B* and HSV Color Spaces with Respect to Color Image Segmentation". In: *arXiv preprint arXiv:1506.01472* (2015).

[7] Hedde HWJ Bosman, Nicolai Petkov, and Marcel F Jonkman. "Comparison of color representations for content-based image retrieval in dermatology". In: *Skin Research and Technology* 16.1 (2010), pp. 109–113.

[8] Brice Burger et al. "Two-handed gesture recognition and fusion with speech to command a robot". In: *Autonomous Robots* 32.2 (2012), pp. 129–147.

[9] Chien-Chern Cheah, Chao Liu, and Jean-Jacques E Slotine. "Adaptive vision based tracking control of robots with uncertainty in depth information". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE. 2007, pp. 2817–2822.

[10] Annett Chilian, Heiko Hirschmüller, and Martin Görner. "Multisensor data fusion for robust pose estimation of a six-legged walking robot". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2011, pp. 2497–2504.

[11] Arnaud Clérentin et al. "Uncertainty and imprecision modeling for the mobile robot localization problem". In: *Autonomous Robots* 24.3 (2008), pp. 267–283.

[12] Adam Coates and Andrew Y Ng. "Multi-camera object detection for robotics". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE. 2010, pp. 412–419.

[13] Ingo Dahm et al. “Robust color classification for robot soccer”. In: *7th International Workshop on RoboCup*. Citeseer. 2003.

[14] Arthur P Dempster. “Upper and lower probabilities induced by a multivalued mapping”. In: *The annals of mathematical statistics* (1967), pp. 325–339.

[15] KS Deshmukh and GN Shinde. “An adaptive color image segmentation”. In: *ELCVIA Electronic Letters on Computer Vision and Image Analysis* 5.4 (2005), pp. 12–23.

[16] Bertram Drost et al. “Model globally, match locally: Efficient and robust 3D object recognition”. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 998–1005.

[17] Didler Dubois and Henri Prade. “Representation and combination of uncertainty with belief functions and possibility measures”. In: *Computational Intelligence* 4.3 (1988), pp. 244–264.

[18] Richard O Duda and Peter E Hart. “Use of the Hough transformation to detect lines and curves in pictures”. In: *Communications of the ACM* 15.1 (1972), pp. 11–15.

[19] Guillermo Enriquez, Sunhong Park, and Shuji Hashimoto. “Wireless sensor network and RFID sensor fusion for mobile robots navigation”. In: *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*. IEEE. 2010, pp. 1752–1756.

[20] Mark D Fairchild. “Color and image appearance models”. In: *Color Appearance Models* (2005), p. 340.

[21] Fang Fang et al. “A new multisensor fusion SLAM approach for mobile robots”. In: *Journal of Control Theory and Applications* 7.4 (2009), pp. 389–394.

[22] Forough Farshidi, Shahin Sirouspour, and Thia Kirubarajan. “Robust sequential view planning for object recognition using multiple cameras”. In: *Image and Vision Computing* 27.8 (2009), pp. 1072–1082.

[23] Gregory T Flitton, Toby P Breckon, and Najla Megherbi Bouallagu. “Object Recognition using 3D SIFT in Complex CT Volumes.” In: *BMVC*. 2010, pp. 1–12.

[24] Mihai Cristian Florea. “Combinaison d’informations hétérogènes dans le cadre unificateur des ensembles aléatoires: approximations et robustesse”. PhD thesis. Université Laval, 2007.

[25] Sylvie Galichet, Reda Boukezzoula, and Laurent Foulloy. “Explicit analytical formulation and exact inversion of decomposable fuzzy systems with singleton consequents”. In: *Fuzzy Sets and Systems* 146.3 (2004), pp. 421–436.

[26] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. “Object categorization using co-occurrence, location and appearance”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[27] Nagham Hamid et al. “A Comparison between using SIFT and SURF for characteristic region based image steganography”. In: *International Journal of Computer Science Issues* 9.33-3 (2012), pp. 110–116.

[28] Xiaowei Han et al. “An approach of color object searching for vision system of soccer robot”. In: *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on*. IEEE. 2004, pp. 535–539.

[29] R Hänsch, T Weber, and O Hellwich. "Comparison of 3D interest point detectors and descriptors for point cloud fusion". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2.3 (2014), p. 57.

[30] Tinne Tuytelaars Luc Van Gool Herbert Bay Andreas Ess. "Speeded-Up Robust Features (SURF)". In: *Computer Vision and Image Understanding* (2008).

[31] Michael B Holte et al. "3d human action recognition for multi-view camera systems". In: *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*. IEEE. 2011, pp. 342–349.

[32] Michael Boelstoft Holte, Thomas B Moeslund, and Preben Fihl. "Fusion of range and intensity information for view invariant gesture recognition". In: *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE. 2008, pp. 1–7.

[33] *International Commission on Illumination. Colorimetry, 2nd Edition.* CIE, 1986.

[34] George H Joblove and Donald Greenberg. "Color spaces for computer graphics". In: *ACM siggraph computer graphics*. Vol. 12. 3. ACM. 1978, pp. 20–25.

[35] Simardeep Kaur and Dr Vijay Kumar Banga. "Content based image retrieval: Survey and comparison between rgb and hsv model". In: *International Journal of Engineering Trends and Technology* 4.4 (2013), pp. 575–579.

[36] Kourosh Khoshelham. "Extending generalized hough transform to detect 3d objects in laser range data". In: *ISPRS Workshop on Laser Scanning and SilviLaser 2007, 12-14 September 2007, Espoo, Finland*. International Society for Photogrammetry and Remote Sensing. 2007.

[37] Youn K Kim, Kyoung W Kim, and Xiaoli Yang. "Real time traffic light recognition system for color vision deficiencies". In: *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*. IEEE. 2007, pp. 76–81.

[38] Jan Knopp et al. "Hough transform and 3D SURF for robust three dimensional classification". In: *Computer Vision–ECCV 2010*. Springer, 2010, pp. 589–602.

[39] Onur Kucuktunc and Daniya Zamalieva. "Fuzzy color histogram-based CBIR system". In: *Proceedings of 1st International Fuzzy Systems, Symposium*. 2009, pp. 231–234.

[40] Christian Lebiere, Florian Jentsch, and Scott Ososky. "Cognitive models of decision making processes for human-robot interaction". In: *Virtual Augmented and Mixed Reality. Designing and Developing Augmented and Virtual Environments*. Springer, 2013, pp. 285–294.

[41] Hyungjik Lee and Seul Jung. "Balancing and navigation control of a mobile inverted pendulum robot using sensor fusion of low cost sensors". In: *Mechatronics* 22.1 (2012), pp. 95–105.

[42] Sang-Geol Lee, Kwang-Baek Kim, and Eui-Young Cha. "Color Inference Using an Enhanced Fuzzy Method". In: *Red (R)* 330 (2013), p. 30.

[43] Tzuu-Hseng S Li et al. "Dynamic balance control for biped robot walking using sensor fusion, Kalman filter, and fuzzy logic". In: *IEEE Transactions on Industrial Electronics* 59.11 (2012), pp. 4394–4408.

[44] Haibin Ling and David W Jacobs. “Shape classification using the inner-distance”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29.2 (2007), pp. 286–299.

[45] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[46] David G Lowe. “Object recognition from local scale-invariant features”. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[47] Ruben Martinez-Cantin et al. “Active Policy Learning for Robot Planning and Exploration under Uncertainty.” In: *Robotics: Science and Systems*. 2007, pp. 321–328.

[48] Z May and MH Amaran. “Automated oil palm fruit grading system using artificial intelligence”. In: *Int. J. Eng. Sci* 11.3035.21 (2011).

[49] Hunny Mehrotra, Banshidhar Majhi, and Phalguni Gupta. “Annular iris recognition using SURF”. In: *Pattern Recognition and Machine Intelligence*. Springer, 2009, pp. 464–469.

[50] Malik Arman Morshidi, Mohammad Hamiruce Marhaban, and Adznan Jantan. “Color segmentation using multi layer neural network and the HSV color space”. In: *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on*. IEEE. 2008, pp. 1335–1339.

[51] Rafael Munoz-Salinas et al. “Multi-camera people tracking using evidential filters”. In: *International Journal of Approximate Reasoning* 50.5 (2009), pp. 732–749.

[52] K Murphy and W Freeman. “Contextual Models for Object Detection using Boosted Random Fields”. In: NIPS. 2004.

[53] Gabriel Nützi et al. “Fusion of IMU and vision for absolute scale estimation in monocular SLAM”. In: *Journal of intelligent & robotic systems* 61.1-4 (2011), pp. 287–299.

[54] Ahmad Osman, Valérie Kaftandjian, and Ulf Hassler. “Improvement of X-ray castings inspection reliability by using Dempster–Shafer data fusion theory”. In: *Pattern Recognition Letters* 32.2 (2011), pp. 168–180.

[55] Abraham Otero et al. “Fuzzy constraint satisfaction approach for landmark recognition in mobile robotics”. In: *AI communications* 19.3 (2006), pp. 275–289.

[56] Chavdar Papazov and Darius Burschka. “An efficient ransac for 3d object recognition in noisy and occluded scenes”. In: *Computer Vision–ACCV 2010*. Springer, 2011, pp. 135–148.

[57] Petra Perner. “Cognitive Aspects of Object Recognition–Recognition of Objects by Texture”. In: *Procedia Computer Science* 60 (2015), pp. 391–402.

[58] Reyes Rios-Cabrera, Tinne Tuytelaars, and Luc Van Gool. “Efficient multi-camera vehicle detection, tracking, and identification in a tunnel surveillance application”. In: *Computer Vision and Image Understanding* 116.6 (2012), pp. 742–753.

[59] Emanuele Rodolà et al. “A scale independent selection process for 3d object recognition in cluttered scenes”. In: *International Journal of Computer Vision* 102.1-3 (2013), pp. 129–145.

[60] Boris Ruf, Effrosyni Kokiopoulou, and Marcin Detyniecki. “Mobile museum guide based on fast SIFT recognition”. In: *Adaptive Multimedia Retrieval. Identifying, Summarizing, and Recommending Image and Music*. Springer, 2010, pp. 170–183.

[61] Aswin C Sankaranarayanan, Ashok Veeraraghavan, and Rama Chellappa. “Object detection, tracking and recognition for multiple smart cameras”. In: *Proceedings of the IEEE* 96.10 (2008), pp. 1606–1624.

[62] Kari Sentz and Scott Ferson. *Combination of evidence in Dempster-Shafer theory*. Vol. 4015. Citeseer, 2002.

[63] Pierre Sermanet et al. “Mapping and planning under uncertainty in mobile robots with long-range perception”. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2008, pp. 2525–2530.

[64] Glenn Shafer et al. *A mathematical theory of evidence*. Vol. 1. Princeton university press Princeton, 1976.

[65] Philippe Smets. “Constructing the Pignistic Probability Function in a Context of Uncertainty.” In: *UAI*. Vol. 89. 1989, pp. 29–40.

[66] Da-Lei Song et al. “Illumination invariant color model selection based on genetic algorithm in robot soccer”. In: *The 2nd International Conference on Information Science and Engineering*. IEEE. 2010, pp. 1245–1248.

[67] Akira Suga et al. “Object recognition and segmentation using SIFT and Graph Cuts”. In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE. 2008, pp. 1–4.

[68] Wandi Susanto, Marcus Rohrbach, and Bernt Schiele. “3D object detection with multiple kinects”. In: *European Conference on Computer Vision*. Springer. 2012, pp. 93–102.

[69] Jean-Christophe Terrillon and Shigeru Akamatsu. “Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images”. In: *Vision Interface*. Vol. 99. Citeseer. 1999, p. 1821.

[70] Marko Tkalcic, Jurij F Tasic, et al. “Colour spaces: perceptual, historical and applicational background”. In: *Eurocon*. 2003.

[71] Federico Tombari, Samuele Salti, and Luigi Di Stefano. “Unique signatures of histograms for local surface description”. In: *Computer Vision–ECCV 2010*. Springer, 2010, pp. 356–369.

[72] Federico Tombari and Luigi Di Stefano. “Hough voting for 3d object recognition under occlusion and clutter”. In: *IPSJ Transactions on Computer Vision and Applications* 4.0 (2012), pp. 20–29.

[73] Yizhou Wang, Mike Brookes, and Pier Luigi Dragotti. “Object recognition using multi-view imaging”. In: *2008 9th International Conference on Signal Processing*. IEEE. 2008, pp. 810–813.

[74] Lior Wolf and Stanley Bileschi. “A critical view of context”. In: *International Journal of Computer Vision* 69.2 (2006), pp. 251–261.

[75] Ronald R Yager. “On the Dempster-Shafer framework and new combination rules”. In: *Information sciences* 41.2 (1987), pp. 93–137.

[76] Eiichi Yoshida et al. "'Give me the purple ball'-he said to HRP-2 N. 14". In: *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*. IEEE. 2007, pp. 89–95.

[77] Yu Zhong. "Intrinsic shape signatures: A shape descriptor for 3D object recognition". In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 689–696.

[78] Chun Zhu and Weihua Sheng. "Human daily activity recognition in robot-assisted living using multi-sensor fusion". In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE. 2009, pp. 2154–2159.

[79] Lalla Meriem Zouhal and Thierry Denoeux. "An evidence-theoretic k-NN rule with parameter optimization". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 28.2 (1998), pp. 263–271.