Master research Internship

Bibliographic report

Hyperion

Robot localization by machine learning and vision algorithms

Domain : Machine Learning - Artificial Intelligence - Computer Vision

Author:
Andreea-Oana Petac

Supervisors:

Cédric Buche[1]
Fabrice Harrouet[1]
Panagiotis Papadakis[2]
Mihai Polceanu[1]

1. IHSEV - Lab-STICC - France
2. IMT Atlantique - France

23 January 2017

Abstract

The topic of my research project is the study of various methods for image-based object detection and localization. As an application to this broad area, it is proposed that the algorithms be tested on NAO robots, in the context of an international artificial intelligence contest. The robots must be able to assess their own position, their teammates' as well as their enemies'. In addition to this, the robots should be able to identify the ball and various landmarks within the field. The research question resides in whether the objects and robots within the field can be detected and localized, taking into consideration all the encountered constraints. Therefore, the purpose is to find the best possible solution with the help of machine learning and computer vision algorithms. In this state-of-the-art, different methods from the machine learning and computer vision areas are presented and compared.

Contents

1 Introduction

1.1 Context

The present bibliographical study has been conducted as a preparation for the internship that I will conduct at the Centre Européen de Réalité Virtuelle (CERV). This internship is in collaboration with the RoboCanes team [Masterjohn et al., 2015] from the University of Miami (UM).

1.2 Problem specification

NAO robot The NAO robot represents the platform selected for our research. It is an autonomous, programmable, humanoid robot produced by the French company SoftBank Robotics SAS (the former Aldebaran Robotics). NAO is 58 cm tall and weights 4.3 kg. It has a built-in Linux based operating system (NAOqi OS [Naoqi, 2016]), making it a completely programmable and interactive robot. Starting from 2007, NAOs have been chosen to be used in the Robot Soccer World Cup (RoboCup), an annual international robot soccer competition. Our main focus is the RoboCup SPL (Standard Platform League), whose aim is to encourage researchers and students to work on robotics and AI challenges, developing more and more efficient approaches in oder to solve the problems faced by autonomous robots [Kitano et al., 1997]. For example, the robot's balance when moving.

SPL contest A match within RoboCup consists of a 5 versus 5 robot soccer game. During a typical SPL match, the robots recognize a number of different objects situated not only within the football field but also outside of it. Apart from the other NAOs, humans may also be present on the field, for example the referees. As a result, the robot must know what happens in its vicinity and must see the other objects within the field in order to be able to localize itself with respect to the other objects. The robot will utilize the detection in order to do the localization. However, the robot is not interested in where it is located from the global point of view, but rather it is required to determine its position with respect to surrounding objects such as people, obstacles, etc.

Constraints Within this context, some constraints must be taken into consideration. Because there are various hardware and software specific issues that need to be addressed within this project, we need to identify the constraints so that the encountered limitations can be overcome. To start with, the latest version of NAO robots have a **1.6 GHz CPU**, which raises the challenge of developping the algorithms with the lowest possible computational cost [Niemüller et al., 2011].

It must also be taken into consideration the fact that the robot needs to split its **limited processing power** (of an x86 AMD GEODE 500 MHz) between various tasks such as vision, localization, behavior and motion control and at the same time while remaining highly responsive enough to play soccer [Khandelwal et al., 2010]. Thus, the **detection process should not take more than 20 ms/cycle**.

Each NAO robot (see Figure 1) is equipped with 2 cameras located along the center of its face at different angular offsets with a **maximum resolution of 1280 x 720 pixels**. Moreover, during the game, **the robots are constantly moving**. Due to the initial manual camera calibration, every unexpected change in the camera settings requires a readjustment of the stable conditions. As a result, the NAO vision system is **not very robust to illumination variations**, that may have a strong influence on the performance of the algorithms.

Another constraint is easily drawn up by taking into consideration that **the code runs on board the robot**. Therefore, it is advised to have a static training dataset, meaning that the learning should be done exclusively offline.

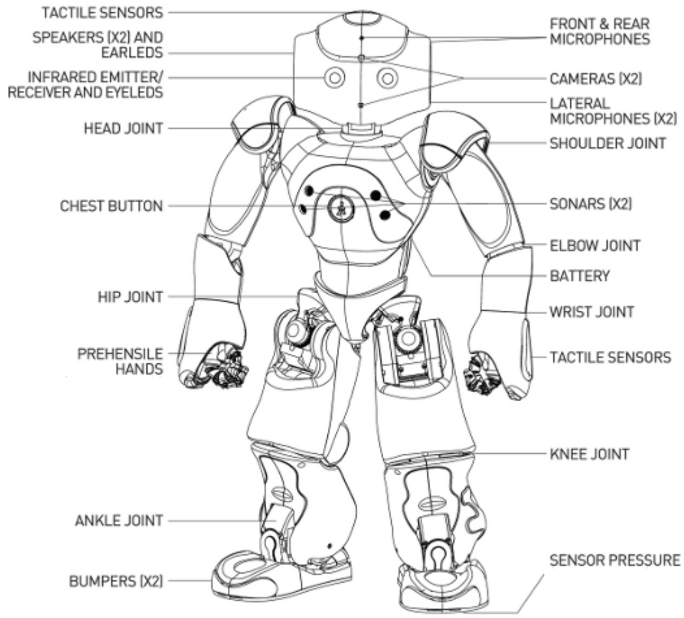


Figure 1: Overview of NAO's hardware parts [Naoqi, 2016]

We can deduce that the objective of the project is to analyze the various methods for object detection and localization, and to implement them within the RoboCup contest. Among these methods, and after testing and optimization, we will find the best possible solution that respects all the constraints presented above.

1.3 Report outline

The research question is focused on the detection and localization of robots and landmarks within the field. It is crucial that the constraints be taken into consideration. Thus, a good balance should be found between good results and low computational cost.

In order to localize itself, the robot must first detect the objects and landmarks from its point of view. Because there is a high number of inputs, it is important to be chosen the most discriminative ones within the dataset in order for the results to be accurate. We need to use the feature extraction because the classification techniques may need extracted values. Briefly, through feature extraction it is understood a method that establishes accurate combinations of traits. This will end by an improvement related to the results. The extracted features will be used in combination with the classification methods. Next, by using a learning algorithm, an object detection and localization could be done.

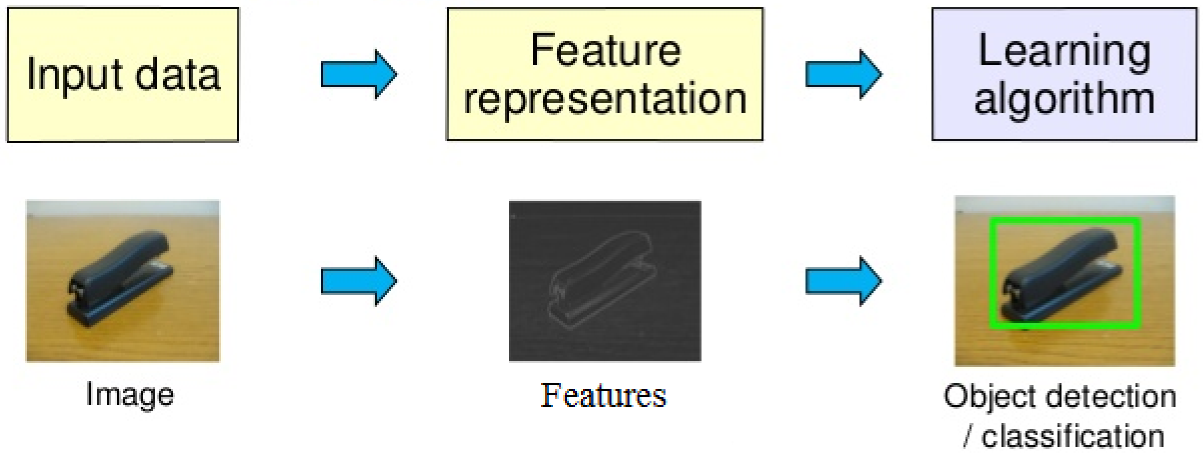


Figure 2: Overview of the learning process by making use of feature extraction

2 State of the Art

In order to achieve our goal, we first have to train a model to find features, namely specific structures in the image such as points, edges or objects within the football field. The action of extracting the features amounts to reducing the total amount of data needed to characterize a large dataset.

2.1 The learning process

Within the following diagram (Figure 3) there are presented the processes of training and prediction. In the example shown below, a picture serves as input to machine learning models. Each machine learning model makes use of an algorithm, depending also on the dataset. Assuming that there will be a high number of inputs, it is important to extract those that have distinctive characteristics.

The purpose of a feature extractor during the training phase is to convert each input value to a set of features. Within these sets of features, the essential information about each input is grabbed that is going to be used to classify it. Next, pairs formed by feature sets and their corresponding labels are provided to the machine learning algorithm, a model being then generated.

As far as machine learning is concerned, it is very important for the right features to be chosen and the right way to represent them, this having a major impact on the ability of the learning method to extract a good model.

Regarding the prediction phase, the trained model is used in order to predict labels of unseen inputs. By using the same feature extractor algorithm, the unseen inputs are converted to feature sets. After these feature sets pass through a classifier model, the predicted labels are generated.

2.2 Feature extraction methods

The extraction of the features is done mainly because various extracted values through the classification techniques may enhance their performance. By performing this action, there are several

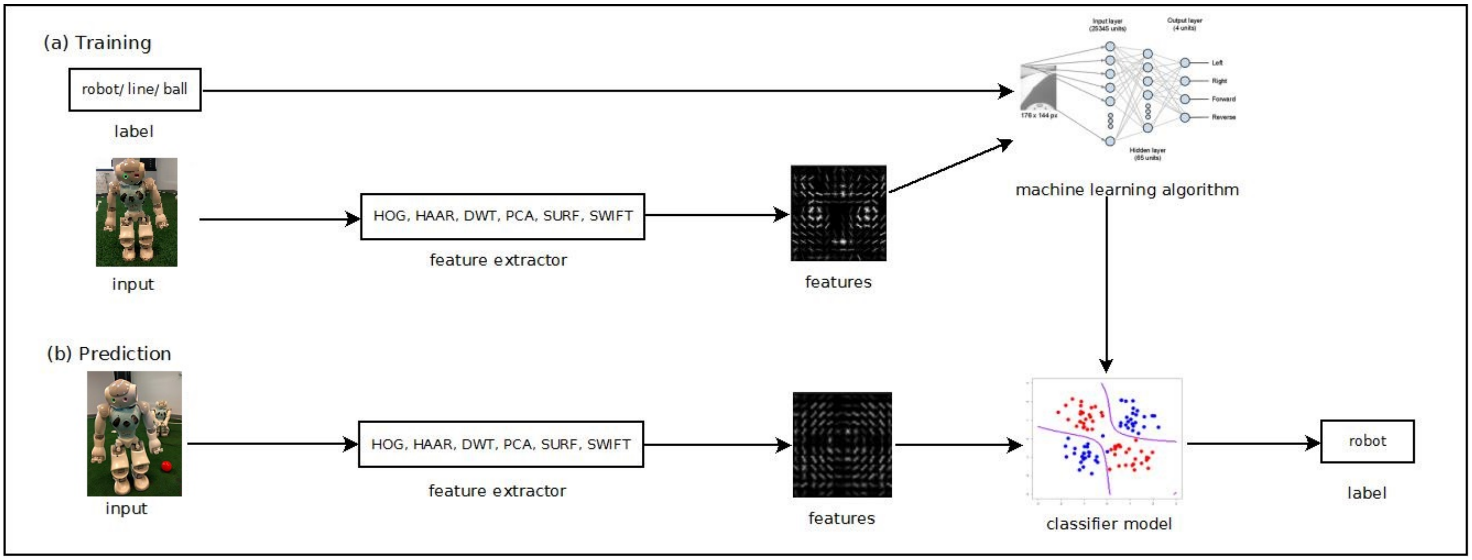


Figure 3: Learning diagram (adapted from [Bird et al., 2009])

advantages such as invariance provided to transformations, noise filtering and a capture of the discriminant.

2.2.1 Histogram of oriented gradients

This is one of the most popular techniques to retrieve shapes from an image. The histogram of oriented gradients (HOG) extracts features within all locations in the image, or within the area of interest. This feature extraction can be used in order to detect shapes of any kind [Dalal and Triggs, 2005]. For example, within some of the classic examples are included the circle detection [Skibbe and Reisert, 2012] and polygon detection [Zeng and Ma, 2010].

By making use of a kernel, it takes subregions from the image, checking then the orientation gradient, namely the slope. After adding it into a bin, it goes further to another region repeating the process. It will stop when the whole image has been sampled. Then it goes into the bin in order to determine the edges [Zhu et al., 2006]. In order to capture the shape of the structures within the region, the algorithm divides the image into small pixels cells and blocks of cells.

Even though HOG does not provide great robustness regarding neither the motion nor the lighting conditions, it has a relatively good performance for various classes of objects. Moreover, by using HOG in accordance to our constraints enunciated above, real time detection can be performed.

2.2.2 Haar-like features

This type of features are made of a class of features that are determined by the difference of the unselected feature region and its subregion. Thanks to the straightforward calculation within an integral image, these features are considered to be very efficient.

By using the values of change in contrast between adjacent groups of pixels, it can be determined the relative light and the dark areas. A Haar-like feature is formed by two or three adjacent groups.

These features can easily be scaled by changing the size of the pixel group, helping to detect objects of several dimensions. [Wilson and Fernandez, 2006] Figure 4 gives some examples of Haar features. Very important to take into consideration is the fact that this type of features can be conveniently rescaled [Lienhart et al., 2003].

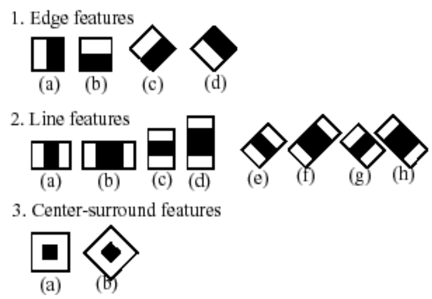


Figure 4: Examples of Haar features (adapted from [Wilson and Fernandez, 2006])

2.2.3 Discrete wavelet transformation

Within this method, a signal is decomposed into several wavelets, namely a combination of wavelets [Chaovalit et al., 2011]. There are various types of wavelets transformations, such as the Harr wavelet or the Daubechies wavelet. DWT can be broken into two types: 1-dimensional (1D) and 2-dimensional (2D). Figure 5 shows a representation of a DWT 1D which has 2 extracted components: approximation and detail [Yaji et al., 2012]. The main advantage obtained by using this method is that it captures the location in time [Mohan and Kumar, 2013].

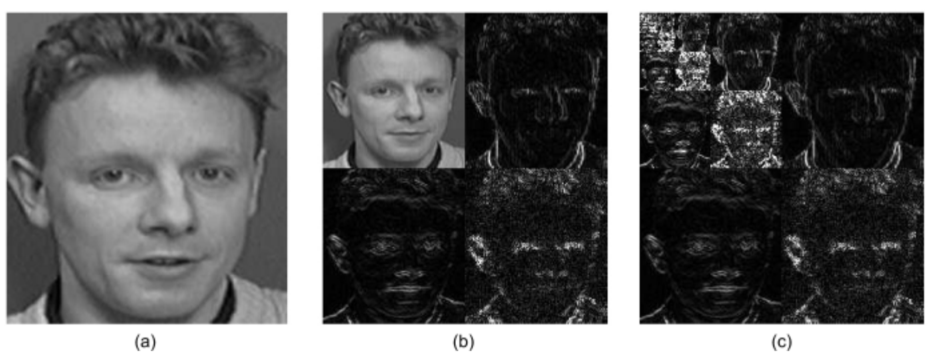
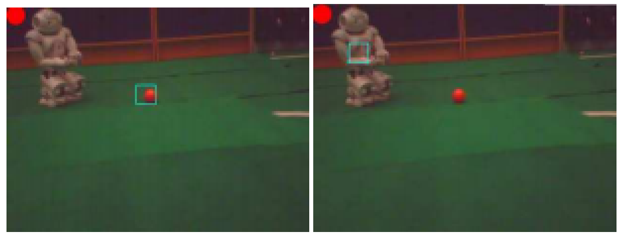


Figure 5: Example of DWT (from [Nicholl et al., 2010])

2.2.4 Object detection by regression

Object detection by regression (ODR) detects the position of an object within an image. The use of this algorithm is common within the RoboCup environment [Visser, 2016]. It works by creating a statistical model of the relation between an image and the position of a given object in that image [Brandão et al., 2012]. This statistical model permits image sampling, all of this being done either offline or online. Its output is represented by the position of the object and a weight of confidence that the object exists (Figure 6). It is known from [Brandão et al., 2012] that ODR performs very well within a precomputed environment and detection of objects.



(a) Ball detection (b) Robot detection

Figure 6: Examples of ball and robot detection using ODR - square indicates the location and detection of a) the ball and b) the robot (from [Brandão et al., 2012])

2.2.5 Speeded-up robust feature

Speeded-up robust feature detector (SURF) is a scale and rotation invariant interest point detector and descriptor [Bay et al., 2006]. This method can be used not only to locate and recognize objects, but also to track them [Shuo et al., 2012]. The original image is converted to a set of coordinates with a technique called the multi-resolution pyramid technique. A new image results with the same size but with reduced decreased bandwidth. A blurring effect is then created, provided that the interest point are scale invariant.

2.2.6 Scale invariant feature rransform

Scale Invariant Feature Transform (SIFT) represents a method to extract feature descriptors that are invariant to rotation, scaling, quality and lighting [Lowe, 2004]. It is represented by a chosen area within the image, which is called a keypoint [Lowe, 2004], combined with a descriptor. The SIFT detector takes care of extracting the keypoints, while the descriptors are computed using the SIFT descriptor.

2.2.7 Principal component analysis

Principal component analysis (PCA) is a way to identify patterns in data and to express data in order to highlight similarities and differences between them. Because the patterns in high-dimensional data are hard to find and as we cannot view the data, PCA is a powerful tool for analyzing them. Another advantage of the PCA method is that once found the patterns, we can reduce the number of dimensions without losing much information. A great advantage obtained by using this method is that it does not make use of large computations.

2.3 Feature extraction comparison

We have identified various extraction techniques in the previous sections that we summed up in Table 1. Based on their strengths and weaknesses, the following table shows them taking into consideration some of the constraints presented above. Because datasets are known to be large regarding the number of the measured variables within each one of them, computational complexity has a very big role. Moreover, we added the parameters within the table. An implementation can also be judged on how many parameters it makes use of, because generally, with more parameters involved comes a higher computational cost.

Feature	Robust to:			6. Offline learning	Computational complexity	Parameters
	3. Motion	4. Camera quality	5. Lighting conditions			
HOG	X [Churchill and Fedor,2014]	X [Churchill and Fedor,2014]	X [Churchill and Fedor,2014]	✓ [Kaaniche and Brémond,2009]	$O(4n^2)$	8 [Kim and Cho,2014]
HAAR	✓ [Lienhart and Maydt,2002]	X [Lienhart and Maydt,2002]	✓ [Gong et al.,2009]	✓ [Gong et al.,2009]	$O\left(14/3n^2\right)$ [Porwik and Lisowska,2004]	2 [Chun-Lin,2010]
DWT	X [Ahmad et al.,2010]	✓ [Ahmad et al.,2010]	✓ [Ahmad et al.,2010]	✓ [Ahmad et al.,2010]	$2(1-1/n)$ [Shukla and Tiwari,2013]	2 [Shukla and Tiwari,2013]
ODR	✓ [Brandão et al.,2012]	✓ [Brandão et al.,2012]	✓ [Brandão et al.,2012]	✓ [Brandão et al.,2012]	$O(n)$ [Brandao et al.,2010]	4 [Brandão et al.,2012]
SURF	✓ [Bay et al.,2008]	✓ [Bay et al.,2006]	✓ [Bay et al.,2006]	✓ [Sergieh et al.,2012]	$O(n^2)$ [Oyallon and Rabin,2015]	4 [Pedersen,2011]
SIFT	✓ [Lowe,2004]	✓ [Lowe,2004]	✓ [Lowe,2004]	✓ [Lowe,2005]	$> O(n)$ [Vinukonda,2011]	9 [Vinukonda,2011]
PCA	✓ [De la Torre and Black,2001]	✓ [De la Torre and Black,2001]	X [Ramamoorthi,2002]	✓ [Boutsidis et al.,2015]	$O(p^3+p^2n)$ [Aspremont et al.,2008]	$2p+p^2/2$ [Natrella,2010]

Table 1: Feature extraction comparison - check mark (it meets the constraint) and cross (it does not satisfy the constraint)

By using the HOG method, better contour of the object can be acquired. On the other hand, if Haar-like features are used, then the regions with a bigger difference in shading will be described better. It can easily be observed that the HOG extractor is not performing as the Haar-like features, mainly because HOG detects the object's shape rather than static objects. Since the robots are moving within the SPL game, it would be extremely hard to get a static shape. DWT is a suitable chosen method for feature extraction because of its low computational complexity. ODR is a robust and computationally efficient algorithm. According to [Panchal et al., 2013], SIFT is said to be more robust than SURF but in the same time, it is slower.
To conclude this section, we will take into consideration the qualities and disadvantages of the studied feature extraction algorithms in order to combine them with the appropriate classification method.

2.4 Classification and localization methods

Having in mind the learning diagram presented above (Figure 3), we will now take the classification methods into discussion. The classification methods take the extracted features as inputs. Within machine learning area, the goal of an algorithm is the learning of a set of rules that have the ability to describe the set of inputs and outputs. As humans, we know those rules, but we are not able do describe them mathematically with enough concreteness. We will use the detection in order to perform the localization.

2.4.1 Support vector machines

Support vector machines (SVM) is a statistic model that separates two data sets [Burges, 1998]. The classes are divided through an optimal separating hyperplane (OSH). The limits of these classes of data and the OSH are called support vectors.

Intuitively, for a set of points divided into two classes, the SVM method finds on one side the hyperplane that separates the highest possible fraction of points that belong to the same class, and on the other side it maximizes the distance between the classes and hyperplane. For a binary classification problem (see Figure 7), the aim is to separate the two classes using a function that is obtained from the available examples.

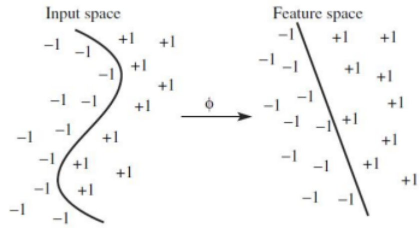


Figure 7: Example of SVM separation (from [Ivanciuc, 2007])

By using SVM, the user can avoid over-fitting due to its regularization parameter. The real help the SVM comes in is when there is randomly distributed data, taking into account a good training.

2.4.2 k-Nearest Neighbour

The k-nearest neighbour (kNN) algorithm [Murty and Devi, 2011] is also a classification algorithm and involves the finding of not just one, but a number of neighbors k. All the inputs are classified, meaning that belong to a class.

As an example, for the face recognition area, all images belonging to the same person forms a class. Then, the person looked for, will belong to the majority class among the k neighbors. For example, if you take the 7 nearest neighbors and 5 of them belong to a class, and the remaining 2 to another class, it can be deduced that the person sought will belong to the first class.

The value of k can be determined from experiments, being chosen the value that gives the least number of errors regarding the classification. For large data sets, to reduce classification errors, a higher value of k can be chosen. kNN is known for its really good performance on basic recognition problems, such as face recognition using the ORL data set. Though, kNN uses its data for classification only, rather than learning from the training part.

2.4.3 AdaBoost

The method Adaptive boosting (AdaBoost) [Guo and Zhang, 2001] is based on the idea of creating a predictor with high degree of accuracy by combining multiple "weak" classification functions. AdaBoost is an adaptive algorithm that combines a sequence of classifiers for which the weights are updated dynamically depending on the errors that occur prior to learning. AdaBoost is a classifier with a high margin of error. Within this method, for training the new iteration it uses the results from the previous ones, in order to improve the performance.

This classifier is known for being simple to implement, improving the accuracy of the classification. It performs the selection of features that result in a simple classifier. A major difference between SVM and AdaBoost is that the latter selects only those features that are known to elevate the predictive power of the model. Through this, it reduces the dimensionality and improves the execution time as the irrelevant features do not need to be computed.

2.4.4 Simultaneous localization and mapping

Self-localization and localization of other robots and objects is an important aspect, especially regarding the RoboCup SPL contest. It is very crucial for the robot to know where he is placed within the field so as to finish doing the next decision making tasks such as passing the ball or plan its future path. The first fact that we have to take into consideration is that the robot does not observe its surroundings in a completely way at a given time, this not being enough to determine its precise position. Moreover, the landmarks within the field can be often vague.

The SLAM method discusses the possibility for a robot to be placed in an unknown environment [Durrant-Whyte and Bailey, 2006], having to move around to collect data of landmarks within the environment. This should be done by using its sensors and by recording its position with respect to that created map. This method is known for working well with a small number of features and various landmarks. Though, it performs slow within high dimensional maps [Aulinas et al., 2008].

We will use this just for localization. Thus, we could also take into consideration the method described in [Coaguila et al., 2016] which uses HOG with stuctural SVM. Afterwards, by using the method proposed in [Kazemi and Sullivan, 2014] we can localize the landmarks in real time.

2.4.5 Particle filter

Particle filter represents a popular choice regarding the localization within the area of robotics [Thrun, 2002]. This method implies that the probability of the robot to be somewhere taking into consideration the observations, i.e. what it is detected within its environment. The purpose of using this method within our context is to track the actual position of the robot, knowing that its precise location is not known by the algorithm.

2.5 Classification and localization methods comparison

For each particular method there are situations for which it is particularly well suited, and others where it performs badly compared to the best that can be done with that data. We have attempted to characterize appropriate situations in our discussions of each of the respective methods. However, it is seldom known in advance which procedure will perform best or even well for any given problem. We took into consideration methods both for detection and localization. So, as characteristics that describe and classify the alg we choose as it follows. To start with, the handling of missing data was taken into consideration due to the fact there are a high number of missing values within the observations. One is seldom able to find a complete observation. Because the variables within the dataset are usually measured on various scales, thus different ones, it is important for the algorithm to be scalable from the computational point of view.

The ability to extract linear combinations of features is important because the entire model can be completely represented by a simple two-dimensional graphic (binary tree) that is easily visualized. Usually only a small fraction of the large number of predictor variables that have been included in the analysis are actually relevant to prediction. Also, unlike many applications such as pattern recognition, there is seldom reliable domain knowledge to help create especially

Algorithm	Handling of missing values	Computational scalability (large n)	Ability to extract linear combinations of features	Predictive power
SVM	X [Hastie et al., 2003]	X [Hastie et al., 2003]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]
k-Nearest Neighbours	X [Hastie et al., 2003]	X [Hastie et al., 2003]	fair [Hastie et al., 2003]	✓[Hastie et al., 2003]
AdaBoost	✓[Hastie et al., 2003]	✓[Cooper and Reyzin,]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]
SLAM	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]
Particle filter	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]	✓[Hastie et al., 2003]

Table 2: Classification methods comparison (adapted from [Hastie et al., 2003]) - fair (provides medium fulfillment), check mark (it meets the constraint) and cross (it does not satisfy the constraint)

relevant features and filter out the irrelevant ones, the inclusion of which dramatically degrades the performance of many methods.

3 Conclusion

Summarizing, the paper's focus is on the detection and localization of objects within the field. Taking into consideration all the literature that we studied so far, we will now take some points into discussion.

3.1 Synthesis

The work presented focuses on the problem of detection and localization of the humanoid robot NAO within the RoboCup SPL match. Section 1 introduces the context by presenting the problem and the challenges faced. It also exposes the software and hardware constraints. Section 2 reports the state of the art of the domain. It starts with a description of the learning process. It also carries out a brief description of each method we used both for feature extraction but also for classification. Furthermore, these methods are discussed with the help of the two tables within which they are compared based on existing literature review. Lastly, section 3 concludes the paper and presents the following steps of the internship.

3.2 Research directions

The presented methods above represent the basis from where we will start implementing, testing, and optimizing the detection and localization of objects within the field (ball, field, robot). The overall goal of the internship is to detect and localize not only robots within our own team and from the opponent's team, but also object (ball, landmarks etc) within the football field.

The research question is placed within the context of whether the objects and robots within the field can be detected and localized, taking into consideration the constraints. Therefore, the purpose is to find the best possible solution with the help of machine learning and computer vision algorithms. In this state-of-the-art, different methods from the machine learning and computer vision areas are presented and compared.

The analyzed methods are made up of steps in order to perform the learning process. Firstly, the inputs are obtained and the detection of objects is done. Afterwards, the object that was detected is characterized and classified.

As future advancements of this master thesis we propose the implementation and testing under various conditions of different combinations of feature extractors with classification methods. The results from the two comparison tables show that HOG+SVM is a good approach to start tackling the task. We could also make use of the combination between AdaBoost (for doing the tracking) and SURF (to extract and match the features). Other good combinations would be ODR and PCA [Brandão et al., 2012], SURF and PCA [Valenzuela et al., 2012].

3.3 Planning of the internship

The internship will start with the implementation and testing of the proposed combinations of methods. Then, a solution that both respects all the constraints and has a satisfying recognition rate will be proposed.

Meanwhile, I am going to prepare an article to submit to the RoboCup International Symposium 2017. Moreover, testing will be executed over a period of three months leaded by my main coordinator within the RoboCanes team at University of Miami, followed by more optimization. The plan is described in the figure 8.

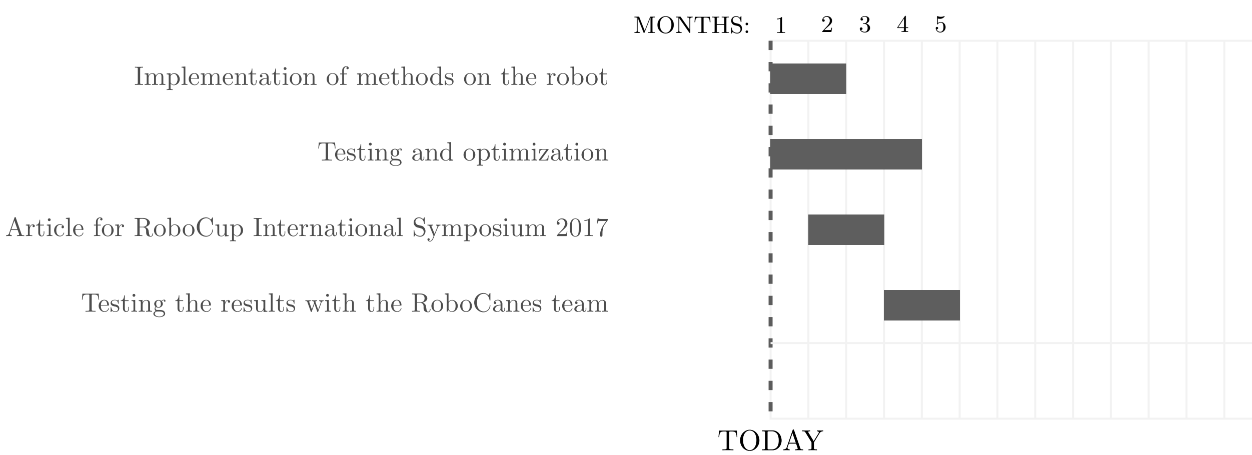


Figure 8: Planning for the internship

References

[Ahmad et al., 2010] Ahmad, W., Hacıhabiboglu, H., and Kondoz, A. M. (2010). Discrete wavelet transform based shift-invariant analysis scheme for transient sound signals. In *13th Int. Conference on Digital Audio Effects*.

[Aulinas et al., 2008] Aulinas, J., Petillot, Y. R., Salvi, J., and Lladó, X. (2008). The slam problem: a survey. In *CCIA*, pages 363–371. Citeseer.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer.

[Bird et al., 2009] Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. " O'Reilly Media, Inc.".

[Boutsidis et al., 2015] Boutsidis, C., Garber, D., Karnin, Z., and Liberty, E. (2015). Online principal components analysis. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 887–901. Society for Industrial and Applied Mathematics.

[Brandão et al., 2012] Brandão, S., Veloso, M., and Costeira, J. P. (2012). *Fast Object Detection by Regression in Robot Soccer*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Burges, 1998] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.

[Chaovalit et al., 2011] Chaovalit, P., Gangopadhyay, A., Karabatis, G., and Chen, Z. (2011). Discrete wavelet transform-based time series analysis and mining. *ACM Computing Surveys (CSUR)*, 43(2):6.

[Chun-Lin, 2010] Chun-Lin, L. (2010). A tutorial of the wavelet transform. *NTUEE, Taiwan*.

[Churchill and Fedor, 2014] Churchill, M. and Fedor, A. (2014). Histogram of oriented gradients for detection of multiple scene properties.

[Coaguila et al., 2016] Coaguila, R., Sukthankar, G. R., and Sukthankar, R. (2016). Selecting vantage points for an autonomous quadcopter videographer. In *FLAIRS Conference*, pages 386–391.

[Cooper and Reyzin,] Cooper, J. and Reyzin, L. Improved algorithms for distributed boosting.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE.

[dAspremont et al., 2008] dAspremont, A., Bach, F., and Ghaoui, L. E. (2008). Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9(Jul):1269–1294.

[De la Torre and Black, 2001] De la Torre, F. and Black, M. J. (2001). Robust principal component analysis for computer vision. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 362–369. IEEE.

[Dray and Josse, 2015] Dray, S. and Josse, J. (2015). Principal component analysis with missing values: a comparative survey of methods. *Plant Ecology*, 216(5):657–667.

[Durrant-Whyte and Bailey, 2006] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localisation and mapping (slam): Part i the essential algorithms.

[Gong et al., 2009] Gong, H., Li, C., Dai, P., and Xie, Y. (2009). Object tracking based on the combination of learning and cascade particle filter. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 978–983. IEEE.

[Guo and Zhang, 2001] Guo, G.-D. and Zhang, H.-J. (2001). Boosting for fast face recognition. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*, pages 96–100. IEEE.

[Hastie et al., 2003] Hastie, T., Tibshirani, R., and Friedman, J. (2003). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer; 1st ed. 2001. Corr. 3rd printing edition (July 30, 2003).

[Ivanciuc, 2007] Ivanciuc, O. (2007). Applications of support vector machines in chemistry. *Reviews in computational chemistry*, 23:291.

[Kaaniche and Brémond, 2009] Kaaniche, M. B. and Brémond, F. (2009). Tracking hog descriptors for gesture recognition. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 140–145. IEEE.

[Kazemi and Sullivan, 2014] Kazemi, V. and Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874.

[Khandelwal et al., 2010] Khandelwal, P., Hausknecht, M., Lee, J., Tian, A., and Stone, P. (2010). Vision calibration and processing on a humanoid soccer robot. In *The Fifth Workshop on Humanoid Soccer Robots*.

[Kim and Cho, 2014] Kim, S. and Cho, K. (2014). Fast calculation of histogram of oriented gradient feature by removing redundancy in overlapping block. *J. Inf. Sci. Eng.*, 30(6):1719–1731.

[Kitano et al., 1997] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., and Matsubara, H. (1997). Robocup: A challenge problem for ai. *AI magazine*, 18(1):73.

[Lienhart et al., 2003] Lienhart, R., Kuranov, A., and Pisarevsky, V. (2003). Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *Joint Pattern Recognition Symposium*, pages 297–304. Springer.

[Lienhart and Maydt, 2002] Lienhart, R. and Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900. IEEE.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[Lowe, 2005] Lowe, D. G. (2005). Demo software: Sift keypoint detector.

[Masterjohn et al., 2015] Masterjohn, J. G., Polceanu, M., Jarrett, J., Seekircher, A., Buche, C., and Visser, U. (2015). Regression and mental models for decision making on robotic biped goalkeepers. In *Robot Soccer World Cup*, pages 177–189. Springer.

[Mohan and Kumar, 2013] Mohan, S. and Kumar, S. S. (2013). *Proceedings of the Fourth International Conference on Signal and Image Processing 2012 (ICSIP 2012)*, volume 1. Springer Science & Business Media.

[Murty and Devi, 2011] Murty, M. N. and Devi, V. S. (2011). *Pattern recognition: An algorithmic approach*. Springer Science & Business Media.

[Naoqi, 2016] Naoqi (2016). Aldebaran documentation, http://doc.aldebaran.com/2-1/naoqi/. [Online; accessed 11-January-2016].

[Natrella, 2010] Natrella, M. (2010). Nist/sematech e-handbook of statistical methods.

[Nicholl et al., 2010] Nicholl, P., Ahmad, A., and Amira, A. (2010). Optimal discrete wavelet transform (dwt) features for face recognition. In *Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on*, pages 132–135. IEEE.

[Niemüller et al., 2011] Niemüller, T., Ferrein, A., Eckel, G., Pirro, D., Podbregar, P., Kellner, T., Rath, C., and Steinbauer, G. (2011). *Providing Ground-Truth Data for the Nao Robot Platform*, pages 133–144. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Oyallon and Rabin, 2015] Oyallon, E. and Rabin, J. (2015). An analysis of the surf method. *Image Processing On Line*, 5:176–218.

[Panchal et al., 2013] Panchal, P., Panchal, S., and Shah, S. (2013). A comparison of sift and surf. *International Journal of Innovative Research in Computer and Communication Engineering*, 1(2):323–327.

[Pedersen, 2011] Pedersen, J. T. (2011). Surf: Feature detection & description. *Technická zpráva, AARHUS University*.

[Porwik and Lisowska, 2004] Porwik, P. and Lisowska, A. (2004). The haar-wavelet transform in digital image processing: its status and achievements. *Machine graphics and vision*, 13(1/2):79–98.

[Ramamoorthi, 2002] Ramamoorthi, R. (2002). Analytic pca construction for theoretical analysis of lighting variability in images of a lambertian object. *IEEE transactions on pattern analysis and machine intelligence*, 24(10):1322–1333.

[Sergieh et al., 2012] Sergieh, H. M., Egyed-Zsigmond, E., Döller, M., Coquil, D., Pinon, J.-M., and Kosch, H. (2012). Improving surf image matching using supervised learning. In *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*, pages 230–237. IEEE.

[Shukla and Tiwari, 2013] Shukla, K. and Tiwari, A. K. (2013). Filter banks and dwt. In *Efficient Algorithms for Discrete Wavelet Transform*, pages 21–36. Springer.

[Shuo et al., 2012] Shuo, H., Na, W., and Huajun, S. (2012). Object tracking method based on surf. *AASRI Procedia*, 3:351–356.

[Skibbe and Reisert, 2012] Skibbe, H. and Reisert, M. (2012). Circular fourier-hog features for rotation invariant object detection in biomedical images. In *ISBI*, pages 450–453.

[Thrun, 2002] Thrun, S. (2002). Particle filters in robotics. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 511–518. Morgan Kaufmann Publishers Inc.

[Valenzuela et al., 2012] Valenzuela, R. E. G., Schwartz, W. R., and Pedrini, H. (2012). Dimensionality reduction through pca over sift and surf descriptors. In *Cybernetic Intelligent Systems (CIS), 2012 IEEE 11th International Conference on*, pages 58–63. IEEE.

[Vinukonda, 2011] Vinukonda, P. (2011). *A study of the scale-invariant feature transform on a parallel pipeline*. PhD thesis, Louisiana State University.

[Visser, 2016] Visser, U. (2016). 20 years of robocup. *KI - Künstliche Intelligenz*, 30(3):217–220.

[Wilson and Fernandez, 2006] Wilson, P. I. and Fernandez, J. (2006). Facial feature detection using haar classifiers. *Journal of Computing Sciences in Colleges*, 21(4):127–133.

[Yaji et al., 2012] Yaji, G. S., Sarkar, S., Manikantan, K., and Ramachandran, S. (2012). Dwt feature extraction based face recognition using intensity mapped unsharp masking and laplacian of gaussian filtering with scalar multiplier. *Procedia Technology*, 6:475–484.

[Zeng and Ma, 2010] Zeng, C. and Ma, H. (2010). Robust head-shoulder detection by pca-based multilevel hog-lbp detector for people counting. In *2010 20th International Conference on Pattern Recognition*, pages 2069–2072.

[Zhu et al., 2006] Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1491–1498. IEEE.