University of Padua

Doctoral Thesis

Object Localization and Recognition for Mobile Robots with Online Learning based on Mixture of Projected Gaussian

Author:
Mauro Antonello

Supervisor:
Prof. Emanuele Menegatti

Intelligent Autonomous Systems Laboratory
Information Engineering

February 2015

In 1966, Marvin Minsky at MIT asked his undergraduate student Gerald Jay Sussman to "spend the summer linking a camera to a computer and getting the computer to describe what it saw". We now know that the problem is slightly more difficult than that.

Richard Szeliski
Computer Vision: Algorithms and Applications

UNIVERSITY OF PADUA

Abstract

Information Engineering

Doctor of Information Engineering

Object Localization and Recognition for Mobile Robots with Online Learning based on Mixture of Projected Gaussian

by Mauro Antonello

One of the primary capabilities required by autonomous robots is recognizing the surrounding environment with high responsiveness, often combined with object recognition and grasping tasks. Moreover robots acting in mutable scenarios are also required to be capable of learning new object models online. Along with peculiar requirements the robotics offers to the object recognition task some unique advantages, such the robot capability to move in the environment. Moreover, usually an autonomous robot can relax the recognition precision obtained at the beginning of its exploration and favour the speed at which this results are obtained. The aim of the work presented in this thesis is to explore a new object recognition method able to exploit this advantages in order to fulfil the features required by autonomous robotics.

In order enhance pose estimation the proposed algorithm prioritize the keeping of the geometrical information from the objects shape and texture. Since the object models also need to be as much lightweight as possible this algorithm relies on local 6 DoF features extraction to describe the object appearance without load the final model of unnecessary information. Once the 6 DoF keypoints are obtained, the proposed method makes the use specifically designed probability distribution, namely the the Mixture of Projected Gaussian (MoPG) in order to learn their spatial distribution. A Bag of Words (BoW) technique has been introduced after the feature detection in order make feature descriptors more invariant to small appearance changes, due to light conditions or perspective distortions.

The choice of using the MoPG distribution lies in one algebraic property of the Gaussian function, namely its closure over the convolution operator. In this thesis this property is exploited in order to obtain a closed form formula for calculating the cross-correlation of MoPG. The recognition algorithm makes use of the cross-correlation between MoPG in order to both identify and localize objects in the scene.

The recognition and localization performances of the proposed technique was validated on two different publicly available datasets, namely the RGB-D Dataset and the BigBIRD Dataset. An analysis of both category and instance recognition results is presented and the emerged advantages or the issues of the proposed technique are discussed. The localization error ($\cos(\Delta_R) = 2°$) and the instance recognition rate (91%) resulted being aligned of the state of art thus justifying a further exploration of the proposed method.

The topics presented in this thesis was further explored in some related works. In particular a collaboration with the Intelligent Systems Research Institute (Sungkyunkwan University, Republic of Corea) led an adapted version of the proposed method that has been successfully integrated in an autonomous domestic robot.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the primary capabilities required by autonomous robots is to recognize the surrounding environment with high responsiveness, often along with object recognition and grasping tasks. Moreover robots acting in mutable scenarios are also required to be capable of learning new object models online. Contrary to the simplicity with which humans deal with this task the scientific research has tried to solve this problem for many years and is still far from achieving solutions that are precise and general enough to be useful for autonomous robotics.

Along with some peculiar requirements robotics offers some unique advantages to the object recognition task, such as the robot capability to move in the environment or the high number of different sensors usually present in autonomous robots. In addition usually an autonomous robot can lower the recognition precision obtained at the beginning of its exploration and favor the speed at which this results are obtained. The aim of this thesis is to explore a new object recognition method that is able to exploit these advantages in order to fulfill the requirements of autonomous robotics.

The object recognition task has been studied widely in Computer Vision but, although several fast and robust algorithms are able to detect the presence of object instances, the majority of these cannot precisely locate these instances in the analyzed scene. Indeed a robot's need to being able to grasp object requires that the detected objects are being precisely located with a 6 Degrees of Freedom (DOF) pose.

In order to enhance pose estimation the proposed algorithm prioritizes the keeping of geometrical information from the objects shape and texture. Since the object models also need to be as lightweight as possible this algorithm relies on local feature extraction to describe the object appearance without loading the final model with unnecessary

information. Since robots deals with 3D objects, the extracted features need to be provided with a 6 DoF pose, with respect to a predefined reference inside the object. Once the position and orientation of these local features is obtained, the proposed method makes use of a novel variant of the Mixture of Gaussian in order to learn their spatial distribution. One of the main issues addressed in the learning process of the poses Mixture of Gaussian (MoG) distribution comes from the need to adapt the EM algorithm to 6 DoF pose points. Indeed, this points lies in the 3rd order Special Euclidean $SE(3)$ group which is a manifold rather than an Euclidean space. The classical EM algorithm [5] requires operations (i.e. taking the mean of a point set) that may be undetermined in the $SE(3)$ space. To overcome this issue, the proposed method makes use of the Mixture of Projected Gaussian (MoPG) distribution [6]: in this distribution the components of the mixture are learned in the 6D tangent space of the $SE(3)$, thus allowing to obtain Gaussian components in a similar way to the classical approach.

Keeping a probability distribution of the feature poses allows the proposed method to be more robust to the high detection error that usually burdens robots RGB-D sensors. Moreover keeping in the models such probability distribution requires far less data than keeping the full set of poses from which it is learned. This aspect becomes particularly important in case the robot needs to learn or integrate its object models with new information, obtained while it moves and sees the objects from previously unseen viewpoints.

The choice of using the MoPG distribution lies in one algebraic property of the Gaussian function, namely its closure to the convolution operator. In the recognition process the proposed algorithm exploits the strict relation between the convolution and the cross-correlation in order to compare two models. The MoPG Probability Density Function (PDF) is a linear combination of Gaussian functions, thus obtaining the cross-correlation of two MoPG PDF is a fast operation and its result is another MoPG. Given the cross-correlation $(f \star g)(x)$ of two signals $f(x)$ and $g(x)$, the value $(f \star g)(\hat{x})$ at a given point $\hat{x}$ expresses the overlap of $f(x)$ and $g(x - \hat{x})$. The proposed algorithm exploits this property in order to find the optimal registration between the two models: once the cross-correlation MoPG is obtained, a fast mode-finding algorithm [4] is used to find the peaks in the cross-correlation; the $SE(3)$ point of these peaks represents the guessed instance poses and their PDF values are proportional to registration overlapping. One of the great advantages of using the proposed cross-correlation based technique is that the recognition and localization tasks are performed at the same time and have low computational requirements.

This novel algorithm is further analyzed and some variants are proposed in order to make it even more robust to environment and sensor noise. A Bag of Words (BoW) technique

[7] is introduced in the feature detection phase in order make feature descriptors more invariant to small appearance changes and aid the object classification. In collaboration with the Intelligent Systems Research Institute (Sungkyunkwan University, Republic of Corea) an adapted version of the proposed method has been successfully integrated in an autonomous domestic robot [8].

Chapter 2

Acquisition

Raw data acquisition may seem a trivial task in which effort is mostly focused on collecting and organizing as much information as possible from different sources. This idea may be correct in many recognition tasks but in the autonomous robotics the processing of raw data is a key task: the multiplicity of sensors present in most robots produce a huge data flow that must be efficiently handled. In our scenario the data acquisition is an open loop and the raw data must be processed in a real-time context. To this purpose particular attention is given to the efficiency of the data flow processing in order to maintain real-time performances despite the limited robot computational power. The information contained in the frames of the data stream usually presents high redundancy that should be eliminated in order to reduce computational requirements of the data analysis. To this end the detection of the visual correspondences among consecutive frames is a great aid in the information extraction process. Similarly, knowing the point of view from which each frame is shot allows for an efficient integration of the data collected by the robot during its motion.

The information extraction and the viewpoint detection are two core requirements for the recognition algorithm presented in this thesis. In the following sections a more detailed analysis of the requirements is presented along with the issues and the solutions inherent to the acquisition process.

2.1 Requirements Analysis

In the last years, the great interest of the academic research in autonomous robotics has lead to the maturation of some of its core design patterns. By assuming the availability of some of these common patterns the algorithm presented in this thesis can exploit the associated advantages without any loss of generality.

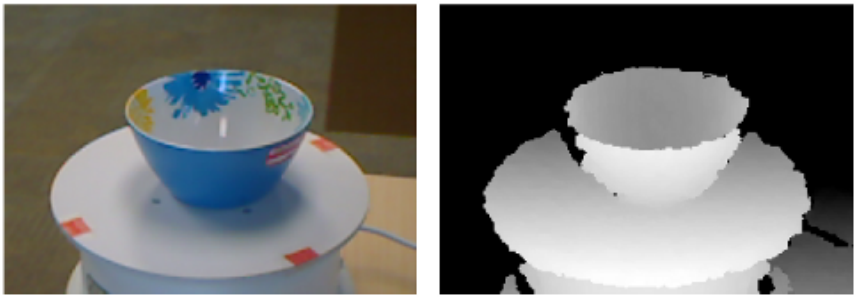

FIGURE 2.1: An RGB-D frame from the RGB-D Dataset [1]. Every frame consists of a RGB image and a Depth image containing information about the distance of the observed surfaces. An RGB-D frame can be converted to an organized point cloud by assigning the 3D coordinates of the observed point to each pixel in the RGB image.

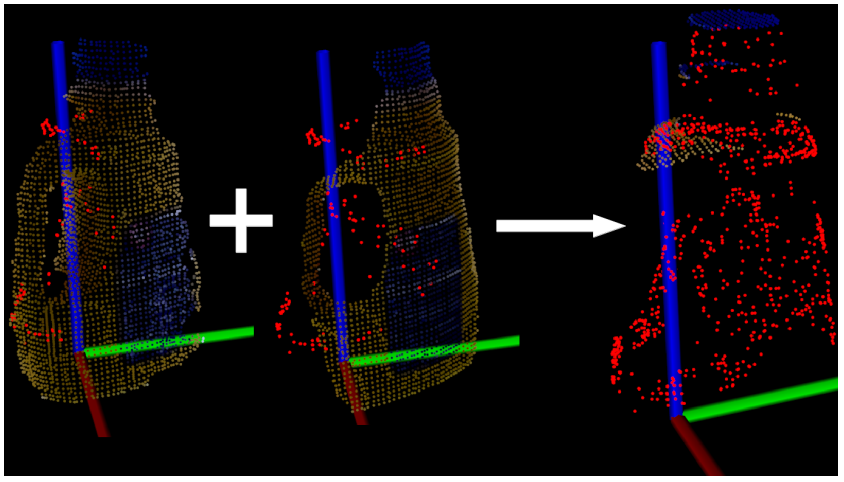

FIGURE 2.2: The transformation between an object and the rgbd sensor reference frames, namely the viewpoint, is required in order to incrementally integrate the point cloud of each frame. The result is a cloud containing keypoints observed by all views.

The first assumption is represented by the presence of an RGB-D sensor. Exploiting the depth information associated to an RGB frame is extremely useful for the object recognition task so that most state of art algorithms [9, 10] already have the same requirement. The availability of an RGB-D sensor represents a very weak constraint since the cost reduction and the quality improvement of these devices has made them very common in autonomous robotics [11, 12].

The second assumption comes from one of the most distinctive features of the autonomous robotics: the capability to move in the environment and, often, manipulate objects too. The mobility allow such robots to observe a scene or an object from many different viewpoints thus overcoming occlusions or other visual artifacts that may affect some viewpoints. The second requirement is the knowledge of the viewpoint from which each frame is shot.

A more detailed overview of these two requirements is presented in the next sections.

FIGURE 2.3: Bumblebee stereo camera. The depth of observed surfaces is triangulated from the slight visual differences in the two camera frames. The depth image quality of passive stereo cameras is inferior with respect to cameras that exploit active structured light projection but unlike these can work outdoor without issues.

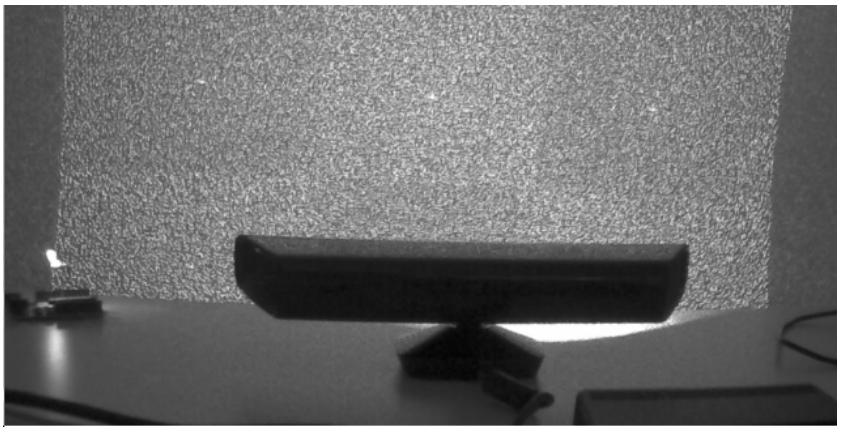

FIGURE 2.4: Kinect sensor. This infra-red image shows the invisible pattern projected by the kinect sensor. From the deformation of the projected pattern an RGB-D sensor can infer the depth of the observed surfaces.

2.1.1 RGB-D Devices

By RGB-D device we mean all sensors able to produce a video stream in which each frame provides both an RGB image I_{RGB} and the associated depth image I_D. Each pixel $I_{RGB}(u, v) = [i_r\, i_g\, i_b]^T$ contains information about the color intensities and the correspondent pixel in the depth image $I_D(u, v) = d$ provides the distance of the observed point from the RGB-D sensor (see figure 2.1). There are actually several families of sensors that fulfill this specific but the most common are based on stereo vision, often aided by structured light projection.

The stereo vision is a wide class of 3D reconstruction techniques whose underlying concept is retrieving the distance of an observed area by the slight difference in its projection over two different camera frames. The original approach only assumes two RGB sensors (see figure 2.3) that shot the same area from a known and slightly different point of view. While very simple and low cost, this technique does not produce very accurate results and the reconstruction is usually computationally expensive for mobile robots.

In order to increase the reconstruction quality it is possible to exploit the projection of a known structured light pattern in the observed scene. Usually this kind of devices are composed of a structured infra-red light projector along with two sensors, one is a normal RGB camera and the second is an infra-red camera. The depth of the RGB frames taken from the first camera is retrieved by observing the deformation of the projected pattern over the surfaces it hits, see figure 2.4. The most commonly used RGB-Sensors (e.g. Microsoft Kinect) are based on this technique and are able to provide good quality results at high frame rates. The downside of this technique is its inability to work on areas hit by direct sunlight or with highly reflective surfaces.

Once RGB and depth images are obtained, it is common to convert RGB-D information into an organized point cloud C. Through the intrinsic sensor parameters each pixel in the depth image $I_D(u, v) = d$ is projected to the 3D point of the corresponding pixel $C(u, v) = [x\, y\, z]^T$ of the organized point cloud.

2.1.2 Viewpoint

In order to incrementally improve the recognition results the presented algorithm aims to integrate RGB-D data acquired from different viewpoints while the robot moves or, alternatively, rotates an object in its manipulation arm. Since the coordinates of the point cloud obtained through the RGB-D sensor are referred to the sensor reference frame an additional technique is needed in order know the spatial transformations between each different viewpoint.

Thanks to the high frame rates of the RGB-D sensors, the visual changes between two temporally close frames are usually small. This allows us to simplify the given problem to finding the spatial transformation that better aligns two similar point clouds. This latter problem is a widely studied topic in 3D vision literature [12–14] and hereafter are presented the techniques that better fit as a solution to our needs. Since the best solution for this problem is still open topic, the strengths and the weakness with respect to an autonomous robotic scenario are also presented with each of the described techniques.

Iterative Closest Point Iterative Closest Point (ICP) [15] is a general purpose technique whose purpose is find the optimal registration between two point clouds. The underlying concept of the many variants of this algorithm is an iterative search for the optimal rotation and translation that minimizes the mean squared distance between the clouds (see figure 2.5).

One possible application for ICP in our scenario is finding the transformation occurred to the sensor pose during the robot motion. This objective can be achieved by using

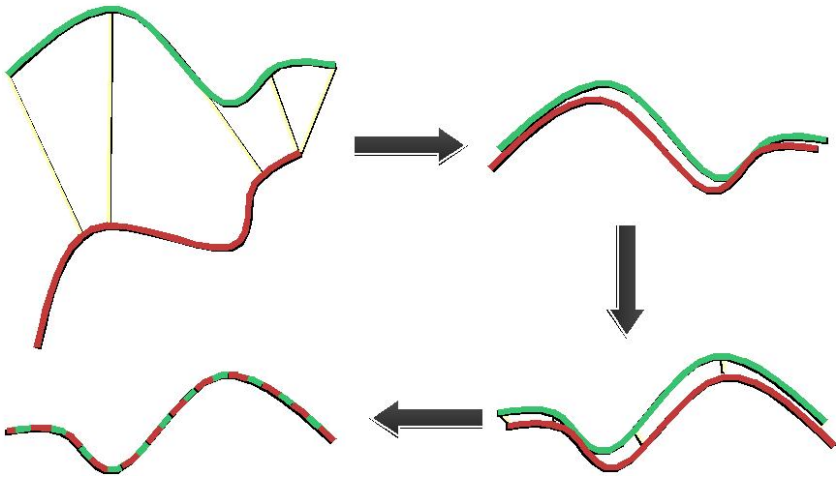

FIGURE 2.5: The Iterative Closest Point (ICP) algorithm is used to find the transformation that best registers two point clouds. Closest points are iteratively associated in order to find the best transformation that minimizes distance between the clouds. This technique can be exploited to find the transformation occurred to the RGB-D sensor (or to the object of interest) between two RGB-D frames.

ICP to find the optimal registrations between a small set of timely close RGB-D frames: the RGB-D sensor pose is iteratively updated by aligning each new frame with its predecessors. Although there exists several optimization specifically designed to this end this process remains computationally expensive. By now most state of art techniques [13, 16] exploit GPU computing in order to maintain good frame-rates.

In order to reduce the computational burden ICP can also be used to align a smaller portion of the original RGB-D frames, focusing the effort on the part that the robot should learn or recognize [17]. This case fits well with the scenario in which the robot is analyzing an object being held in the manipulation arm. In this case the RGB-D sensor is still and only ICP can be applied to the portions of the RGB-D frames that refer to the object being rotated by the robot arm. Unfortunately this scenario is not very effective in practice due to its bad performances on small or symmetrical point clouds. In the experiments conducted using the RGB-D Dataset (see section 2.1.3) a great part of the commonly used household objects are too small or noisy if seen from medium distance and ICP fails in the alignment of their point clouds.

Odometry Approaches based on visual or geometrical appearance of the object suffer from some intrinsic difficulties connected to the alignment of consecutive frames. In particular great issues usually come in the registration of clouds relative to highly symmetrical objects or objects with large reflective parts.

An easy way to overcome these issues is to exploit other sources of information such as the robot odometry or its joint positions. Most autonomous robots are capable of

FIGURE 2.6: The encoder on robot joint motors can be exploited to retrieve the pose of the hand in respect to the RGB-D sensor.

estimating their position relative to a global reference frame in various ways, from motors encoders to inertial measurement units. Since the robot geometry is usually well known, once the robot position is known, the RGB-D sensor position and orientation can be roughly guessed. While this approach is usually less precise than ICP it can be used to obtain an initial estimate for ICP or to maintain reasonable results during periods in which ICP fails for any reason.

In case the robot is equipped with a manipulation arm the robot encoders on its joints can be exploited to obtain the hand pose in respect to the RGB-D sensor (see figure 2.6). This approach is usually more precise and reliable than ICP in case the robot is learning or recognizing an object in its hand. A similar approach can also be used to train object models offline: by using a turntable an object can be observed by an RGB-D sensor from various known point of views in order to create the object model and then provide it to a robot.

2.1.3 Datasets

In order to maintain a higher level of generality and result significance the experiments conducted to test the algorithms presented in this thesis (see chapter 6) used two publicly available RGB-D datasets. Although there exist several datasets for RGB-D object recognition only a few of them provide informations about the viewpoints from which

FIGURE 2.7: The RGB-D Dataset [1] is a large dataset of 300 common household objects. The objects are organized into 51 categories arranged using WordNet hypernym-hyponym relationships (similar to ImageNet).

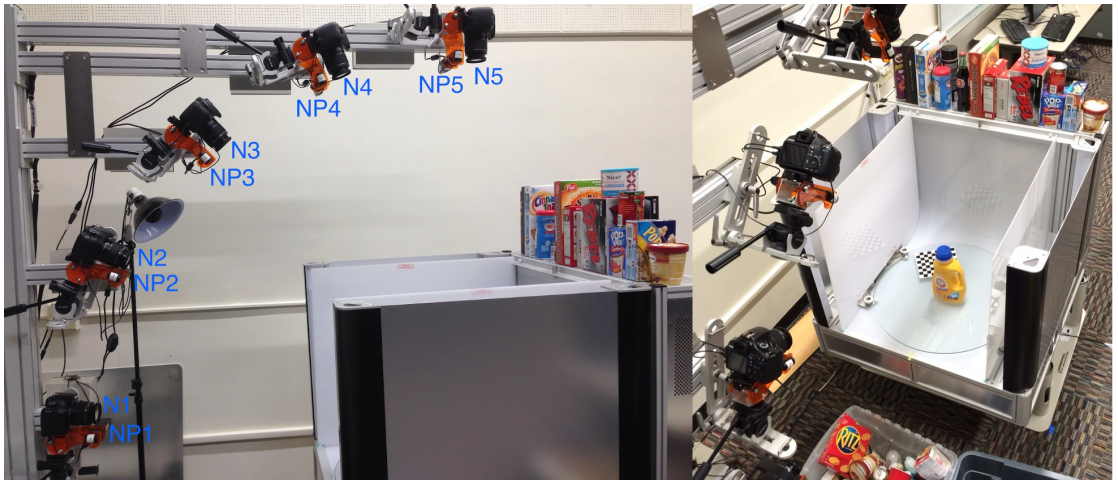

FIGURE 2.8: The BigBIRD dataset [2] offers a very high quality set of RGB-D frames for 100 common objects. For each object they provide 600 3D point clouds and 600 high-resolution (12 MP) images spanning all views.

object images have been taken. Among these the two dataset used in this thesis are presented below.

RGB-D Dataset The RGB-D Dataset [1] (see figure 2.7) is a large dataset of 300 common household objects. The objects are organized into 51 categories arranged using WordNet hypernym-hyponym relationships (similar to ImageNet). This dataset was recorded using a Kinect style 3D camera that records synchronized and aligned 640x480 RGB and depth images at 30 Hz. Each object was placed on a turntable and video sequences were captured for one whole rotation. For each object, there are 3 video sequences, each recorded with the camera mounted at a different height so that the object is viewed from different angles with the horizon. Although very complete, this

dataset is quite challenging: the objects region of interest in the RGB-D frames is very small and several object instances (i.e: vegetables) look very similar to each other. A further difficulty is represented by the low accuracy in the turntable rotation angles and the absence of the pose of the Kinect in respect to the turntable reference frame. The lack of full 6 DoF viewpoints has been handled by exploiting the ICP algorithm presented in section 2.1.2.

BigBIRD The BigBIRD dataset [2] offers a very high quality set of RGB-D frames for 100 common objects (and growing). For each object they provide 600 3D point clouds and 600 high-resolution (12 MP) images spanning all views. The acquisition system (see figure 2.8) exploits a novel method for jointly calibrating a multi-camera system in order to provide an accurate pose for all RGB-D frames. For our purpose the BigBIRD dataset fits better than the RGB-D Dataset but offers much less comparative recognition results due to its recent publication. The point clouds provided by this dataset are not organized preventing the detection and the exploitation of 2D features like SIFT (see section 3.1).

Chapter 3

Description

One of the main issues when dealing with object recognition is how to separate useful information from the raw source data. Images contain a huge amount of data in respect to the portion that could really be informative to the recognition task and considering unnecessary data usually increase the computational effort and decrease the recognition rate. This phenomenon is even more marked when dealing with the frames provided by a robot RGB-D sensor, especially considering the limited computational power of mobile robots.

The description process tries to extract as much useful information as possible from the whole data flow. In our case each RGB-D frame is individually processed and the output of the description step is a set of low level features that are then used in higher levels of the recognition flow to describe the analyzed frame. The next sections briefly presents the adopted description techniques.

3.1 Features

The feature detection is the process performed in order to locate and extract useful information from images. The concept of how useful an information is depends on the specific use case but some simple visual patterns, such as curves or planes, are able to synthesize the content of an image (see figure 3.1) while being very general purpose. The features collected in the description process belong to this set and are referred as low-level features; the aim is providing numerical values for characteristics that could be used to describe every object well. The value assigned to these features is referred as descriptor $\mathbf{d} \in \mathbb{R}^n$; each object will be lately described as a set $\{(F, \mathbf{d})_i\}$ of these feature-descriptor couples.

FIGURE 3.1: The contour extraction is a simple yet effective feature extraction technique. While most of the color data is discarded during the process the remaining contours still contains most of the information useful for the object recognition.

Different features try to describe different aspects but there are some general characteristics that every feature aims to meet. First of all, features that refers to visually similar areas should be retrieved with similar descriptor values. We will refer to this characteristic as robustness of the descriptor with respect to various sources of noise or alterations that may affect the feature.

A first classification of features is the distinction between local or global features. Local features describe only a small portion of the whole image and are provided with an associated keypoint, namely the spatial coordinates of the described point. Global features instead refer to a characteristic of the whole object such as its height.

The proposed algorithm is designed for the use of local features. In addition, since the presented method deals with 3D point clouds, the keypoints associated to these features are composed by a 3D location and a 3D orientation. Thus each detected feature F_i will be associated to a full 6 DoF reference frame $\mathbf{k}_i = [R|\mathbf{t}]$.

Even if the proposed algorithm does not require specific features the two that performed best during our test will be described in more detail in the next paragraphs.

SIFT In the proposed scenario the RGB-D frames provided by the robot sensor are composed by a an RGB image and a depth image thus allowing the use of 2D features, detected on the RGB image. Among other 2D features, SIFT [3] are one of the most diffused tanks to its descriptor robustness: SIFT are invariant to rotations on the image plane, are very robust to light changes and are robust to small geometrical distortions in the detection area; through pyramidal detection a partial scale invariance can be achieved too.

FIGURE 3.2: Global features (left) describe a property of the whole object like its width or its height. Local features (right) describe the appearance of a small part of the image and are associated with a keypoint, namely the position and the shape of the described area.

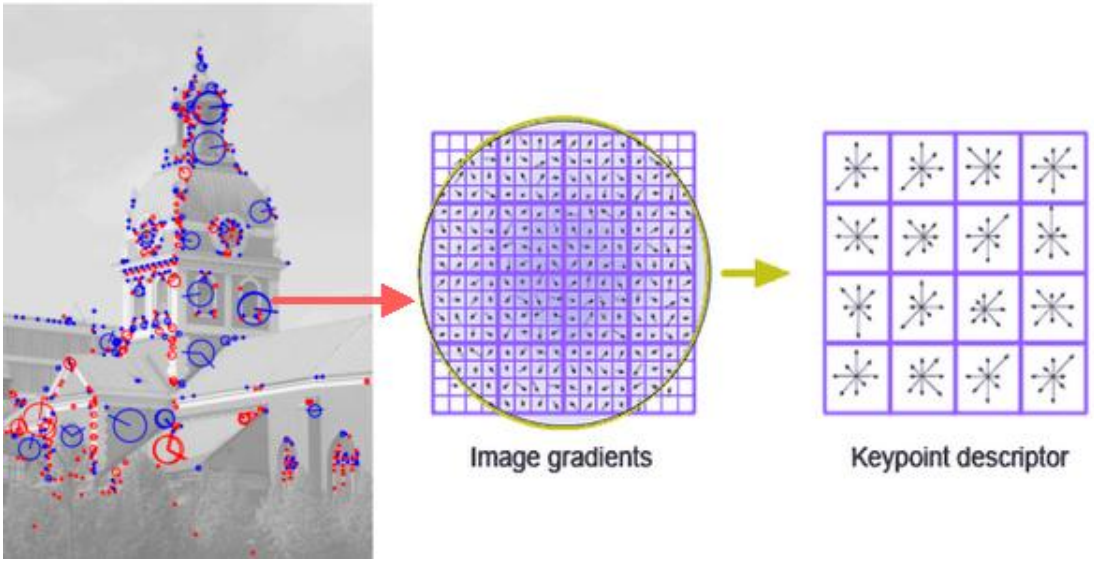

FIGURE 3.3: SIFT [3] are local 2D features whose descriptor is given by the histograms of the color gradients of the described image area. SIFT keypoints includes the detection location (u, v), the dimension of the described area and, since SIFT are invariant to rotation, the angular orientation α corresponding to the principal gradient direction.

The detection of most 2D features, SIFT included, takes much less computational effort than the 3D alternatives but requires an extension of their 2D keypoints to 6 DoF reference frames. The SIFT 2D keypoint

$$\mathbf{k} = \{(u, v), \alpha\}$$

is composed by a 2D point (u, v) in image coordinates and a rotation α on the image plane. Most SIFT detection algorithms usually perform the feature detection process on several image scales thus varying the actual feature sizes; in the use case presented in this thesis this behaviour is undesired since it complicates the back-projection process hereafter described, for this reason the multi scale detection has been disabled during the presented experiments. Exploiting the organized point cloud C associated to the RGB-D frame a SIFT 2D keypoint can easily be back-projected to a 6 DoF keypoint. The translation part $\mathbf{t} = [t_x\, t_y\, t_z]^T$ of the keypoint reference frame is retrieved from the organized point cloud

$$\mathbf{t} = C(u, v).$$

The rotation can be obtained from the cross product of two orthogonal vectors; for the SIFT case the feature orientation α can be used to obtain a first vector $\mathbf{n}_\alpha$, which lies on the SIFT patch plane P, and the second can be the normal $\mathbf{n}_P$ to this plane, as shown in figure 3.4

$$R = \begin{bmatrix} \mathbf{n}_\alpha \\ \mathbf{n}_P \\ \mathbf{n}_\alpha \times \mathbf{n}_P \end{bmatrix}.$$

ISS 3D Despite their worst performances, in respect to the 2D alternatives, 3D features can provide some peculiar advantages. The most relevant for our application is that most 3D features can work with unorganized point clouds, namely the point clouds where points have lost the information about their projection coordinates $I_D(u, v)$ on the depth Image. In our work this property is particularly useful since many dataset do not provide organized point clouds. Moreover 3D features are usually oriented in the description of a local 3D shape making them more robust to the lack of texture in the described area.

In the presented work the ISS 3D keypoints [18] has been adopted due to their good performance and low computational effort.

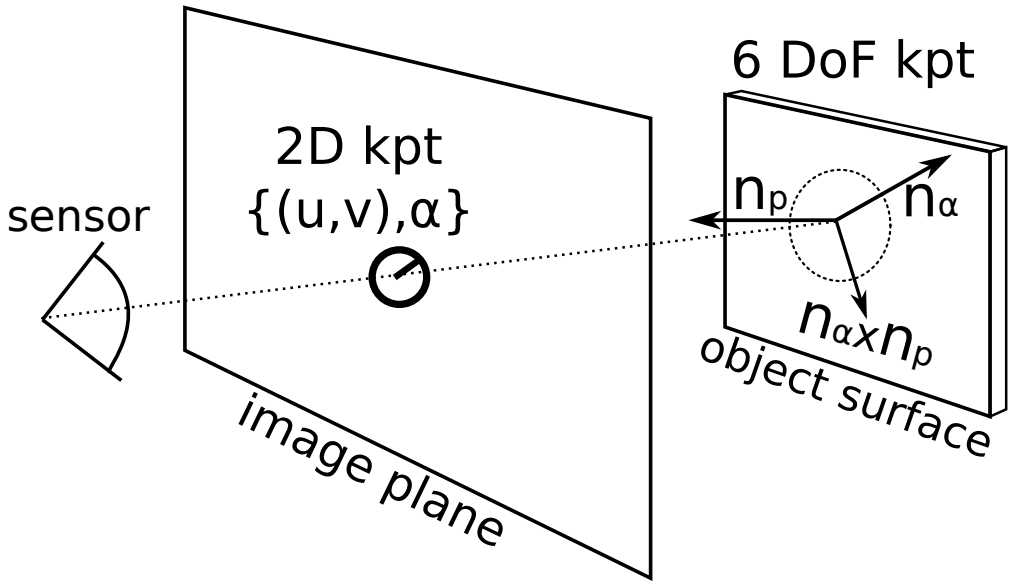

FIGURE 3.4: After its detection a 2D feature is back-projected from its 2D keypoint $\{(u, v), \alpha\}$ in image coordinates to a 6 DoF keypoint in the sensor reference frame. The organized point cloud is exploited to retrieve the 3D translation coordinates of the point (u, v); the normal to the projection surface $\mathbf{n}_P$ and the gradient direction $\mathbf{n}_\alpha$ are used to obtain the 6 DoF keypoint orientation.

3.2 Bag of Visual Words

The Bag of Wordss is a recognition technique originally designed for text classification and lately successfully adapted to 2D object recognition [7]. The core concept of this latter technique is the description of objects through a weighted set of visual words chosen from a fixed vocabulary. The vocabulary $V = \{w_1 \dots w_N\}$ is a set of N representatives of all possible feature descriptors that the robot is likely to detect in its typical environment. The vocabulary is trained by collecting all feature descriptors detected in the environment and by clustering them in order to obtain the representatives; each representative w_i is called visual word and the centroid $\mathbf{c}_i \in \mathbb{R}^n$ of the cluster it represent is used as its descriptor.

Once the vocabulary has been trained the world observed by the robot is described by means of visual words. Whenever the robot detects feature the vocabulary is checked in order to substitute it with one of the visual words. One feature $\{F, \mathbf{d}\}$ is always substituted by its closest visual word $\hat{w}$ where the distance is given by the euclidean norm of the descriptors difference:

$$\hat{w} = \arg\min_{w_i \in V} \|\mathbf{d} - \mathbf{c}_i\| .$$

In order to limit the ambiguity, features whose word assignment is ambiguous are discarded. An assignment is considered ambiguous when the distance ratio r_{ij} between the

feature $\{F, \mathbf{d}\}$ and its two closest words $\{w_i\, w_j\}$ is higher than a predefined threshold (i.e. 0.8)

$$r_{ij} = \frac{\|\mathbf{d} - \mathbf{c}_i\|}{\|\mathbf{d} - \mathbf{c}_j\|} > \bar{r}\,.$$

During the training phase of a Bag of Words model all visual words detected for an object are collected in a histogram. The histogram of an object counts the number of detections for each each visual word in the vocabulary, this represents a simple yet effective discriminator for objects category.

It is important to notice that the size of the vocabulary is a key parameter of this technique, an insufficient number elements limits the discriminative power of the visual words but having too many elements leads to noisy histograms.

In the presented work has been introduced in order to aid the recognition process but also to enhance the detection robustness at the same time: since each visual word is the representative of a large set of original descriptors, small variations in a detected descriptor are likely to lead to the same visual word.

Chapter 4

Modeling

The process that aggregates the features collected for an object in order to distinguish it from others, namely the modeling, is a key topic in the object recognition research. Object models may be interpreted as high-level descriptors and share many of the required characteristic with the simpler visual features. In particular a model should be as much robust to noise as possible, this includes small occlusions, light changes or sensors noise. Some additional requirements may be posed by the particular use case, this is also the case of the autonomous robotics and these key modeling aspects will be briefly discussed below.

In the analyzed scenario the set of objects the robot will be required to recognize is not fixed and object models needs to be learned or refined at run-time. This implies object models need to be integrated with newly detected features in a scalable manner while the robot observe the associated object from different point of views. Moreover, since mobile autonomous robots are often equipped with a manipulation arm, the only recognition of objects present in the observed scene is not sufficient: in most cases the correct interaction with a recognized object requires the identification of its pose with respect to the robot. Models created in this phase should not only provide a tool to recognize visible objects but also to localize them with a full 6 DoF pose.

Autonomous robots are usually expected to fulfill assigned tasks with time performances comparable to humans. Along with the limited computational power this severely limits the amount of computation time available for the processing of each frame provided by the robot sensors. Although this fact is likely to limit the quality in the recognition results mobile robots can overcome the issue by refining these results while new data is acquired. The recognition quality requirements can be lowered in a first instance to favor the response time as long as these results can be efficiently improved by an integration of newly acquired frames.

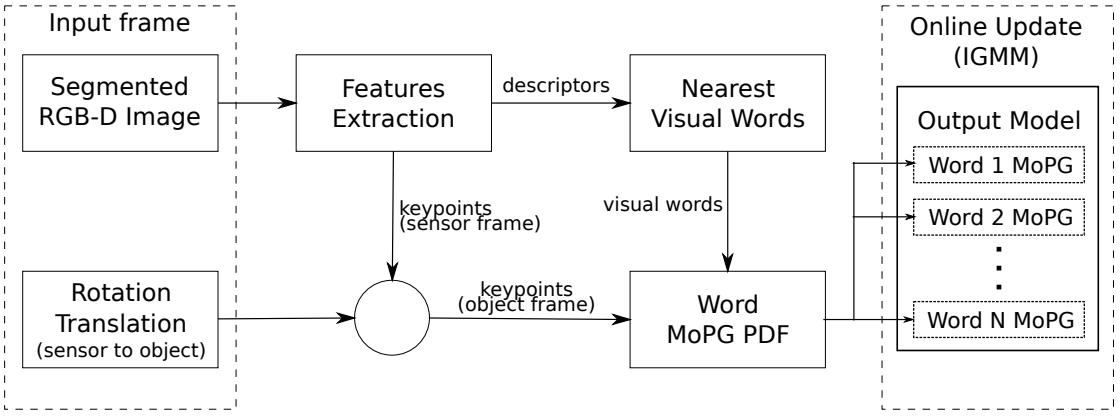

FIGURE 4.1: Model training flow diagram. The modeling input is a set of RGB-D frames, the viewpoint viewpoint of each frame and a region of interest that specify the learned object bounds. The keypoints of all detected features are transformed from the sensor reference frame to the object reference frame in order to allow an incremental update. Each feature is then substituted with its closest visual word by exploiting a Bag of Words paradigm. The set of visual words along with their keypoints is integrated in the words spatial distributions that compose the output object model. Precisely, all keypoints associated with the same visual word are used to train a probability distribution over the pose space, namely the Mixture of Projected Gaussian.

Object Modeling The algorithm proposed in this thesis is focused in the fulfillment of these peculiar requirements. The model designed in order to identify an object is based on learning the spatial distribution of its detected features in respect to its assigned reference frame. The keypoints at which an object features are observed directly reflects the object geometry and appearance (see figure 4.2) thus only visually similar objects are likely to produce similar models. Maintaining this strong link between an object model and its geometry also aids in its localization in terms of position and orientation.

The flow diagram of an object modeling process is shown in figure 4.1 and each its part will be exhaustively presented in the next sections of this chapter. As described in section 2.1 the input of the proposed algorithm consists on a RGB-D video and the sequence of the poses from which the RGB-D sensor shot each frame. Additionally, during the model training a region of interest on each frame is also needed in order to specify the learned object bounds. The keypoints of the features detected on each frame are transformed from the sensor reference frame to the object reference frame in order to allow an incremental update of the model. Each feature is then substituted with its closest visual word by exploiting a Bag of Words paradigm as described in section 3.2. The set of visual words along with their keypoints is then integrated in the words spatial distributions that compose an object model. Precisely, all keypoints associated with the same visual word are used to incrementally train a probability distribution over the 6 DoF pose space, namely the Mixture of Projected Gaussian described in section 4.1.1.

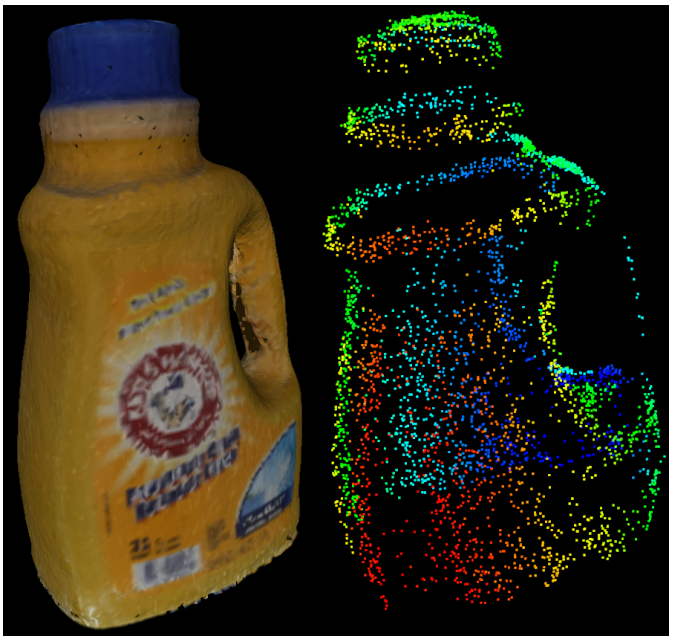

FIGURE 4.2: The keypoints point cloud (right) can represent an object appearance well (left). An efficient alternative of collecting and keeping all keypoints is to learn their spatial distribution as described in section 4.1.

Scene Modeling As previously discussed the proposed models contain the keypoint distribution of modeled objects hence the localization process is connected to finding an area in the scene with a similar keypoint distribution. For this reason the scene model is created through the same process used for object models. The techniques adopted in order to efficiently learn such features spatial distribution are described in the following sections, while the methods developed for comparing these models are described in chapter 5 .

4.1 Statistical Modeling

A model is built by the integration of the features collected by the robot as long as an object is observed by the robot, possibly from different points of view. In order to maintain the process scalable, the proposed model does not keep every single keypoint found in memory but instead learns and keeps only a few parameters of the spatial probability distribution of all these keypoints. A statistical modeling not only limits the amount of data kept in memory but also improves the robustness of the system to sensors noise.

One of the desired properties for local features is the repeatability, that is, its capacity to be detected on the same point of an object regardless of changes in the viewpoint, light conditions or minor deformations. In practice even with a good repeatability

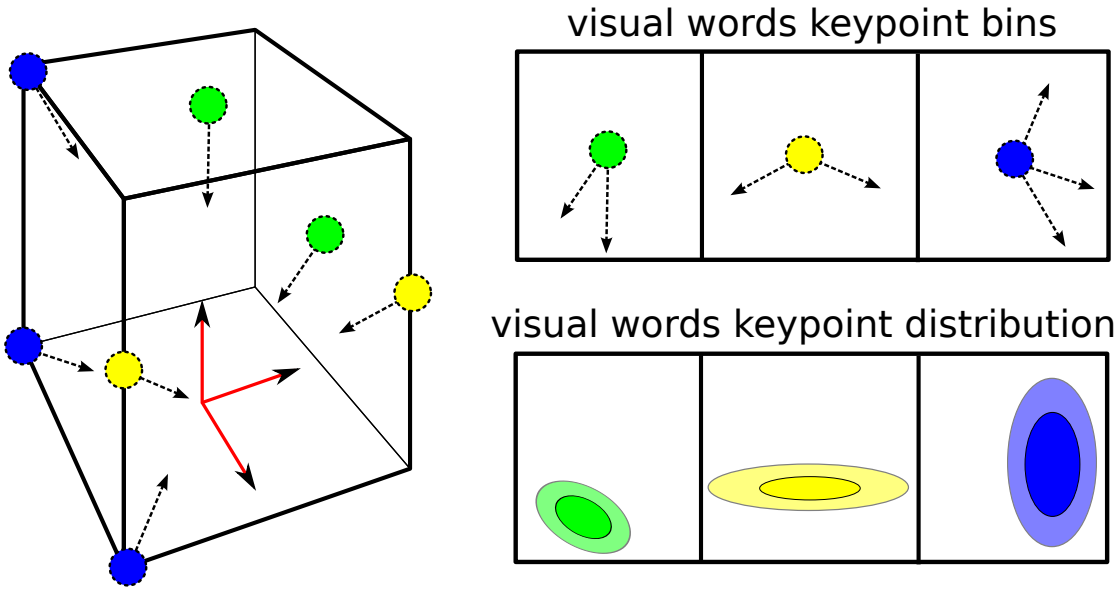

FIGURE 4.3: The keypoints point cloud (right) can represent an object appearance well (left). An efficient alternative to collecting and keeping all keypoints is learning their spatial distribution as described in section 4.1. Each keypoint is first referred to the object reference frame then all keypoints found are clustered by visual word value and integrated in the associated MoPG distribution.

the keypoint of a feature is always affected by random noise and its location slightly varies among different frames. The best distribution for modeling the spatial error in the keypoints detection is difficult to generalize with respect to the wide range of the possible RGB-D sensors but the Gaussian distribution is a common choice that fits most cases well.

As discussed in section 3.2, the method presented in this thesis describes objects by means of a small set of visual words so that each word can be detected in several locations of the same object. The natural choice to approximate the spatial distribution of visual words spatial distribution is the Mixture of Gaussian (see section 4.1.1). This distribution fits the purposes of the presented algorithm well not only thanks to some of its algebraic properties (see paragraph 5.1) but also because it can be incrementally learned.

Since our model tracks the spatial distribution of every visual word independently, one different Mixture of Gaussian is learned for every different word that has been detected during an object modeling. In order to fully exploit the Bag of Words method every visual word is also associated to a counter that tracks its detection rate in respect to the others. These counters have the same purpose of the Bag of Words histogram and can be exploited in order to guess an object category thus aiding its recognition process.

4.1.1 Gaussian Mixtures

The Mixture of Gaussian (MoG) is a widely used probability distribution whose probability distribution function (PDF) is given by the following equation:

$$\mathbf{x}, \boldsymbol{\mu}_i \in \mathbb{R}^n, \quad \Sigma_i \in \mathbb{R}^{nxn}$$
$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_i, \Sigma_i) \quad \text{with} \quad \sum_{i=1}^{n} w_i = 1 \tag{4.1}$$

where

$$\mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})} \tag{4.2}$$

is the PDF of the Normal distribution.

As discussed at the beginning of this chapter, the presented method aims to approximate the spatial distribution of 6 DoF keypoints by means of a MoG but there are some issues when dealing with such data points. The 6 DoF keypoints are comprehensive of both the position and orientation and lie inside the $SE(3)$ that is the roto-translation manifold. The definition of many algebraic operations, like the sum, have different definitions and behavior when they are applied to points in $SE(3)$ rather than in an Euclidean group. For this reason several core tools of statistical analysis are not compatible with data lying on a manifold, most training algorithms are among these (see section 4.2).

In literature some solutions have been proposed to overcome this issue but the solution that best fit with the purpose of the presented work is the approximation of the MoG to a Mixture of Projected Gaussian (MoPG). The MoPG has been proposed by Feiten et al. [6] and exploits a projection the $SE(3)$ points to a tangent space in $\mathbb{R}^6$. The distribution of a set of projected points can be parametrized by a Normal distribution through classical approaches thus leading to an approach for creating a distribution similar to the MoG. In the following paragraph the MoPG is briefly presented.

Mixture of Projected Gaussian Following the original work [6] of Feiten et al. our approach adopted Dual Quaternions (DQ) as representation for the 6 DoF keypoints of the detected features. Quaternions are a well known algebraic object used to calculate the product and the sum of points in the rotation group; dual quaternions extend the quaternions in order to handle points in the roto-translation group. The dual quaternions

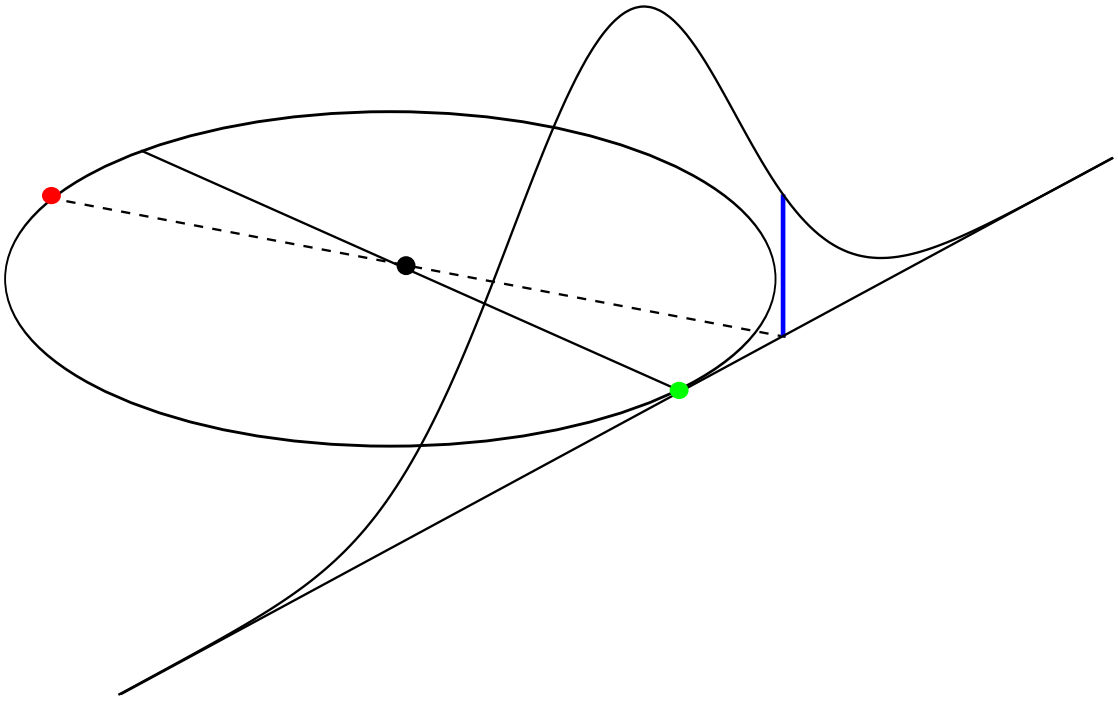

FIGURE 4.4: The figure represents a Gaussian PDF over the plane tangent to a point (green dot) of the manifold. The probability density (blue line) of a point in the manifold (red dot) is evaluated in the Gaussian PDF; the evaluation point is obtained through central projection.

ring $\mathbb{H}_D$ is defined as

$$\begin{aligned} \mathbb{H}_D &= \{dq | dq = q_1 + \epsilon \cdot q_2;\ q_1, q_2 \in \mathbb{H}\} \\ dq^A + dq^B &= (q_1^A + q_1^B) + \epsilon \cdot (q_2^A + \epsilon \cdot q_1^A) \\ dq^A * dq^B &= (q_1^A + q_1^B) + \epsilon \cdot (q_2^A * q_1^B + q_1^A * q_2^B) \\ dq^* &= q_1^* + \epsilon \cdot q_2^* \end{aligned} \tag{4.3}$$

where $\mathbb{H}$ is the quaternions ring and ϵ is a dual unit with the following properties $\epsilon \cdot 1 = 1 \cdot \epsilon$ and $\epsilon^2 = 0$. Any rigid 3D transformation can be represented as a DQ; given a transform T with rotation expressed as a quaternion q_r and translation embedded in a quaternion $q_t = [0\, t_x\, t_y\, t_z]$ the dual quaternion $dq^T = q_r + \epsilon \cdot 0.5 q_t * q_r$ represents the transform. The unit quaternion q_r that represent the rotation of a DQ lies on the unit sphere $S(3)$ embedded in $\mathbb{R}^4$. Accordingly with the procedure described by Feiten at al, a 6D space TS_{q_r} tangent to dq^T is constructed by taking the tangent space of $S(3)$ in q_r and extending it to include the translation coefficients of q_t.

Given a projection point q_r Feiten and Lang propose a procedure in order to create a mapping from $SE(3)$ to TS_{q_r} (see figure 4.4)

$$\Pi_{q_r} : TS_{q_r} \longrightarrow S(3) \times \mathbb{R}^3 \sim SE(3)$$

and given a transform m defines the Projected Gaussian (PG) PDF as

$$\begin{aligned} \mathcal{N}(m \,|\, q_r,\, \boldsymbol{\mu},\, \Sigma) &:= \frac{1}{C} p_{TS}\left(\Pi_{q_r}^{-1}(m)\right) \\ C &:= \int_{S(3)\times\mathbb{R}_3} p_{TS}\left(\Pi_{q_r}^{-1}(m)\right)\, \mathrm{d}m \end{aligned} \tag{4.4}$$

where $p_{TS}(m)$ is the PDF of a Normal distribution $\mathcal{N}(\boldsymbol{\mu},\, \Sigma)$ on the tangent space. The probability on $SE(3)$ of a transform m_0 with a rotation orthogonal to q_r is defined as zero for smooth completion.

4.2 Model Training

Given a set of N keypoints $K = \{k_i\}_N$, the parametrization of their distribution through a Mixture of Projected Gaussian is a key aspect of the presented algorithm. Although the projection of SE(3) data to an Euclidean tangent space simplifies the determination of the parameters for each projected Gaussian component the optimal convex combination of projected Gaussians poses further issues. Training a Mixture of Projected Gaussian is a problem similar to the non projected version and algorithms originally designed for Mixture of Gaussian can be easily adapted for this purpose. The methods that gave the best results for both the on-line and batch training scenarios are described in the following section.

4.2.1 Batch Training

In [19] , Feiten et al. propose a variant of the classical EM algorithm [5] adapted in order to train a MoPG with M components:

1. Set the initial value for the means $\boldsymbol{\mu}_i$, covariance matrices Σ_i and weighting coefficients λ_i and evaluate the log likelihood with these values.
2. E step: Evaluate the responsibilities $\gamma(k_n, i)$ using the current parameter values:

$$\gamma(k_n, i) := \frac{\lambda_i \mathcal{N}(k_n \,|\, q_i,\, \boldsymbol{\mu}_i,\, \Sigma_i)}{\sum_k \lambda_k \mathcal{N}(k_n \,|\, q_r,\, \boldsymbol{\mu}_k,\, \Sigma_k)}$$

3. M step: Estimate the new parameters using the current responsibilities:

$$\boldsymbol{\mu}_i^{new} = \frac{1}{N_i} \sum_{j=1}^{N} \gamma(k_j, i) \cdot k_j$$

$$\Sigma_i^{new} = \frac{1}{N_i} \sum_{j=1}^{N} \gamma(k_j, i)(k_j - \boldsymbol{\mu}_i^{new})(k_j - \boldsymbol{\mu}_i^{new})^T$$

$$\lambda_i^{new} = \frac{N_i}{N}$$

   where $N_i = \sum_{j=1}^{N} \gamma(k_j, i)$.

4. Evaluate the log likelihood:

$$\sum_{j=1}^{N} \ln \left(\sum_{i=1}^{M} \lambda_i \mathcal{N}(k_j \,|\, q_i,\, \boldsymbol{\mu}_i,\, \Sigma_i) \right)$$

   and check the convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to the E step.

Similarly to EM this algorithm is not suitable for on-line integration of new data but in our experiment gave the best results. A possible use case scenario is the training of the model relative o an object that is being manipulated in the robot arm or through a turn-table: in these cases the training could be started as a batch process only after the acquisition of sufficient data.

Like EM the the algorithm proposed by Feiten at al. requires a prior knowledge of the optimal number of components for Mixture of Projected Gaussian. This information can be estimated through common entropy based criteria like AIC or BIC.

4.2.2 Online Training

In a realistic scenario some objects are only partially visible to the robot when it first learns them. The need to integrate the already learned models with new data comes as soon as the robot sees these objects from additional view points. In our work the IGMM algorithm [20] proposed by Engel et al. has been exploited in order to adapt the parameters of a previously learned MoPG accordingly to incoming new data. The IGMM algorithm automatically tries to estimate the optimal components number by adding new components whenever the support of the newly added keypoint is lower then a threshold specified by a novelty factor τ. According to this definition, given an MoPG of M components a newly added keypoint k_j is integrated as new component if

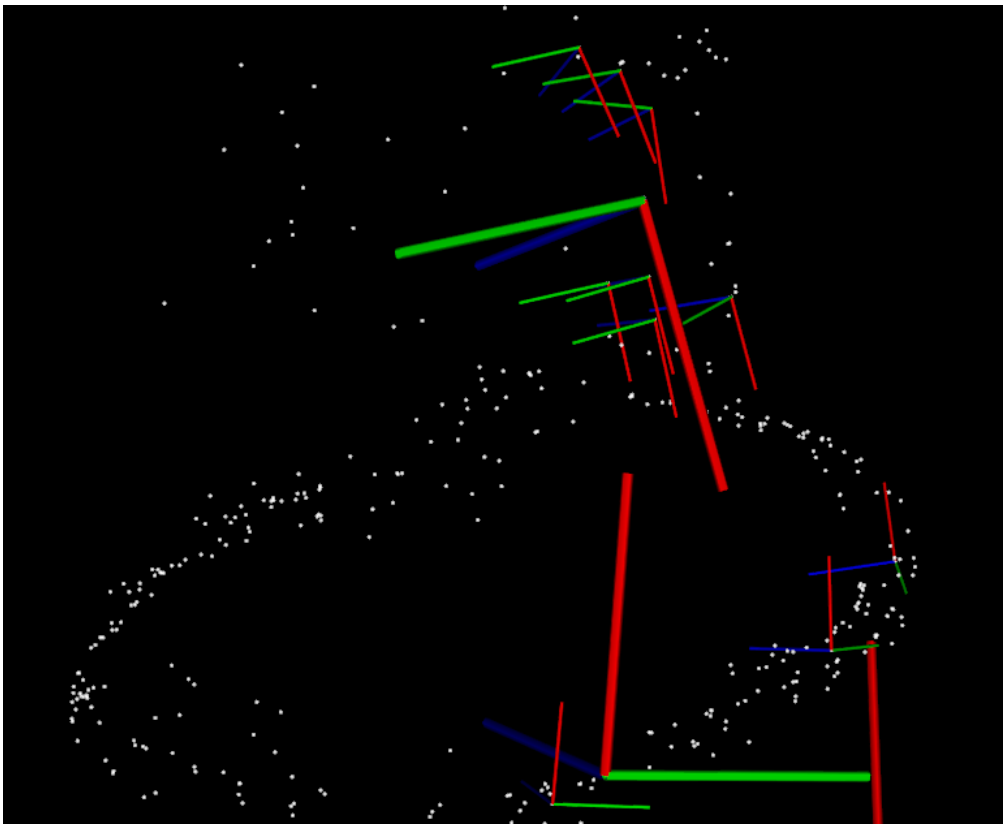

FIGURE 4.5: The white dots represent the top part of the 6 DoF keypoints cloud shown in figure 4.2. The small reference frames are the subset of keypoints relative to a single visual word and the large reference frames are the mean values of the associated MoPG components, learned through the IGMM algorithm.

and only if

$$\mathcal{N}(k_j \,|\, q_i, \boldsymbol{\mu}_i, \Sigma_i) < \frac{\tau}{\sqrt{(2\pi)^6 |\Sigma_i|}} \quad \forall i = 1..M$$

otherwise k_i is integrated in the MoPG as described in [20].

4.3 Global Features Modelling Approach

The modeling technique proposed in this chapter is designed to make use of local features in order to track the keypoints distribution. Global features refer to properties that are related to the whole object (i.e. its height) thus it would be conceptually wrong considering their spatial distribution. Nevertheless, in practice the perception of some global features may be affected by the observer viewpoint. To this end the MoPG presented in section 5.1 can be adapted in order to track the change of a global feature in respect to the viewpoint; the following paragraph explores a use case related to this variant [8], developed during as part of the collaboration between the Intelligent

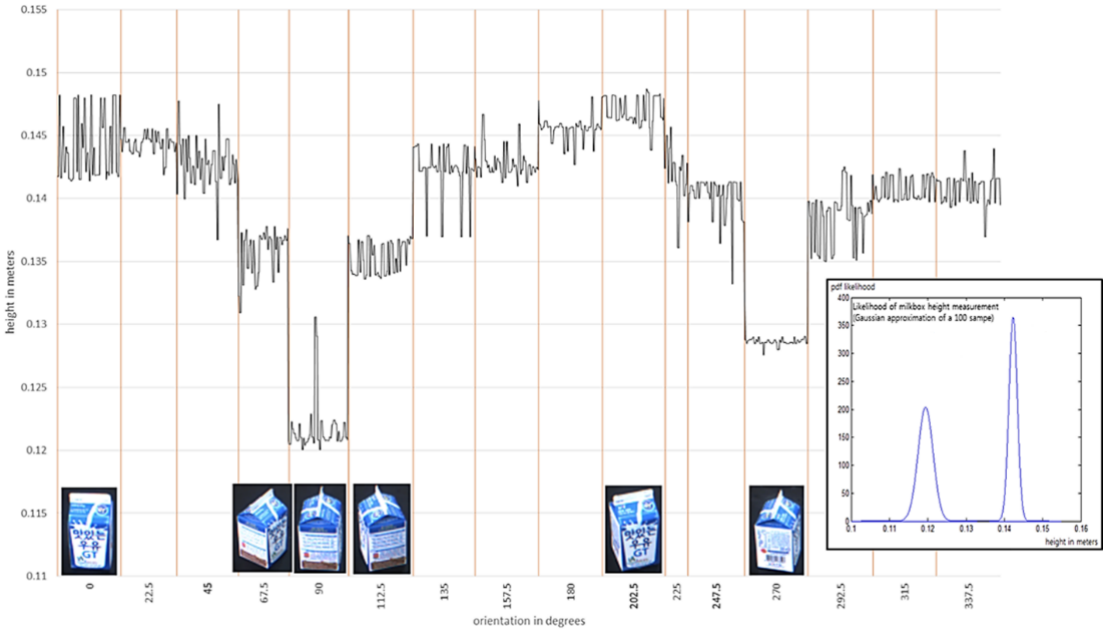

FIGURE 4.6: Example of the changes that may affect a global feature descriptor in respect to the observer viewpoint. The measured height of the milk box varies according to its orientations as the thin panel at the top may not be detectable at the particular orientations of the milk box.

Autonomous Systems Laboratory (IAS-Lab) at Padua University and the Intelligent Systems Research Institute (ISRI) in Sungkyunkwan University.

Similarly to what proposed by Lee et al. in [21], in the presented case study the considered global features consist of object height, width and aspect ratio. The descriptor of these features that may take different values in respect to the orientation as well as to the distance, due to the complexity of the 3D shape of the object. This is illustrated in the left side of figure 4.6, where the measured height of the milk box varies according to its orientations as the thin panel at the top may not be detectable at the particular orientations of the milk box. This leads to a singularity in measurements for the given sensor and sensing algorithm. The proposed MoPG overcomes this representation problem by providing a multi-modal likelihood distribution of the descriptor over the pose space.

Chapter 5

Recognition

The models created with the technique in chapter 4 contain information about the keypoint spatial distribution of object features. The recognition process exploits this information in order to distinguish objects from each other or localize them in the scene. In general the object recognition task is independent from the localization of the object which often involves an onerous search, such as sub-windowing or scene segmentation, in order to find the candidate objects. Instead of running the recognition and the localization processes separately the algorithm presented in this thesis unifies the two tasks in order to exploit their strict correlation.

As discussed in chapter 4 the localization task is connected to finding an area in the scene with a similar keypoint distribution (see figure 5.1). Finding the optimal registration points of a pattern over a larger image is a very common task for 2D object recognition

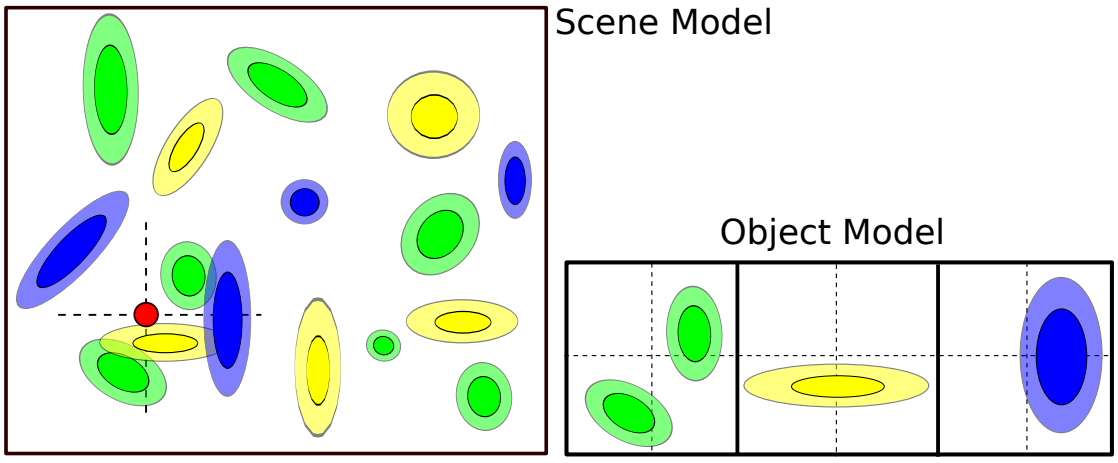

FIGURE 5.1: An object instance (red dot) is guessed at a point in which the words spatial distribution is similar to the the one learned for the instance model. The search for the optimal registrations is performed separately for each of the words MoPG; the final recognition score is given by the sum of individual words registration scores.

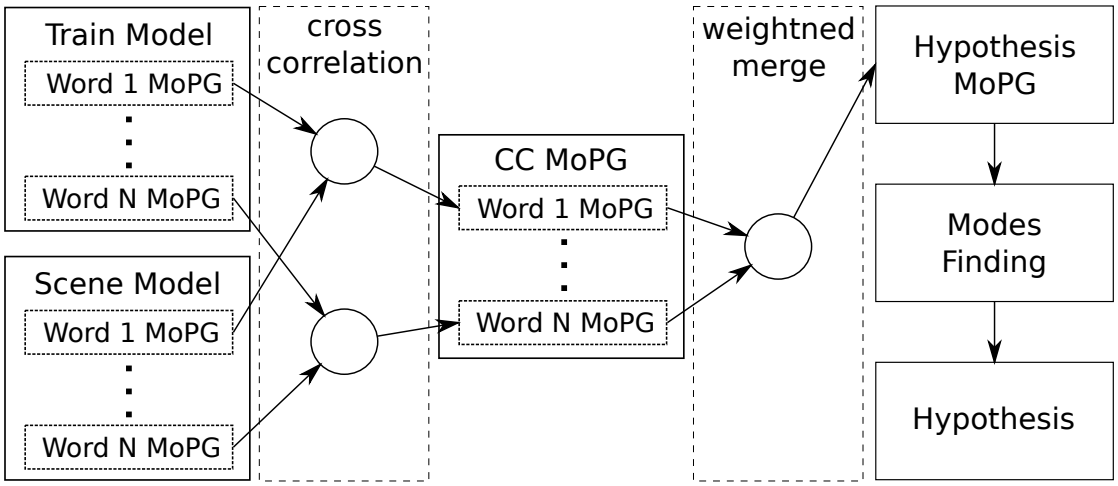

FIGURE 5.2: Model detection flow diagram. The MoPG of corresponding visual words are cross-correlated together in order to find the registration hypothesis contribution of each visual word. Thanks to the closure of the MoPG under the cross-correlation operator, the result is still a set of MoPG. All the components of these MoPG are then fused together exploiting the BoW histogram in the weighted merging process. The peaks in the resulting MoPG are then retrieved by an efficient mode finding algorithm; the peaks are $SE(3)$ and represent the location hypothesis of the model, the peak value is proportional to the confidence of the instance detection.

and the presented algorithm is inspired to one of the most used template matching techniques namely cross-correlation (see section 5.1). Since both scene and object models are composed by a set of MoPG the registration point of these two entities will be in the MoPG PDF space, that is, a dual-quaternion in $SE(3)$. A given model M_T can be registered in the scene model M_S in several locations $\{dq_i\}$ (see figure 5.1); a good registration represents an instance guess $I_i^T = (dq_i, l_i)$ for the template object T and is associated with a registration quality l_i which represents the detection likelihood.

Although the recognition process of the presented models may result more complex than other black-box algorithms (i.e.: SVM or Neural Networks) the strong link between a model and the object geometry allows several optimizations. If an instance likelihood is insufficient for the robot needs the scene model can be incrementally refined by moving the robot around the instance location in order to add RGB-D data from different viewpoints. Moreover, if in a given registration some areas of the object and scene do not overlap, the robot will gain insight of how the scene should be observed in order to increment the result likelihood. Another simple strategy to improve guess confidence is to exploit the word detection counters associated with each model (see section 3.2): the word distribution of a guessed object can be compared to the word histogram of the candidate model and their distance can be used to rise or lower the result confidence.

Although the cross-correlation is in general computationally expensive the presented recognition method exploits some algebraic properties of the Gaussian function in order

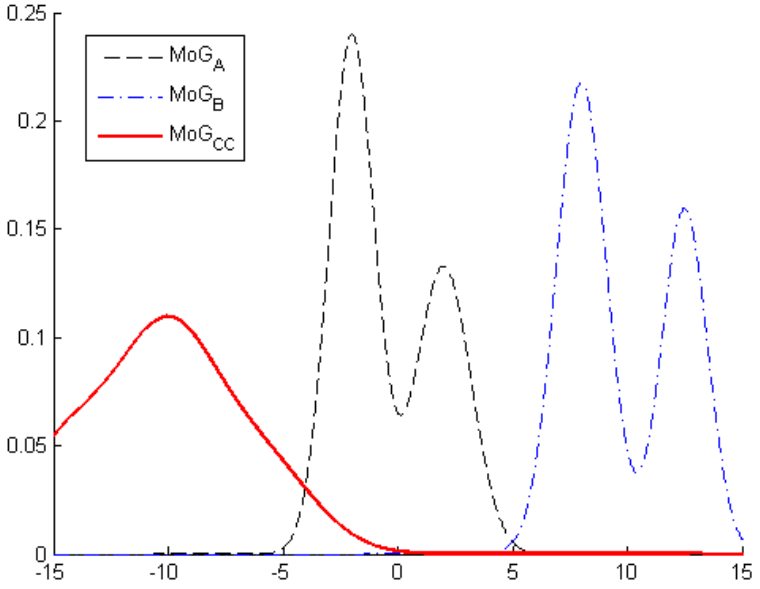

FIGURE 5.3: The point in which the cross-correlation function (MoG_{CC}) presents a peak corresponds to the translation at which the source MoG (MoG_B) is best registered over the destination MoG (MoG_A). The height of the peak is proportional to the registration quality of the two source signals.

to provide an efficient solution for the cross-correlation of MoG or MoPG distributions. Section 5.1 presents the algebraic basis of this solution. Once the cross-correlation function is obtained a peak detection method is needed in order to find the guessed instance locations, an efficient algorithm that fits well with our purposes is described in section 5.2.

5.1 Cross-Correlation

The efficiency of the proposed method derives from the closure of the Gaussian function under the convolution operator. Given the PDFs of two Multivariate Normal distributions

$$\mathbf{x}, \boldsymbol{\mu}_i \in \mathbb{R}^n, \quad \Sigma_i \in \mathbb{R}^{nxn}$$

$$X_1 = \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1), \quad X_2 = \mathcal{N}(\boldsymbol{\mu}_2, \Sigma_2)$$

$$f_{X_i}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k |\Sigma_i|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i)}$$

their convolution

$$(f_{X_1} * f_{X_2})(\mathbf{x}) = \int^{\mathbb{R}^n} f_1(\boldsymbol{\tau})\, f_2(\mathbf{x} - \boldsymbol{\tau})\, d\boldsymbol{\tau} = \frac{1}{\sqrt{(2\pi)^k |\Sigma_c|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_c)^T (\Sigma_c)^{-1} (\mathbf{x}-\boldsymbol{\mu}_c)} \tag{5.1}$$

is another Multivariate Normal distributed PDF, with $\boldsymbol{\mu}_c = \boldsymbol{\mu}_1 + \boldsymbol{\mu}_2$ and $\Sigma_c = \Sigma_1 + \Sigma_2$.

The cross-correlation of any two real continuous functions f_1, f_2 can be computed in terms of their convolution as follows

$$(f_1 \star f_2)(\mathbf{x}) = f_1(-\mathbf{x}) * f_2(\mathbf{x})\,. \tag{5.2}$$

This strict relation can be exploited along with (5.1) in order to obtain a closed form solution for the cross-correlation of a Normal distributed functions. Recalling the Multivariate Normal PDF (4.2) we can can define a Normal distributed variable $\bar{X}_1$ such as

$$f_{X_1}(-\mathbf{x}) = f_{\bar{X}_1}(\mathbf{x}) \tag{5.3}$$

where $\bar{X}_1 = \mathcal{N}(-\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$.

From equation (5.2) and (5.3), we can define a Normal distributed variable $C_{1,2}$ whose PDF is the cross-correlation of f_{X_1} and f_{X_2}:

$$f_{C_{1,2}}(\mathbf{x}) := (f_{X_1} \star f_{X_2})(\mathbf{x}) = (f_{\bar{X}_1} * f_{X_2})(\mathbf{x})$$

where

$$\begin{aligned} C_{1,2} &= \mathcal{N}(\boldsymbol{\mu}_{1,2}, \Sigma_{1,2}) \\ \boldsymbol{\mu}_{1,2} &= \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1 \\ \Sigma_{1,2} &= \Sigma_2 + \Sigma_1. \end{aligned} \tag{5.4}$$

These results can be extended from Gaussian functions to Mixture of Gaussian or to Mixture of Projected Gaussian distributions. Let A and B be two MoGs:

$$\begin{aligned} f_A(\mathbf{x}) &= \sum_{i=1}^{N} w_i^A f_{X_i^A}(\mathbf{x}), \quad \sum_{i=1}^{N} w_i^A = 1, \quad X_i^A = \mathcal{N}(\boldsymbol{\mu}_i^A, \Sigma_i^A) \\ f_B(\mathbf{x}) &= \sum_{j=1}^{M} w_j^B f_{X_j^B}(\mathbf{x}), \quad \sum_{j=1}^{M} w_j^B = 1, \quad X_j^B = \mathcal{N}(\boldsymbol{\mu}_j^B, \Sigma_j^B)\,. \end{aligned}$$

Exploiting the distributive property of the convolution we obtain

$$(f_A \star f_B)(\mathbf{x}) = \sum_{i=1}^{N}\sum_{j=1}^{M} w_i^A f_{X_i^A}(\mathbf{x}) \star w_j^B f_{X_j^B}(\mathbf{x})$$

and from (5.3) we can express it in terms of convolutions

$$\begin{aligned}(f_A \star f_B)(\mathbf{x}) &= \sum_{i=1}^{N}\sum_{j=1}^{M} w_i^A w_j^B f_{\bar{X}_i^A}(\mathbf{x}) * f_{X_j^B}(\mathbf{x}) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} w_{i,j} f_{C_{i,j}}(\mathbf{x}) \end{aligned} \tag{5.5}$$

where from (5.4)

$$\begin{aligned} w_{i,j} &= w_i^A w_j^B \quad \forall i,j \\ f_{C_{ij}}(\mathbf{x}) &= \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}_j^B - \boldsymbol{\mu}_i^A, \boldsymbol{\Sigma}_j^B + \boldsymbol{\Sigma}_i^A) \quad \forall i,j\,. \end{aligned} \tag{5.6}$$

Since $\sum_{i=1}^{N}\sum_{j=1}^{M} w_{ij}^{CC} = 1$, the cross-correlation $C_{A,B} := (f_A \star f_B)(\mathbf{x})$ is still an MoG with NM components. Since the MoPG is a convex sum of Gaussian function the same results can be derived for the MoPGs with a slight modification: the convolution of two projected Gaussian PDF must be done in the same tangent space thus the tangent point of the second operand must be changed prior to the tangent point of the first.

5.2 Mode Finding

Let A, B be two MoPGs and $C_{A,B}$ their cross-correlation, the peaks $\{dq_1, ..., dq_n\}$ in the PDF of $C_{A,B}$ represent the 6 DoF poses for which A is best registered over B (see figure 5.3). As discussed in section 5.1 MoPG are closed under cross-correlation so $C_{A,B}$ is another MoPG. This property is exploited in order to provide the proposed recognition method with an efficient mode finding technique based on Carreira and Perpinan [4] algorithm.

In [4] Carreira and Perpinan provide several important results in order to constrain the search for modes in a MoG. In particular Carreira et al. provide a partial proof that the number of modes cannot be more than the number of components, and are contained in the convex hull of the component centroids. These results along with a derivation of the exact Hessian and gradient formulas for the MoG has been exploited in order to obtain an efficient gradient ascending algorithm for MoG mode finding.

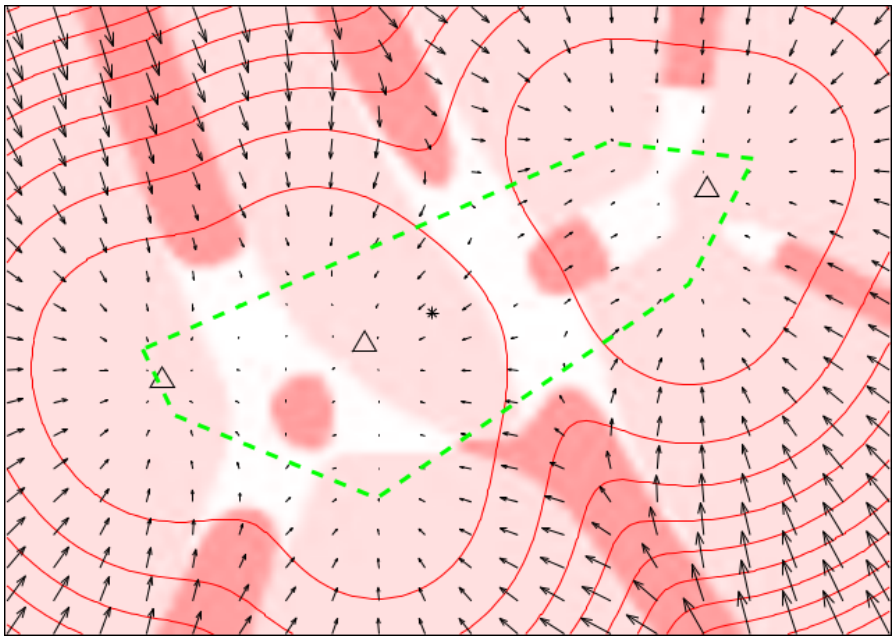

FIGURE 5.4: The modes (triangles) in a MoG distribution are always located inside the convex hull (green line) of its component centroids. In [4] Carreira et al. provide Hessian and gradient formulas for the MoG and describe an efficient gradient ascend algorithm in order to find these modes.

The Hessian and gradient formulas provided by Carreira et al. are valid also for the components of a MoPG since in its tangent space a Projected Gaussian is similar to a Normal distribution (see section 4.1.1).

The downside of the cross-correlation procedure explained in section 5.1 is that the number of components in $C_{A,B}$ is quadratic with respect to the input, namely NM where N and M are the size of A and B. It is important to notice that the variability in the $C_{A,B}$ weights is greatly accentuated in respect to the source mixtures due to their multiplications in (5.6). This variability can be exploited by removing from $C_{A,B}$ all the components whose weight is less than a threshold $\theta > 0$ (e.g. $\theta = 0.01$) in order to speed up the mode-finding process. This practical approximation has been first proposed by Carreira et al. in [4] and is justified by the really low impact that these low-probability components have in the modes position.

Chapter 6

Results

6.1 Experimental Setup

The experiments presented in this section have been run over two different dataset, namely the BigBIRD and the RGB-D Dataset presented in section 2.1.3. Both these two dataset provide the viewpoint information required by the presented method but its presentation differs and the two dataset required a slightly different setup.

The point clouds associated to the RGB-D frames provided by the BigBIRD dataset are not organized (see section 2.1.1) thus the SIFT back-projection method described in section 3.1 is not suitable; the natively 3D features are the only practical option, to this end the PFHRGB features [22] have been used for their ability to take into account both the shape and the color information. For the RGB-D Dataset the organized point clouds allowed to use 2D features so, in order to test the proposed algorithm with different description methods, the SIFT back-projection has been used.

6.1.1 Training

The output of the modeling process described in chapter 4 is a collection of Mixture of Projected Gaussian (see section 4.1.1); each of these MoPGs approximates the spatial distribution of the locations in which its associated Visual Word (see section 3.2) has been observed during the model training. Thus the training phase involves a preliminary batch phase in order to train the Visual Vocabulary and a run-time process for the MoPG training.

Vocabulary Training The visual vocabulary is a set of representatives chosen among all possible feature descriptors that the robot is likely to observe in the environment.

It's important to notice that the set of descriptors used to train the visual vocabulary should not be limited to the ones found in known objects since the deriving visual words may not be general enough to describe new unknown objects. Thus in the presented work both objects and scenes have been processed in order to extract the set feature descriptors used in the vocabulary training.

The vocabulary trained in the experiments associated with the presented results consists of 200 visual words obtained through a k-mean clustering among all collected descriptors. As described in section 3.2 whenever a feature is later detected in the environment its two closest visual words are retrieved from the vocabulary through a KNN search; if the distance ratio between the first and the second closest visual words is over 0.8 the feature is discarded otherwise it is substituted with its closest visual word.

Model Training As previously discussed a model is a collection of MoPGs whose parameters are trained incrementally through the IGMM algorithm exposed in section 4.2. As explained in section 2 a fundamental prerequisite for training these spatial probability distributions is that all added keypoints are referring to the same reference frame. There exist several ways for recovering the viewpoint of an RGB-D frame (see section 2.1.2) but the most suitable for testing purposes is the exploitation of a turntable. By knowing the RGB-D sensor pose in respect to the turntable center and by controlling table rotation it is possible to compute the 6 DoF viewpoint with precision.

Although both dataset contain the information regarding the rotation angles of the turn table, only the BigBIRD dataset provided the full 6 DoF pose of the camera sensors in respect to the turntable. In order to retrieve the pose of the camera in the RGB-D Dataset an approach based on ICP has been exploited among consecutive frames. Although the ICP method worked quite well on some objects the majority of the regions of interest in the RGB-D frames were too small or noisy to get good results thus often only some parts of an object have been reconstructed. Since the development of a robust SFM or ICP method is out of the scope of this thesis the instance recognition tests have been focused on the objects form the BigBIRD dataset.

6.2 Results

The results presented in the following sections have been collected during the experiments and rely only on the information provided by the cross-correlation of the models MoPG described in section 5.1. Although the exploitation of the Bag of Words histogram could enhance the confidence of these results the aim of the presented thesis is

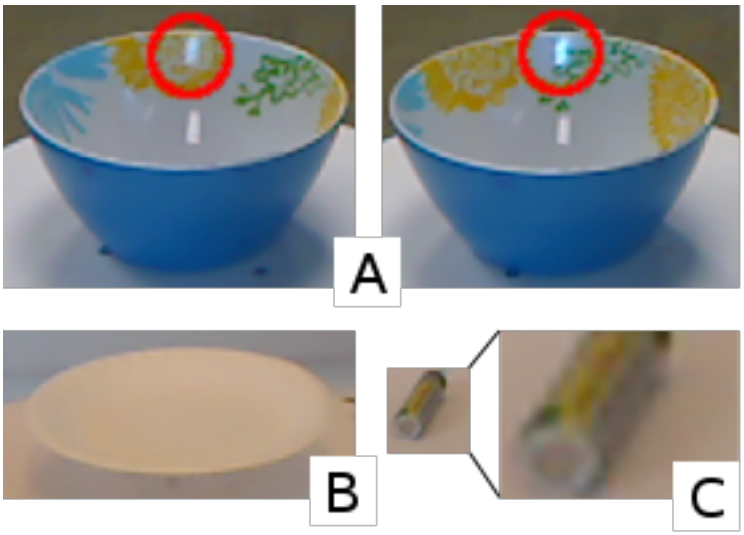

FIGURE 6.1: Example of some issues affecting the proposed recognition method on the RGB-D Dataset. On some objects the light reflections (A) produce a consistent number of SIFT whose keypoints are not consistent with the object rotation, the integration of those keypoints in the model degrade the recognition performances. The absence of texture (B) or the small size (C) of some objects severely limit the number of SIFT keypoints found for the model training.

to evaluate the recognition and localization performances of the cross-correlation when applied to MoPGs. For better generality, independent tests have been evaluated for both the BigBIRD and the RGB-D Dataset but since the two dataset do not allow a similar modeling setup (see section 6.1.1) the strengths of each dataset have been exploited in order to evaluate different aspects.

In order to test the method robustness to sensor noise, the classification experiments have been repeated in three different modalities. In the first mode models were created by using the RGB-D frames of the BigBIRD dataset without any alteration; in the second mode a white noise ($\mu = 4mm, \sigma = 1$) was added to all keypoints in order to test robustness to sensor noise; in the third mode the keypoint noise was combined with the sub-sampling of the RGB-D frames used to train the models, the 50% of overall frames are discarded in order to simulate occlusions.

6.2.1 RGB-D Dataset

The good category and instance organization of the RGB-D Dataset has been exploited to test the category recognition performances (see figure 6.2). The model training for object categories followed the classic leave-one-out procedure: for each category an instance has been selected as query and the other served as training set, the procedure has been repeated varying the query object at each iteration. All keypoints found on

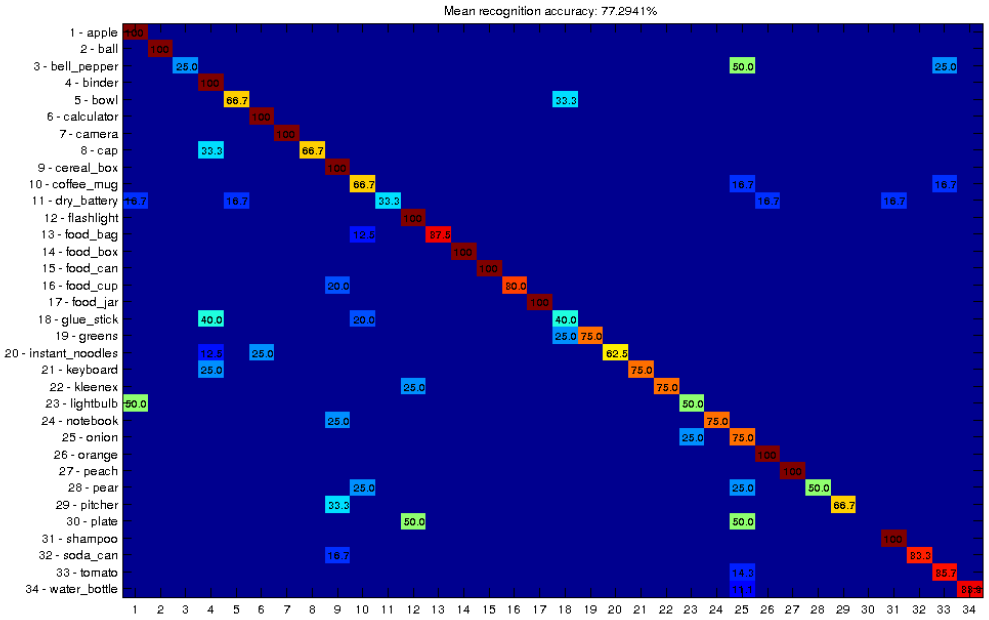

FIGURE 6.2: RGB-D Dataset Category Recognition results. The variability of the results in respect to the category shows a strong dependency to the underlying feature choice: the SIFT keypoints used in the RGB-D Dataset setup rely on the object texture and did not perform well on reflective or untextured objects.

the training objects of a learned category were integrated together and the resulting category models were compared to test object models.

Category Recognition Despite the advantage of allowing the use 2D features this dataset revealed some weaknesses of the modelling setup based on back-projected SIFT (see figure 6.1). Many of the household objects contained in the dataset present strong light reflections or low texture thus compromising the keypoint spatial stability or producing an insufficient number features. For this reason, although on many categories the recognition rate was satisfying, the overall classification rate has been only been of 78%, compared to a 90% of the state of art.

Localization The average localization error reported for the RGB-D Dataset has been $\cos(\Delta_R) = 5°$ degrees in the orientation (rotation cosine) and $\Delta_{\mathbf{t}} = 3mm$ in the position (translation norm). Although these results are comparable to state of art results the deviation of the average error among different categories is marked and reflects the considerations done for the category recognition results. It is important to notice how many of the recognized objects presented significant symmetries (i.e. bowls, balls) that intuitively led to high rotation errors $\Delta_{\mathbf{t}} \sim 2cm$ and $\cos(\Delta_R) \sim 30°$.

BigBIRD Recognition		Frames (%) 100	50
Noise (mm)	0	100%	94%
	4	98%	90%

TABLE 6.1: BigBIRD Instance Recognition results. Introducing a small white noise on the keypoints does not significantly affect the results since the noise is almost diminished by the statistical modeling (see section 4.1). The removal of a frame percentage from the training has more impact on the results, especially on similar objects.

Query	Result	ratio
(A) aunt_jemima_original_syrup	aunt_jemima_original_syrup	0.82
	(B) tapatio_hot_sauce	0.17
(F) palmolive_orange	(E) softsoap_purple	0.56
	palmolive_orange	0.44
(C) quaker_chewy_peanut_butter	quaker_chewy_peanut_butter	0.64
	(D) quaker_chewy_chocolate_chip	0.36

TABLE 6.2: The table shows the first two results for three sample queries; the confidence value is shown next to each result. The image associated to the table shows the meshed model of the involved models. The higher similarity between (C) and (D) in respect to (A) and (B) reflects in a wider gap in the confidence for the results of (A) and (F). In some cases the reflective or transparent parts of some objects, i.e. (E) and (F), compromise the model training and can lead to wrong classifications.

6.2.2 BigBIRD

The use of several and well calibrated RGB-D sensors led to the significantly better quality of the BigBIRD data in respect to the RGB-D Dataset. The increased size of the region of interest of the objects in the RGB-D frames along with the availability of a precise 6 DoF for all point clouds allowed a better evaluation of the instance recognition capabilities of the proposed method.

Instance Recognition On BigBIRD objects the instance recognition results are presented in table 6.1 and show a classification rate of 90% in the worst case scenario. Despite its good quality at the time of writing, the BigBIRD dataset has recently been

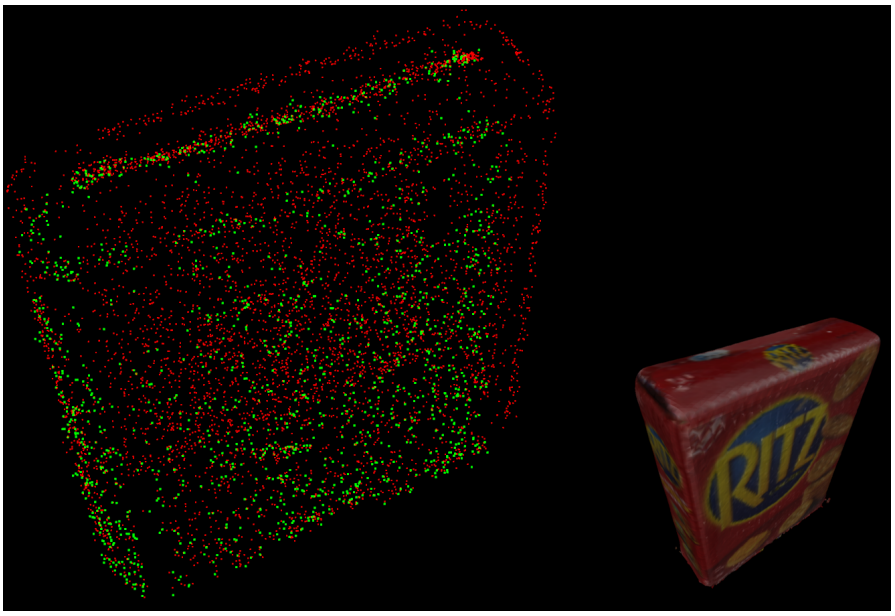

FIGURE 6.3: Localization result for one sub-sampled object in the BigBIRD dataset. The green point cloud represents the set of keypoints of the sub-sampled model, its coordinates are transformed exploiting the localization result in order to register it with the full object model (red point cloud). The recognized model mesh is shown in the bottom-right corner.

published so comparison of the results could not be found. Looking at results with similar dataset, like the RGB-D Dataset, the recognition performances obtained by the proposed algorithm are aligned with other state of art techniques. For every object the two most likely results are compared and the ratio between their likelihoods is presented in order to evaluate the confidence of the instance recognition; three sample results are proposed in table 6.2.

Localization The average localization error reported for the BigBIRD dataset has been $\cos(\Delta_R) = 2^\circ$ degrees in the orientation (rotation cosine) and $\Delta_{\mathbf{t}} = 2mm$ in the position (translation norm). These results are similar to the state of the art and are slightly better than the results obtained for the RGB-D Dataset. The reason behind this precision improvement is the lower number of low-textured or symmetric objects in the BigBIRD dataset.

6.3 Gaussian Mixture Library

Most of the the source code developed for algebraic and statistical tools presented in this thesis has been organized in a general purpose C++ library, namely the Gaussian Mixture Library (GML). The library is designed focusing the portability, the efficiency

and the extendibility of the contained tools following an OO paradigm; to this end all core classes allows templetization for all compile-time parameters. The classes organization follows the design pattern of the Eigen library, which also represents its only non-optional prerequisite; as in the Eigen library the polymorphism for template classes is achieved by exploiting the CRTP pattern.

The GML library is not limited to tools associated to the Gaussian Mixture; indeed most algorithms are designed to work with generic Mixture objects, whose component type can be specified as template parameter. Although the GML library provides the definition for Gaussian or Projected Gaussian components additional distributions can be easily added by extending the base Component class.

The functionalities provided by this library has already been exploited in some projects. Among others in [23] the GML library has been extended introducing the Doughnut distribution as Mixture component. The resulting Mixture, namely the DMM, has been used as regression tool in a learning by demonstration framework.

The library is organized in several modules as follow:

core This module contains all basic functionalities and its basic structures are organized as follows:

- **Mixture**: represents a generic Mixture distribution. This structure is not limited to Gaussian components and allow the specification of the component type as template parameter. The component number can be specified as template parameter for fixed size mixtures or can be omitted in order to allow run-time component insertions or deletions.
- **ComponentBase**: represents the base class of all Mixture components. Accordingly to the CRTP pattern this class can be extended in order to create virtually any kind of Mixture distribution. Extending classes are only required to provide a method that compute the associated distribution PDF; the domain of this PDF needs to be specified as template parameter.
- **GaussianComponentBase**: represent the base class for all Gaussian components. Similarly to ComponentBase this class can be extended following the CRTP pattern. This class provides to extending objects several methods to handle the parametrization of any Normal PDF; among these are present some built-in utilities to retrieve or modify the eigenvalues of the covariance matrix.

- **GaussianComponent**: represent a sample extension of GaussianComponentBase that can be used as Mixture component in order to obtain a Mixture of Gaussian distribution.

train This module provides some tools to infer the parameters of a given Mixture from a set of sampled points. In current implementation some common training algorithms are already provided for both MoG or MoPG distributions, namely the Expectation Maximization, IGMM and K-Means. Some entropy-based utilities (AIC, BIC) to compute the optimal number of components are also included.

tools This module provides general purpose tools such as an implementation of the mode finding algorithm discussed in section 5.2 or a method to compute the cross-correlation of MoGs or MoPGs.

mopg In this module the the GaussianMixtureBase class is extended to obtain the Projected Gaussian components, consequently allowing the use of MoPG distributions. In this module is also include a specialization of all algorithms from the train and tools modules in order to deal with MoPG. This module contains all necessary structures needed to handle the projection and back-projection from dual quaternions to the $SE(3)$ tangent space.

visualization This is the only non-header module and extends the PCL library visualization functionalities in order to provide a visual UI to represent MoG or MoPG distributions.

6.3.1 Related Results

The GML library along with some of the modelling tools presented in this paper has been exploited in some works related to the statistical analysis of angular data through the Gaussian Mixture Model (GMM). Among them, an MoG regression technique for EMG signals proposed by Michieletto et al. in [24] is presented in the next paragraph.

GMM-based Signals Regression In [24] Michieletto et al. explored the use of a Gaussian Mixture Model (GMM) for the estimation of single-joint angle, and in particular the angular aperture of the knee (see figure 6.4). EMG signals from eight leg muscles and the knee joint angle were acquired during a kick task from three different subjects.

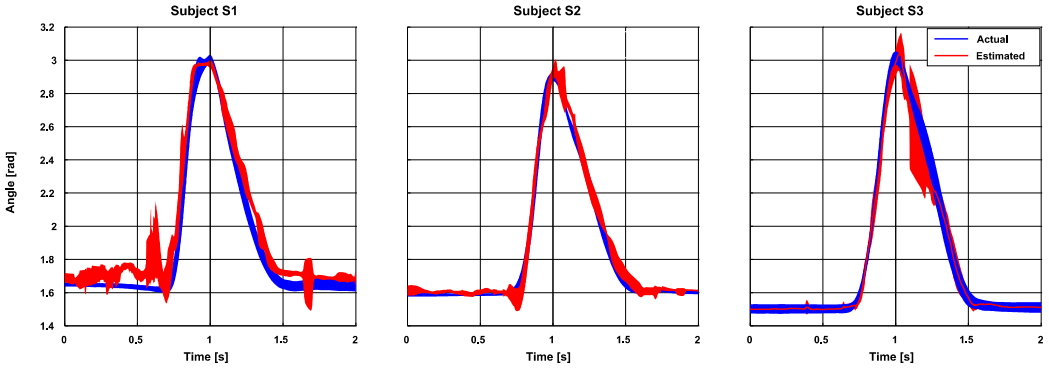

FIGURE 6.4: Actual (in blue) and estimated (in red) angular aperture for the three subjects. Mean and standard deviation are reported (solid line and bounds, respectively). Vertical black line corresponds to the moment with maximum angular aperture during each kick.

A GMM was trained in order to model the angle variation with respect to the EMG signal. The GMM was validated on new unseen data and the classification performances were compared with respect to the number of EMG channels and the number of collected trials used during the training phase. A Gaussian Mixture Regression (GMR) technique was then used to retrieve the data from the trained model. This approach enables an autonomous extraction of the constraints encoded in EMG signals, while still maintaining an appropriate generalization. Modeling input dataset in terms of Mixture of Gaussians (MoG) distributions requires only a reduced number of parameters to be kept, resulting in lightweight models. A GMM/GMR framework was chosen because it usually requires less training data to achieve good results and provides a faster regression in respect to other techniques, like Neural Networks (NN). Due to its characteristics, the GMM/GMR framework is particularly suitable for robotic applications and has been widely adopted in related state of art methods such as Robot Learning from Demonstration (RLfD).

Bibliography

[1] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. *ICRA*, pages 1817–1824, May 2011. doi: 10.1109/ICRA.2011.5980382. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980382.

[2] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. *ICRA*, 2014.

[3] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[4] MA Carreira-Perpinan. Mode-finding for mixtures of gaussian distributions. *Pattern Analysis and Machine Learning*, pages 1–23, 2000. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=888716.

[5] AP Dempster, NM Laird, and DB Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977. URL http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstracthttp://www.jstor.org/stable/10.2307/2984875.

[6] Wendelin Feiten, Pradeep Atwal, Robert Eidenberger, and Thilo Grundmann. 6d pose uncertainty in robotic perception. In Torsten Kröger and FriedrichM. Wahl, editors, *Advances in Robotics Research*, pages 89–98. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-01212-9. doi: 10.1007/978-3-642-01213-6_9. URL http://dx.doi.org/10.1007/978-3-642-01213-6_9.

[7] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[8] Mauro Antonello, Sukhan Lee, Ahmed Naguib, and Emanuele Menegatti. Object recognition and pose estimation by means of cross-correlation of mixture of projected gaussian. *IAS*, 2014.

[9] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Sparse distance learning for object recognition combining rgb and depth information. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4007–4013. IEEE, 2011.

[10] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Depth kernel descriptors for object recognition. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 821–826. IEEE, 2011.

[11] Matteo Munaro, Filippo Basso, and Emanuele Menegatti. Tracking people within groups with rgb-d data. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2101–2107. IEEE, 2012.

[12] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *In the 12th International Symposium on Experimental Robotics (ISER*. Citeseer, 2010.

[13] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[14] Nikolas Engelhard, Felix Endres, Jürgen Hess, Jürgen Sturm, and Wolfram Burgard. Real-time 3d visual slam with a hand-held rgb-d camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*, volume 180, 2011.

[15] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[16] Corina Kim Schindhelm. Evaluating slam approaches for microsoft kinect. In *ICWMC 2012, The Eighth International Conference on Wireless and Mobile Communications*, pages 402–407, 2012.

[17] Thibaut Weise, Thomas Wismer, Bastian Leibe, and Luc Van Gool. In-hand scanning with online loop closure. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1630–1637. IEEE, 2009.

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

[19] Muriel Lang and Wendelin Feiten. Mpg-fast forward reasoning on 6 dof pose uncertainty. *Proceedings of ROBOTIK*, (3):273–278, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6309521.

[20] PM Engel and MR Heinen. Incremental learning of multivariate gaussian mixture models. *Advances in Artificial Intelligence*, pages 82–91, 2011. URL http://link.springer.com/chapter/10.1007/978-3-642-16138-4_9.

[21] Sukhan Lee, Muhammad Ilyas, Kim Jaewoong, and Ahmed Naguib. Evidence filtering in a sequence of images for recognition. In *2012 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–8. IEEE, 2012. ISBN 978-1-4673-4559-0. doi: 10.1109/AIPR.2012.6528203. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6528203.

[22] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[23] Stefano Michieletto, Alberto Rizzi, and Emanuele Menegatti. Robot learning by observing humans activities and modeling failures. In *IROS workshops: Cognitive Robotics Systems (CRS2013)*. IEEE, Nov 2013.

[24] Stefano Michieletto, Luca Tonin, Mauro Antonello, Roberto Bortoletto, Fabiola Spolaor, Enrico Pagello, and Emanuele Menegatti. Gmm-based single-joint angle estimation using emg signals. *IAS*, 2014.