University of Tartu
Faculty of Mathematics and Computer Science
Institute of Computer Science
Computer Science Curriculum

Anastasia Bolotnikova

Melioration of color calibration, goal detection and self-localization systems of NAO humanoid robots

Bachelor's thesis (9 ECTP)

Supervisor: Assoc. Prof. Gholamreza Anbarjafari

Tartu 2015

NAO humnoid robotite värvide kalibreerimise, väravate tuvastamise- ja lokaliseerimissüsteemide arendamine

Lühikokkuvõte:

Selle lõputöö teemaks on autonoomsete robotite jalgpalli tarkvara arendamine.Vaatluse all on teemad nagu värvide kalibreerimine, objetkituvastus ja lokaliseerimine. Uus YUV värviruumi põhine automaatne värvide kalibreerimine on pakutud. Esitatakse detailne kirjeldus automaatse värvide kalibreerimise algoritmi implemenmtreerimisest koos visuaalsete näidetega, mis illustreerivad algoritmi toimimist. Samuti räägitakse täpsemalt muutustest, mis on implementeeritud väravate tuvastamise moodulis ja põhjustest nende muudatuste taga, andes hea ülevaate objekti tuvastamise algoritmi loogikast. Kirjeldatakse hetkel kasutatavat lokaliseerimissüsteemi ja pakutakse välja ning seletatakse lokaliseerimissüsteem parandamise tehnikat.

Võtmesõnad:

NAO robotid, RoboCup SPL võistlus, lokaliseerimissüsteem, automaatne värvide kalibreerimine, objektituvastus, tehisnägemine

Melioration of color calibration, goal detection and self-localization systems of NAO humanoid robots

Abstract:

In this thesis, work regarding to autonomous robot soccer software development is presented. The work covers color calibration, object detection and robot localization topics. Novel YUV color space based method for the automation of color calibration is proposed. Detailed description of automatic color calibration technique implementation is provided along with the visual results illustrating performance of the method. Changes implemented to the goal detection module and motivation behind them are described in detail, providing good overview of the logic of the object recognition algorithm. Utilised localisation system is also described and, finally, the localization system enhancement technique is proposed and explained.

Keywords:

NAO robots, RoboCup SPL competition, self-localization, automatic color calibration, object recognition, computer vision

Contents

Introduction

The platform selected for research which is introduced in the given thesis is an autonomous, programmable humanoid robot NAO produced by the French company Aldebaran Robotics.

NAO is a 58 cm tall programmable humanoid robot with 25 degrees of freedom, i.e. 25 different motors that control the joints of the robot. NAO robot weights 4.3 kg and has a built-in Linux-based operating system (NAOqi OS). Latest version of NAO robots have a CPU Intel Atom 1.6 GHz, which introduces the challenge for programmers to develop the algorithms which would have lowest computational cost possible, so that even CPU of 1.6 GHz can handle the processing tasks. Each NAO robot has two cameras with maximum resolution of 1280x720, two infrared emitters and receivers, nine tactile sensors, sonar rangefinder, eight pressure sensors and four microphones. Development process of NAO robots began in 2004. Since then they have been widely used for research and educational purposes in universities and research centers all over the world. Starting from year 2007, NAO robots have been chosen to serve as robots used in Robot Soccer World Cup (RoboCup) Standard Platform League (SPL), which is an annual international robot soccer competition[rob].

The aim of RoboCup SPL is to encourage researchers and students from different countries to work on robotics and artificial intelligence challenges and develop new and more efficient approaches of solving the problems faced by autonomous robots. As the platform used in competitions of the same type is fixed, it makes the RoboCup to be purely a programming competition.

There are several main challenges of a different type in RoboCup competition, such as robot soccer, rescue robots, self-organizing robots for logistics and robots in human society. In given thesis the main focus is on the robot soccer, as it is the challenge that Tartu university RoboCup team participates in.

Robot soccer challenge is divided into 3 main sub-challenges: Soccer Competition, Any Place Competition and Drop-in Player Competition. Soccer Competition is a 5 versus 5 robots soccer game where a team of 5 robots represents a certain university or country. Any Place Competition is a type of soccer match where robots are not assigned to any specific role in the team which makes it more complicated for a robot to make a decision during the game as the range of possible actions becomes much wider. In a Drop-in Challenge one robot from each university is picked in order to compose a mixed team of robots where each player is running a different software. This type of competitions aims to encourage the development of flexible algorithms that would be perform well even when there is no certain information available on what is the game strategy of other team mates.

The RoboCup SPL team, Philosopher, that represents University of Tartu (UT) in the RoboCup World Championship, has been established in January 2013. Since then significant work has been done in order to get familiar with the concepts of programming for NAO robots and to develop the algorithms that enable NAO robots of Tartu University RoboCup team to play soccer and compete at the world level. In 2014 team Philosopher got qualified for international RoboCup competition which took place in Brazil in summer of 2014. During this competition team Philosopher attended competitions of all 3 main challenges of robot soccer. In the year 2015 team Philosopher got qualified to attend RoboCup competition in China which will take place during the summer of 2015 in the city of Hefei.

Figure 1: NAO robot from RoboCup team Philosopher.

In this thesis the detailed description of improvements intended to provide a better game performance in the upcoming competitions introduced to the NAO color calibration, object detection and localization systems are presented. It is important to note that due to lack of processing unit and also the response time of NAO within the competition various pre-processing applications such as illumination enhancement cannot be employed [DOA10, DAJ08].

In the Chapter 1 of the given thesis the research regarding the automatic color calibration is presented including motivation behind the research, overview of similar research previously conducted in this area, followed by proposed technique with the description of it's implementation and results.

Work regarding white goal detection, motivation for this topic, description of the current goal detection system, description of conducted work and results can be found in Chapter 2.

Localization enhancement related research along with current module description, proposed technique, its implementation and performance results are presented in Chapter 3.

Chapter 1

Automatic color calibration

1.1 Motivation for automation of the color calibration process

In order to distinguish between different objects on the soccer field, first the color information needs to be processed to create so-called "blobs" which are same color areas in the acquired frame that can be used in the further processing stages such as object detection and localization. In order to perform the blob formation process quickly and accurately the color calibration procedure needs to be performed.

Color calibration is a process of defining the main color classes used by NAO robot during the game (e.g. green, orange, white) by assigning the appropriate pixel values to the corresponding classes. As a result of the color calibration procedure the so-called color table or look-up table (LUT) is obtained where all necessary color classes are associated with the corresponding pixel color values. Defined color classes can be used in order to segment the image into similar color areas, where all shades of the same color will be represented by a label of corresponding class. If the process of color calibration has not been done properly, NAO robot will not be able to form blobs correctly and thus will fail to detect objects on the field. Thus it is critical to have an accurate and robust color calibrating system.

Currently the manual point-and-click approach is used in order to perform the color calibration before each game. According to the point-and-click approach a person clicks on the area in the image that should be classified as some particular color and all pixel values similar to the chosen pixel value are classified accordingly. The main disadvantage of this technique is that it is very time consuming as it is a manual approach and it needs to be repeated each time the lighting of the room changes slightly. Advantage of a manual approach to the color calibration is that it is accurate since the decision on what color should be classified is made by a human.

An accurate and robust automatic color calibration technique is a very useful component of a RoboCup soccer software for NAO robots. In this thesis the development process of automation color calibration technique that aims to maintain the accuracy of a manual approach is proposed.

1.2 Related research in the field of automatic color calibration

As mentioned previously it is of a high importance for a RoboCup team to have an accurate, robust to luminance variation and reliable color calibration system within their robot soccer software. In order to save human resources and time spent on the manual color calibration, many teams have tried to develop an automatic color calibration algorithms. In this section an overview of several methods for automatic color calibration is presented.

1.2.1 HSI colour space based color calibration

The HSI color space based automatic color calibration technique has been proposed by the members of RoboCup team NUBots from the University of Newcastle, Australia[HKC08]. In their research frames acquired by NAO camera have been converted from YUV (default NAO camera image format) to the HSI color space in order to obtain the hue component which represents the angular color value of the pixels in the frame. Graphical representation of the HSI color space can be seen in Figure 1.1

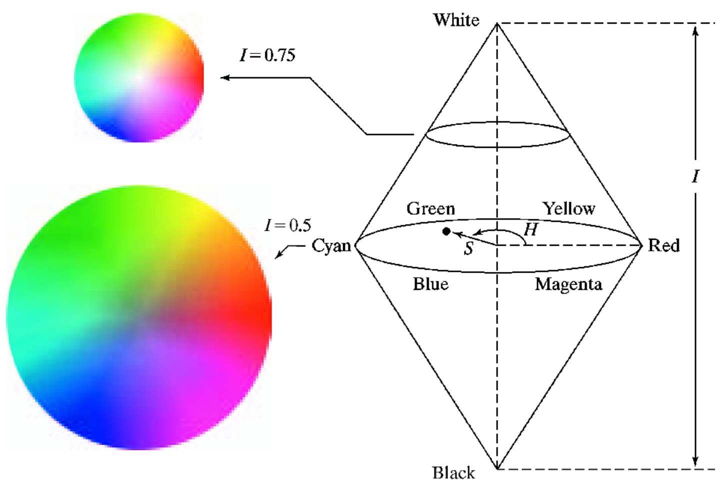


Figure 1.1: HSI color space graphical representation [Gon09].

HSI color channels can be described as follows:

- **Hue** - angular value in range from 0°to 360°, describes to which degree pixel color value is similar or different to the main color angular values defined as red (0°), yellow(60°), green(120°), cyan(180°), blue(240°), magenta(300°);
- **Saturation** - value from 0 to 1 that describes colorfulness of a pixel, i.e. the amount of color of a corresponding hue value;
- **Intensity** - value from 0 to 1 that describes the amount of light in the pixel.

After the conversion between color spaces, when the Hue value of the pixels is obtained, it can be used in the process of decision making when labelling the pixel values, e.g all the pixel values, whose Hue component falls in range 100-140 degrees, will be classified as green. This approach reduces the dimension used for classification to just one channel of a HSI color space and provides user with a readable histogram data that can be easily analysed by a human.

In the research of NUBots team expectation maximisation algorithm was proposed to be applied to the histogram data in order to compute the parameters of multivariate Gaussian distributions of each color class. Then the HSI color table is generated based on the standard deviation values obtained from the distributions of each color class. Proposed method also suggests to defined so called "soft color" classes, which will represent the pixel values that according to the Gaussian distribution's standard deviation are equally probable to be classified as 2 different color classes. Blobs of pixels classified as some "soft color" class will only be processed when the blob of sufficiently big size is formed.

The last step of the proposed technique is the computation of a YUV color table according to the data obtained in the HSI color table. The whole process of described algorithm takes approximately 10 minutes to obtain the automatically generated YUV color table. Experimental results presented in this paper state that proposed technique provides around 83-94% object recognition rate depending on the type of object.

Object recognition rates, obtained from experimental results of aforementioned technique, are still lower than those obtained when calibrating colors manually. It also needs to be noted that one of disadvantages of the proposed technique is the conversion from HSI to YUV color spaces. The aim of converting between two formats is to reduce the number of dimensions used in the process of color classification from two to one which might not provide sufficiently significant changes for the performance of classification algorithms while causing the necessity of having additional computation steps of conversion between color spaces in the beginning and at end of the automatic color classification procedure. It is believed that all necessary color information can be obtained by analysing combination of U and V color channels while preserving the luxury of not having additional computation steps in the automatic color calibration technique.

1.2.2 YUV/RGB color space based colour calibration

Apart from expectation maximisation algorithm mentioned in previously described work of NUBots RoboCup team, David M. Budden in his thesis "Applications of Machine Learning Techniques to Humanoid Robot Platforms"[Bud12] writes about several other techniques that can be applied to NAO robots in order to perform clustering of colors in the frame obtained from NAO camera and conduct unsupervised color recognition by automatically generating the color table. Budden's thesis focuses on the mean shift and mode finding based segmentation methods. Detailed implementation description of the algorithms such as K-means clustering, expectation maximisation and mean shift are presented along with the performance results for each method.

Apart from describing several clustering techniques, color table generalisation method based on Support Vector Machines (SVM) classification model was suggested to be implemented in order to make the clusters well organised by removing extreme value, filling the "holes" within the clusters and making the shape and boundaries of a cluster to be smoother [CCS03, TC01, RMSK13]. As a result of research conducted in Budden's thesis the conclusion is made that optimal method for automatic pixel values clustering is application of a K-means algorithm without application of SVM based color table generalisation [Bud12]. K-means clustering method proved to produce color tables that provide best object recognition rates, satisfying both classification sensitivity, degree to which object detected on the segmented frame was actually present on the frame, and specificity, degree to which object that is not present in the frame is not being detected. Cluster classification in Budden's work was performed by finding shortest distance from corner values of RGB cube to the cluster values. That approach of cluster classification suffers from the fact that unrealistic toxic color values, that correspond to the RBG color space cube corners or regions close to the corners, will be classified in favor of more realistic color shades.

Algorithm proposed in the given thesis tries to benefit from the advantage of using YUV color coding and K-means algorithm for color clustering. When using K-means algorithm, number of means (k) needs to be defined, along with the initial mean values. In the classical version of the K-means algorithm initial mean values are assigned randomly. In order to ensure that for each desired color class there will be at least one cluster present in the output of the K-means algorithm , certain amount of initial meas will be assigned systematically. Other initial mean values will be calculated from the input data. Initial means assignment is described in detail in Section 1.4.2.

Cluster classification will be performed in YUV color coding by analyzing the luminance information of the input data. The points that will be used to find shortest distances to the cluster values will be assigned correspondingly to the results of luminance information analysis. This approach of cluster classification will allow to accurately find most suitable clusters from the range of similar clusters. As a result, realistic color shades will be classified as desired color classes in favor of classifying too bright or too dark clusters of similar type. Details of cluster classification part of a automatic color calibration algorithm are presented in Section 1.4.6.

1.3 The proposed automatic color calibration technique

1.3.1 Color space selection

From the machine-learning point of view, automatic color calibration is a process of classifying sample data into desirable color classes based on the features extracted from given data sets. Features extracted from the data will represent the geometrical position of a sample data point in the color space. There are many different color models available for purpose of representing color information. The most well-known is RGB, in which all colors are represented by the combination of red, green and blue channel values. Each channel in the color space can represent a certain dimension in a 3D space. The RGB color space geometrical representation is shown in the Figure 1.2.

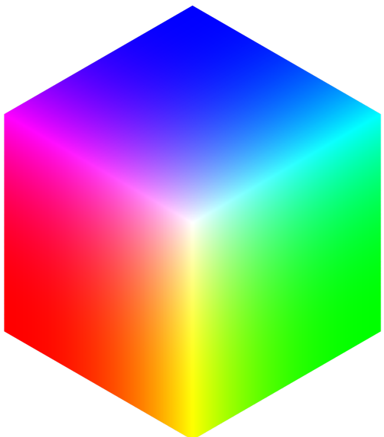


Figure 1.2: RGB color space cube [Gon09].

In the process of developing an automatic color calibration algorithm, first, the color space needs to be selected as it will determine the nature of the data that will be provided to the algorithm as an input. Native color space of the NAO camera is YUV[ald]. The highest performance in terms of time is achieved by working with color spaces of YUV types as it does not require additional computational steps of conversion between default camera color space format and such color spaces as RGB or HSI.

RGB colour space represent luminance and chrominance components together hence slight change of the light cause significant changes in definition of colours. However in YUV colour space similar to HSI, luminance (Y and I) is separated from chrominance (UV and HS). Thus some changes in the lightening will not affect on the colour information.

In the YUV color space Y channel, also referred to as luminance or luma, represents the brightness of the color, while the combination of U and V channels represent the color information, also referred to as chrominance. Geometrical representation of the YUV color space in 2D and 3D is shown in the Figure 1.3. This structure of the color space allows to analyse and manipulate luminance and color information separately as they are not correlated with each other. The values of the Y channel range from 0 to 1, which is usually represented as a range from 0 to 255. U and V values range from -0.5 to 0.5, which corresponds to the range from -128 to 127 in signed digital form, and from 0 to 255 in unsigned form. Further in this thesis the digital unsigned form of channel values range will be used.

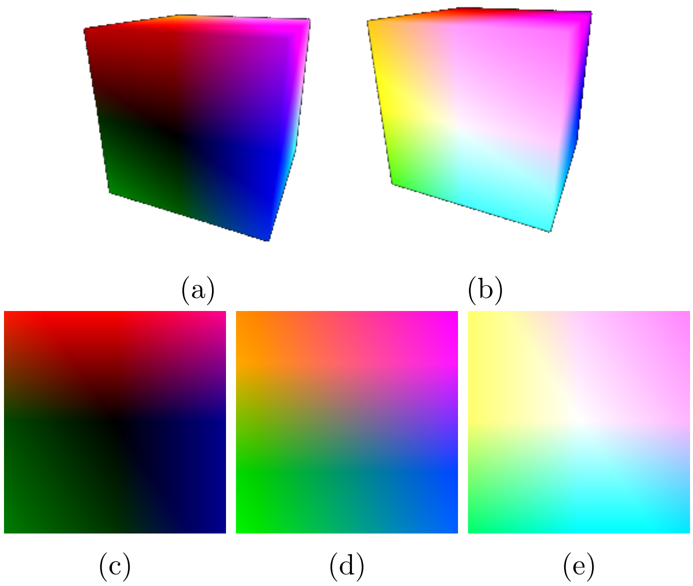


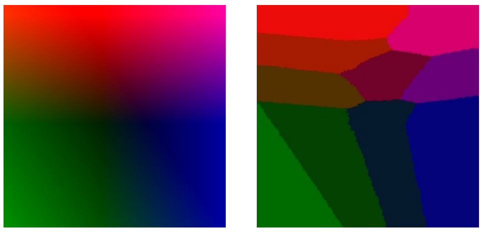
Figure 1.3: Geometrical representation of colors in YUV color coding system. (a) Color space cube from dark side; (b) Color space cube from bright side; (c) UV plane at Y=0; (d) UV plane at Y=128; (e) UV plane at Y=255 [Gon09].

In conclusion, the YUV color space has been selected to be used in this thesis for data representation in the automatic color calibration algorithm. In case of programming for NAO robots the utilisation of the YUV color space will allow to maintain low computational complexity of the algorithm while preserving the advantages of separate analysis of chrominance and luminance information acquired from the camera frame.

1.3.2 Color Clustering

In order to classify the input sequence quickly and accurately a two stage process is suggested to be used for the purpose of automatic color calibration. The first stage of the algorithm is the input pre-processing, which in case of the proposed method is color clustering [Cel90, BASK08]. Second step of the algorithm is the cluster classification, described in details in the Section 1.4.6.

Color clustering is the process of combining colors of the similar type into one cluster and assigning one single label to all the members of the cluster. Value being assigned will be a centroid value of a cluster. As a result of this process a simplified version of the input with less details and similar color combined together is obtained. Further in the text the pre-processed image is referred to as segmented image. Example of the input and the output of the color clustering by K-means method can be found in the Figure 1.4, where 65536 different colors, each represented by 3 8-bit integers, are combined into 10 clusters (i.e. k=10).



(a) (b)

Figure 1.4: (a) UV plane (Y=20); (b) Segmented version of (a) after applying K-means algorithm (k=10). Colors of the same type are combined into clusters.

The aim of color clustering procedure is to reduce the number of different pixel values that can be present in the input image. That would allow for a faster color classification, as in case of pre-processed image, there is no longer need to deal with a wide range of possible pixel values, as all the similar pixel values have already been combined into clusters[DH14].

There are many different techniques that can be applied in order to combine the pixel values into clusters. As stated in the Section 1.2.2, K-means method has been experimentally proven to provide best performance results among all the other clustering methods of the mean shift and mode finding family of clustering techniques[Bud12, GWR06, LLX$^+$13]. For that reason, this clustering technique has been chosen to be applied for the purpose of color clustering in given thesis.

K-means algorithm will also allow for a more accurate cluster classification, as in the process of color clustering the K-means algorithm will attempt to find cluster centers so that within-cluster distances will be minimized and between-cluster distances will be as big as possible.

K-means is a clustering method that aims to segment the data into given number of clusters (k-number of clusters) by performing following steps:

1. Initialization step: Select initial means randomly;
2. Assignment step: Assign data point to the cluster with the nearest mean to form initial clusters;
3. Update step: Calculate new means for each cluster;
4. Reassign data points to clusters according to new means;
5. Repeat steps 3 and 4 until convergence.

Mathematically K-means for color clustering can be formulated as a problem of minimizing the squared error function J:

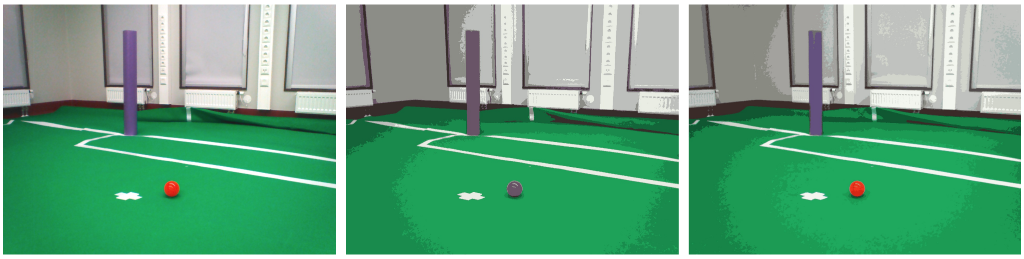
$$J = \sum_{i=1}^{k} \sum_{p \in C_i} \parallel p - \mu_i \parallel^2 \tag{1.1}$$

where given a set of pixels $(p_1, p_2, ..., p_n)$ presented in the frame, they are assigned to a cluster C_i so that within-cluster sum of squares of euclidean distances is minimized[KMN$^+$02, AMFM11].

Apart from original K-means algorithm, there exist many variations of K-means clustering. Depending on the application, it is possible to define differently such properties, as how to assign the initial means, when to stop the algorithm (what is the maximum number of algorithm's iterations), which metric to use in order to calculate distance of sample points to the mean value of a cluster etc. The original version of the K-means algorithm is described above and it uses randomly assigned initial mean values. In this thesis the utilization of the original version of a K-means algorithm is not suitable due to the randomness in initial means assignment. When assigning the initial means randomly, the probability of not having a separate cluster for each of desired color classes in very high, especially for clusters of a small size(e.g. soccer ball). Figure 1.5 illustrates examples of the output of K-means algorithm when selecting means randomly.

As can be observer from sample frames in the Figure 1.5, when selecting means randomly, some colors (orange and purple) that are supposed to represent two different desired color classes are combined into one cluster along with some noise data from the walls.

In order to make sure that pre-processed segmented image definitely contains at least one cluster for each of color classes that are meant to be classified as a result of automatic color calibration algorithm, the proposed method uses systematical approach to the selection of initial means for the K-means algorithm. Following logic is applied in order to define initial means: 4 means are selected based on the



(a) (b) (c)

Figure 1.5: (a) Input frame; (b) Output of K-means with randomly selected means; (c) Desired output of K-means color clustering for current application.

average color value of the field objects calculated from the testing set, remaining initial means depend on the input frame and are calculated in a following manner:

1. Input frame is divided into 4 by 4 grid;
2. For each part of the grid the average value of the grid is calculated;
3. Average value of a grid is added to the array of initial means;

As a result, previously described initial means set will allow to cluster the image into clusters that will represent the desired color classes as well as the clusters that will combine the noise into bigger areas in the frame.

1.3.3 Cluster classification technique

In order to classify the clusters into correct classes, currently being Green (soccer game field), Orange (RoboCup soccer ball), Purple (Philosopher team soccer goals), White (soccer field lines) and Undefined (noise), the center points need to be defined, which will then be used in order to find shortest distance from the available clusters to the defined centers. Based on the shortest distance cluster will be assigned to the center point that is the closest to the value of a cluster.

As stated in the Section 1.2.2, in the previous research [Bud12], such method as classifying the clusters by using the minimum distance from the corners of RGB color space cube that correspond to the closest suitable color (e.g. red corner (255, 0, 0) for classification of the orange) was used. This method is relatively efficient, but has a problem of classifying clusters of unrealistically bright colors in favor of color shades, which in case of RGB color space are located more far away from the corners of the color space cube than the unrealistically bright colors (e.g. toxic red (255, 0, 0)).

In this thesis a new approach to the cluster classification problem is proposed. New method aims to find most suitable centers in the YUV color space cube by analyzing the luminance information of the input data. Result of luminance analysis, which is described in detail in Section 1.4.2, is applied to the average values of desired color classes obtained from a set of sample frames. As a result 4 values that represent specific point in the YUV color cube, further in the text those points are referred to as points for classification.

Apart from those points, 4 threshold values will be defined for each color class. Those thresholds values along with the points for classification will form a spherical or semi-spherical (in case when center is located near the edge of color space cube of on the edge) areas in the YUV color space cube that would correspond to the areas that represent the pixel values which are valid to be classified as one of the desired color classes. If the value of a cluster does not fall into any of defined spherical areas, this cluster will be classified as Undefined color, which means that this cluster does not belong to any of the object of interest (ball, field, line or goal) in the segmented image.

1.4 Implementation

1.4.1 Sample frames

First important aspect that needs to be considered in an automatic color calibration algorithm development is the testing set camera frames. Hence one should answer this question: what images of the soccer field and surrounding environment need to be selected in order to make sure all the classical cases will be covered, as well as special ones?

Main requirement to the testing images is that they would contain the objects of interest such as soccer field, goals, RoboCup ball and some amount of noise that will serve as an example of data that does not need to be classified at all. In the resulting segmented image, noise (e.g. falls, ceiling) will be classified as undefined color class. Figure 1.6 shows some examples of frames taken from NAO robot camera that will serve as a testing samples in algorithm development.

(a) (b) (c)

Figure 1.6: (a) Frame that includes all objects of interest (ball, goal, lines, field); (b) Frame that includes some objects of interest along with specific noise; (c) Frame from a different angle, objects of interest are far away.

Another important requirement to the testing set is the luminance value of separate frames and overall luminance average value of a testing set. As the luminance average of the frame currently being processed will be used to calculate position of initial means for K-means algorithm and points that will be used for cluster classification, the average of testing set should be near the average value of Y component in YUV color model. This way, following generalization can be made for brighter or darker images: position of the initial means and points for cluster classification will change form initially defined values proportionally to the difference between average value of Y component (i.e. 128) and the average luminance value of the frame currently being processed.

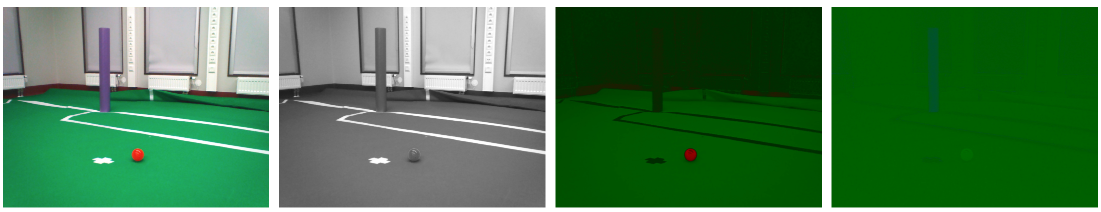
To satisfy aforementioned requirements, 33 images of a soccer field under different angles and with Y component average varying from 119 to 139 have been chosen to be in the testing set. Total luminance value average of a testing set is equal to 130, which is sufficiently close to the Y average in YUV color coding system.

1.4.2 Luminance analysis

As mentioned previously, when working with pictures that are represented in YUV color coding system, luminance information can be processed separately from the color information. Image in YUV color coding consists of 3 matrices:

- First matrix is the **Y** component representing luminance of the pixels;
- Second matrix is the **U** component representing blue-light contribution to the pixel value;
- Third matrix is the **V** component representing red-light contribution to the pixel value.

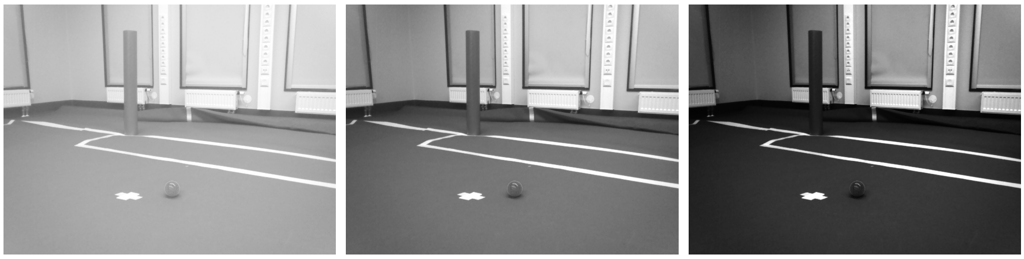
In case of luminance analysis, only first matrix (Y component) needs to be considered. In this matrix 0 value would indicate that given pixel is very dark, and 255 would indicate that pixel is very bright. Example of separation of an image into 3 components of YUV color model is shown in Figure 1.7



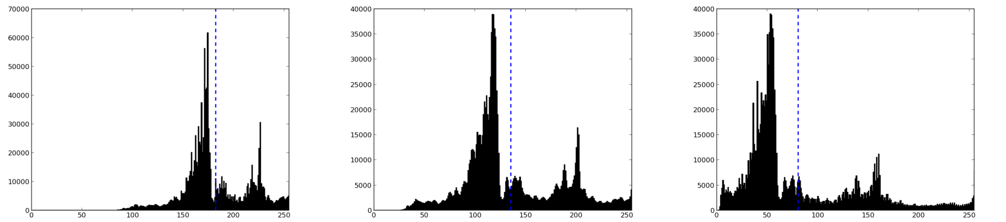
(a) Original camera frame (b) Luminance intensities (c) Reg light intensities (d) Blue light intensities

Figure 1.7: Example of separation of a frame into different YUV components

When the matrix with luminance information is obtained, the histogram of luminance intensity can be calculated. This histogram can be defined as a table, where for each Y component value the frequency of its value presence in the frame is calculated (Figure 1.8).



(a) Bright frame (b) Normal frame (c) Dark frame



(d) Histogram of frame a, average 183 (e) Histogram of frame b, average 125 (f) Histogram of frame c, average 80

Figure 1.8: Frames, histograms and average values of Y component.

The histogram of luminance intensity is used for the further luminance analysis. The average value of luminance intensity is calculated from a histogram of an image currently being processed. The average value of luminance intensity is then compared to the average value of Y component in the YUV color model, which is equal to 128. Difference between two values is used in order to compute the Y component value of 4 initially assigned means for the K-means clustering. Those means will also be used in the further cluster classification process.

Initial values for 4 initially assigned means have been calculated based on the average of testing set. The initial values of the means are following:

(a) Orange mean (100, 90, 220) (b) White mean (230, 128, 128) (c) Purple mean (80, 160, 180) (d) Green mean (100, 70, 115)

Figure 1.9: Initial mean values in YUV color coding.

When the luminance analysis of currently processed frame have been made, the difference between average value the luminance intensity of a currently processed frame and the average of Y component in YUV color model is added to the Y component values of initially assigned means. That way values of initial means and points for classification will be corrected in terms of luminance. For the brighter images the values of initial means will shift to the brighter side of YUV color cube and for darker images they will shift to the darker side. Example of the luminance analysis and its influence on the initial means values is illustrated in the Figure 1.10

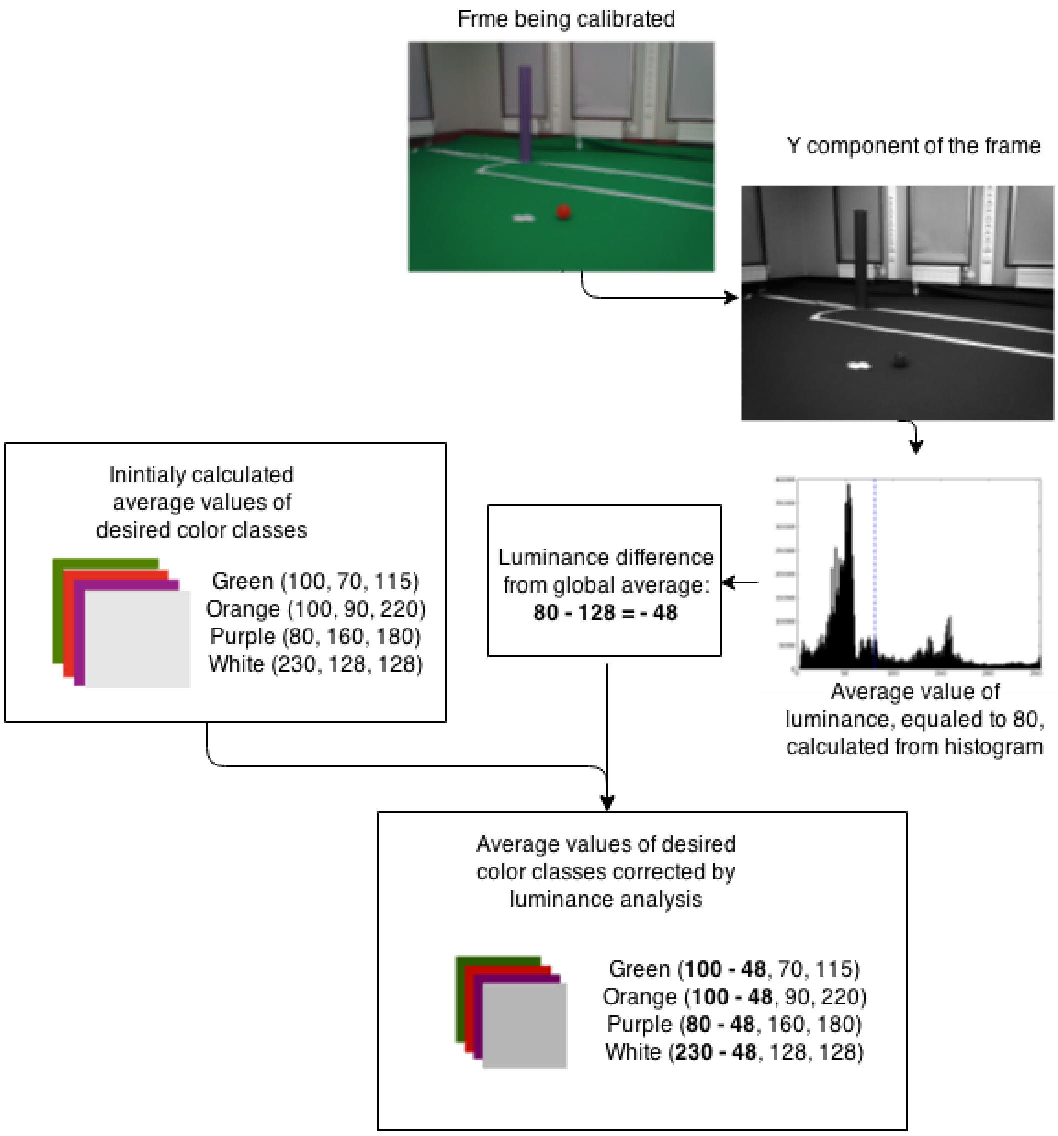


Figure 1.10: Example of luminance analysis result.

1.4.3 Number of means

When using K-means algorithm the number of means, variable k, needs to be defined according to the output expectations. If too few means are selected, not all desired clusters will be present in the output, as noise might be combined into one cluster along with the data that needs to be classified. And too large value of k is unnecessary as then many similar colors will be clustered into different clusters. Figure 1.11 illustrates some examples of fortunate and unfortunate number of clusters.

(a) Clustered image, k=10 (b) Clustered image, k=20 (c) Clustered image, k=10 (d) Clustered image, k=20

Figure 1.11: Initial mean values in YUV color coding.

There are many different approaches to calculate the optimal value for k[WCR+01, RT99, CKV13]. For example, most simple rule of thumb states that the number of clusters should be approximately equal to the square root of number of data points divided by 2. However, in case of NAO robot camera, number of data points if each frame is equal to 1280*960, and according to the rule of thumb number of clusters should be equal to 784, which is an unnecessary large number of clusters for current application.

Another popular approach to determining number of clusters in data set is so called Elbow method. According to the Elbow method, when increasing number of clusters, the within clusters sum of squares for all the clusters is decreasing. In beginning, the drop in value of within clusters sum of squares will be more significant than for larger numbers of k. So the value of k, after which the decrease in sum of squares becomes less significant, is defined as an "Elbow" and is considered to be the optimal number for variable k. However, in particular case of automatic color calibration, Elbow method can not be applied due to the wide variability of the input data. In reality, the optimal number of means will differ significantly for different input frames. For some cases, when there is only green field with white lines present in the frame, even the k equaled to 2 will be enough to accurately cluster and classify the input data. But in cases, where large amount of noise is also present in the frame, number of clusters needs to be increased significantly.

Experimental results have shown that in case of NAO camera and RoboCup field the optimum number of clusters is 20.

1.4.4 Initial means for clustering

As stated in Section 1.3.2, the proposed method implements following logic to select initial means for color clustering: 4 means will correspond to the desired color classes average values in the YUV cube obtained from sample images of soccer field objects and corrected by the result of luminance analysis. Remaining 16 means are selected based on the input data. Acquired frame is divided into 4x4 grid and mean value of every grid is selected as one of the initial mean value for the color clustering.

In case where there is a lot of noise in the image, input data based initial mean values selection will serve mainly as a “noise catcher” and will help to select the initial mean values that will combine the noise data of the similar type into clusters.

Example of initial mean assignment is shown in Figure 1.12

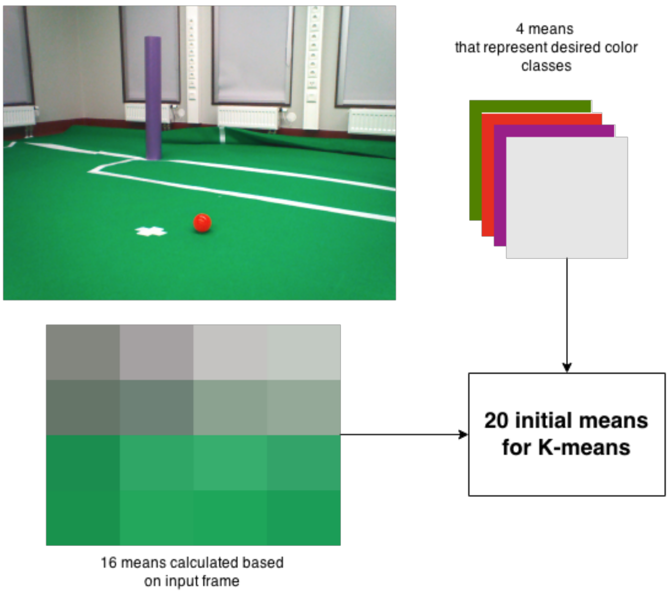


Figure 1.12: Initial means used for K-means clustering.

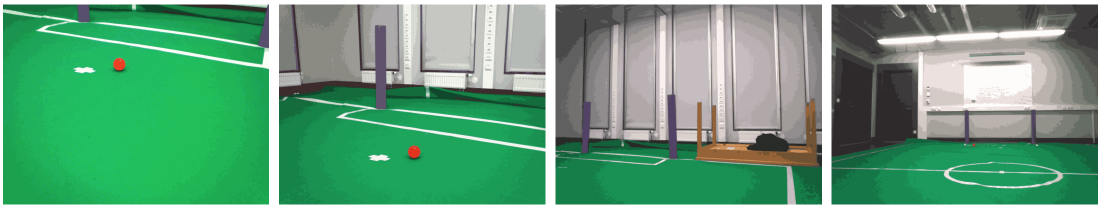
1.4.5 Clustered images

The algorithm for clustering was implemented in Python 2.7 using Scikit-learn library built-in function K-means[PVG+11]. Initial means and number of iterations to run algorithm from the start till convergence is provided as arguments to the build in function. The logic of calculating the values for initial means is described in section 1.4.3.

When implementing the original version of the K-means algorithm (i.e. when initial means are selected randomly), the algorithm might be run several times from start till convergence in order to produce several outputs which can be analysed for the purpose of selecting the output that will have the minimal value of sum of squared distances inside clusters.

However, in particular case, as the values for initial means are fixed for every particular frame and there is no randomness involved in calculating the values for initial means, the algorithm will be deterministic and one iteration of running algorithm from the start till convergence is enough.

Figure 1.13 illustrates the output of the clustering that will serve as an input to the classification part of automatic color calibration algorithm.



(a) (b) (c) (d)

Figure 1.13: Output of K-means clustering (k=20). (a), (b), (c), (d) Clustered versions of different sample frames.

1.4.6 Cluster classification

After K-means clustering is performed, clusters need to be classified into 5 classes: Orange, White, Purple, Green and Undefined. For that purpose, previously defined and corrected by luminance analysis average values of desired color classes are used, here those points are referred to as points for classification.

Euclidean distance from every cluster centroid to every point for classification is calculated. The shortest distance is then compared to the corresponding threshold value, which varies depending on the color class. If value of the shortest distance is less than the value of defined threshold, the cluster will be assigned a label of a corresponding color class. Otherwise cluster will be assigned the Undefined label.

Aforementioned threshold values have been determined by conducting several cluster classification tests on a set of sample frames, which was describe in the Section 1.4.1. Best results were achieved, when values of the thresholds were set to 40, 60, 35, 50 for the Orange, White, Purple and Green color classes respectively.

In the future, values of the thresholds can be manipulated if new conditions do not allow to maintain satisfactory performance of the color calibration algorithm. After several different competitions and additional testing of the algorithm, threshold values for the cluster classification will be assigned to some final static value.

1.5 Complete automatic color calibration algorithm scheme

Figure 1.14 illustrates schematically the complete automatic color calibration algorithm.

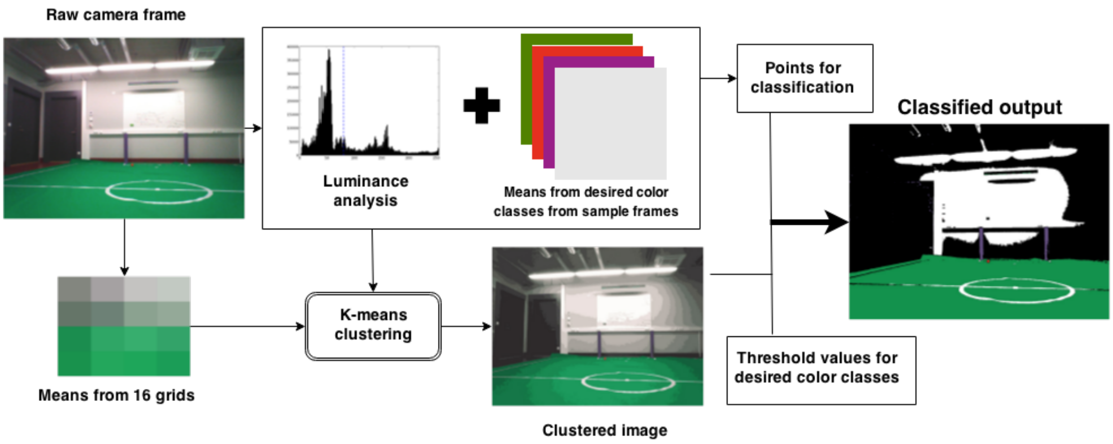


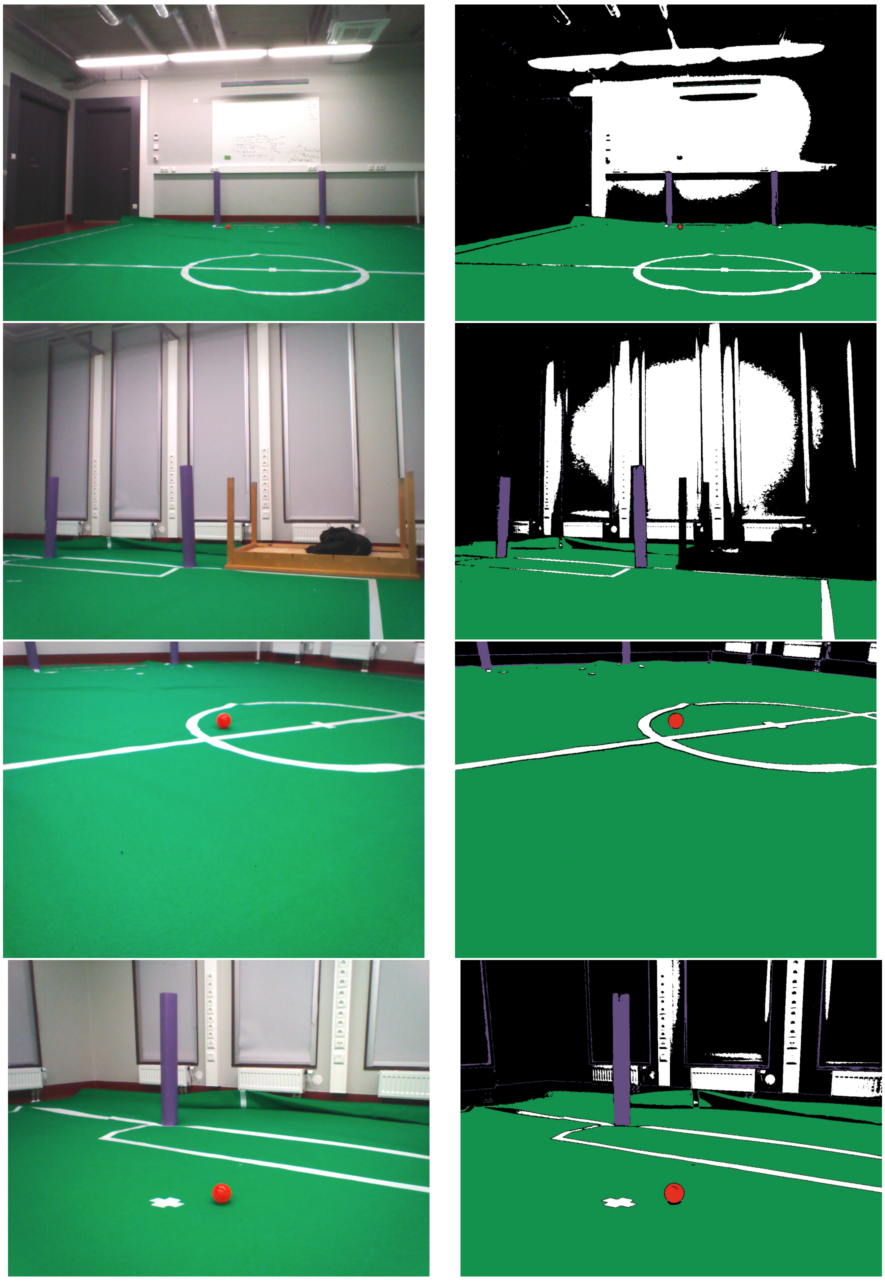
Figure 1.14: Automatic color calibration algorithm scheme.

1.6 Results and future work

As a final result of the proposed color calibration method, all the pixel values in the frame will be represented by one of the 5 main color classes: 4 desired colors for field objects and Undefined label, which is represented by black color in visual examples of the output of the algorithm.

Figure 1.15 illustrates examples of input and corresponding output of the automatic color calibration algorithm. As can be seen from examples, all objects of interest (ball, field, lines and goals) are assigned correct color class and noise (table, ceiling etc.) are mainly assigned to Undefined color class.

Before the soccer match, several images of the field objects will be taken and automatic color calibration algorithm, described in this thesis, will be applied to them. Techniques utilised in the algorithm aim to minimize the probability of having incorrect classification result, but still the probability is present. Hence, after the results of automatic color calibration are obtained, quick human confirmation is required to eliminate the unfortunate classification results from the set of the classified frames that will be used to form a color look-up table.



(a) (b)

Figure 1.15: (a) Input raw camera frames; (b) Output of automatic color classification algorithm.

In the further stages of proposed algorithm development, the results of a current system will be analysed and new additional techniques might be added to the algorithm for the purpose of minimizing the probability of incorrect classification to a sufficiently low value (<10%), so that human confirmation will no longer be needed.

Chapter 2

White goal detection

2.1 Motivation

Now, that the color calibration question is solved, it is possible to proceed with the next step of visual data processing, which is object detection. Several new challenges have been introduced in the year 2015 for RoboCup participants. One of them is the challenge of detecting white goals. In the previous years the color of the goals used in RoboCup competition was yellow, which is easier to distinguish from white field lines. Because of new challenge current modules, such as object detection, in particular goal detection and classification, need to be updated before they can correspond to the new RoboCup requirements. In this thesis, overview of the old algorithm and the changes introduced to the current goal detection system are described in detail.

2.2 Current system description

In order to describe current goal detection system following terms need to be explained.

- **Frame** - raw image acquired from NAO robot cameras (top or bottom);
- **Segmented frame** - frame from NAO camera after being filtered using LUT, here only defined color classes are present;
- **Horizontal histograms** - table where for each row of segmented frame there is a value of how many yellow pixels (goal color in previous years) is present in this row;

- **Vertical histograms** - table where for each column of segmented frame there is a value of how many yellow pixels (goal color in previous years) is present in this column.

In the current system in order to detect goals on the soccer field following steps of frame processing need to be conducted:

1. Retrieve frames;
   (a) Retrieve top camera image;
   (b) Retrieve bottom camera image.
2. Scale frames;
   (a) Scale top camera image from 1280x960 to 160x120;
   (b) Scale top camera image from 1280x960 to 80x60.
3. Compute vertical histogram;
   (a) Get color information from segmented frame;
   (b) Construct color histogram for every column of the frame.
4. Analyse vertical histogram;
   (a) Search for maximum values of yellow color in vertical histogram;
   (b) Form rough vertical bounds for goal posts.
5. Compute horizontal histogram;
   (a) Get color information from segmented frame;
   (b) Construct color histogram for every column of the frame.
6. Analyse horizontal histogram;
   (a) Search for maximum values of yellow color in horizontal histogram;
   (b) Estimate horizontal bounds of possible goal posts.
7. Evaluate the quality of possible goal posts.
   (a) Look for a strong edge in the centre and base of possible goal posts;
   (b) Calculate the percentage of yellow color in the segmented area of possible goal post;
   (c) If percentage of yellow color in the segmented area of possible goal post is less than 50 % eliminate the goal candidate from the list of possible goals;
   (d) Check if base of the possible goal post is below the estimated field edge.

2.3 Conducted work

A system for goal detection has to be designed so that the algorithm would depend on color information as little as possible. Apart from that, a new sanity check need to be introduced in order to distinguish white goals from white lines on the soccer field. In this section, new or updated parts of the algorithm of goal detection system are described in detail with some visual examples.

First modification that needs to be introduced into the system is that instead of yellow color histograms white color vertical and horizontal histograms will be calculated and used in order to find vertical and horizontal boundaries of a post candidate in the frame by triggering the local maxima values in the histograms that exceed the values of defined threshold (Figure 2.1).

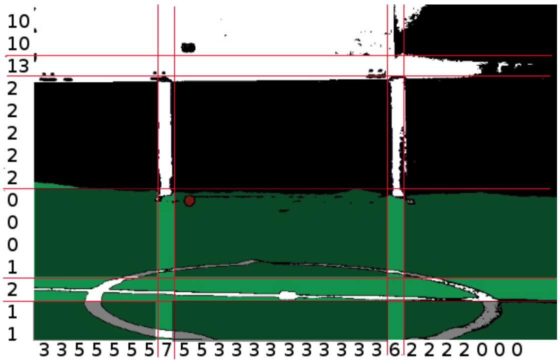


Figure 2.1: Goal post candidates areas after the analysis of the white color histogram. Numbers on the left and at the bottom of the image represent pseudo-values for the vertical and horizontal histograms respectively. In provided example, threshold value for minimum value of a local maxima is set up to be 5 and 2 for horizontal and vertical histograms respectively.

The main complication at this point comes from the fact that many field lines will also be assigned a bounding box. Noise caused by the white field lines is be eliminated in the further processing steps, so-called sanity checks.

Apart from sanity checks that are already present in the previously utilised goal detection system (Section 2.2), new system will implement the check for whether the top horizontal bound is above estimated field edge or not. In case of field lines, which represent the main noise factor in this particular object detection algorithm, top horizontal bound should always be below the field edge value.

However, since both field edge line and a top horizontal bounds are estimated, most likely, there is some amount of noise in those estimations, which may cause a situation, where top horizontal bound of the bounding box around the field lines, which are located at the edge of a soccer field, are estimated to be above the field edge. This situation is most probable, when the robot tries to predict values of the objects far away from robot's current location on the field.

In order to solve this issue, a threshold value for a minimum possible difference between top horizontal bound and field edge is defined. Considering the threshold, if the situation occurs, that the top horizontal bound of a field line bounding box is predicted to be above the field edge, difference between Y coordinates of the bound and a field edge with high probability will be less than the value of a threshold, which will cause the bounding box to be eliminated from the set of possible goal post candidates. As a result of new sanity check, the noise caused by the white soccer field lines will not be classified to represent a possible location of a goal post.

2.4 Results and future work

After the algorithm was developed, the test have been conducted to evaluate the reliability of new goal detection system. Robot have been placed on several points in the soccer field facing the goal. Figure 2.2 illustrates the position of the robot during the test.

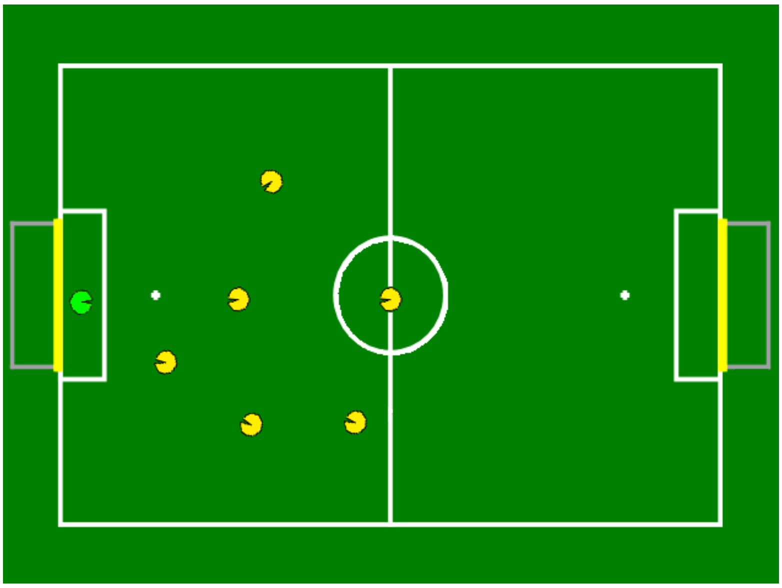


Figure 2.2: Robot positions during the white goal detection test. Yellow circles are positions of the robot that is detecting the field objects. Green circle is a robot used to represent a goalie in the frame.

Testing revealed that the detection of other object of a similar type is interfering with the goal detection algorithm on a system architecture level. The priority of a robot detection system is given a higher priority than the goal detecting system. Due to that fact, when an area in the frame satisfies the requirements of both robot and a goal detection algorithms, it will be assigned the label of a robot rather than the label of a goal post.

Figure 2.3 illustrates example of the frame where the white goal posts have been detected as a robot.

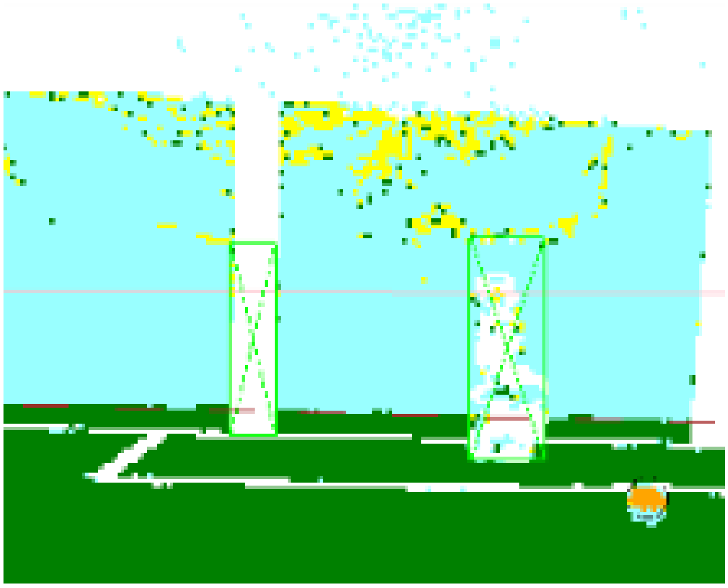


Figure 2.3: Detection of the robot instead of a goal post. Green bounding box with the cross indicates the area of a frame, where the robot is detected.

In conclusion, it has been shown, that more general and global changes need to be introduced to the system in order to be able to detect white goals and distinguish them from the noise of similar objects on the field. For the future research, one solution to the revealed problem might be an introduction of the probabilistic model that will evaluate white blob areas in the frame and based on the features of the blob it will assign a probability value for a white blob to be a robot, line or a goal. Those probability values will then be compared and most probable model will be selected.

Chapter 3

Localization system enhancement

Localization system heavily depends on visual information acquired by robot's camera. One of the most popular approaches for the localization of the robot on the field is to use the visual information in the probabilistic model such as Kalman or Particle Filter to generate several models of robot's possible location and select the most probable model of all to represent the robot's actual position on the field. In this thesis, the approach of introducing certain features into the game strategy in order to improve localization performance is described.

3.1 Current system description

Currently utilized soccer software uses implementation of multi-modal Kalman Filter to keep track of positions of the robot, ball and 4 teammates on the soccer field. In the Kalman Filter, both noise and signal value are assumed to have a Gaussian distribution. In case of a Gaussian distribution, mean value will represent the most probable value of a random variable (Figure 3.1).

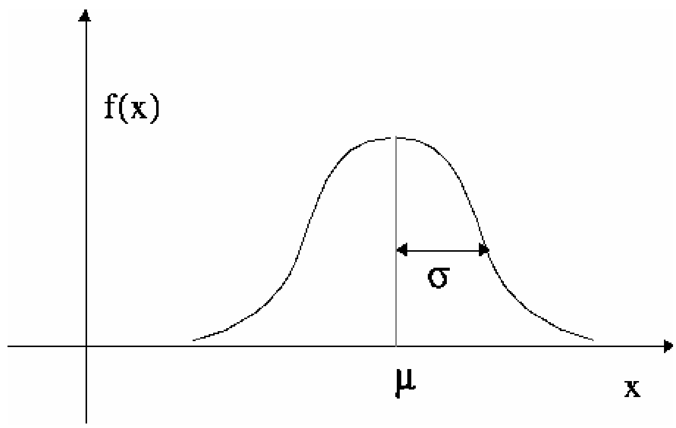


Figure 3.1: Probability density function of a Guassian distribution[RCA93].

In particular case of robot soccer, the state, or a state space, is a 19 dimensional vector that contains mean values of a Gaussian distributions that model position of robots (3 values for every robot: X and Y coordinate and the heading) and the ball (4 values: X and Y coordinate and 2 values to represent ball velocity in a 2D environment). Taking into account weighted values of previous state space estimate and new observations from robot camera, such as ball, goals or field lines, current state estimation is being made. Detailed description of the multi-modal Kalman Filter algorithm is outside the scope of this thesis.

3.2 Localization enhancement approach

The idea for a localization enhancement technique is that, when the number of observations increases, it has a positive influence on the accuracy of predicted state estimate[SU07, JU97, Dav03]. By using the values of standard deviations calculated from the covariance matrix of a Kalman Filter, it is possible to trigger the moment when the accuracy of prediction of state space elements is no longer sufficient for a robot to perform well in a soccer game. When this situation occurs the behaviour of the robot will change so that larger number of field landmarks is observed.

As the Kalman Filter is a probabilistic approach of estimating the robot's location, such statistically important variables as variance and correlations of random variables of a Gaussian distributions are also computed and updated each time during the process of pose estimation. Those variables are represented by a covariance matrix and can be utilised in order to evaluate the accuracy of the output of probabilistic model, by calculating standard deviation values.

In given case, covariance matrix is a symmetric 19 by 19 matrix. Diagonal elements of covariance matrix represent the variance values of Gaussian models of each variable in a state space. Off-diagonal elements represent the covariance between elements of a state space. The elements of interest, in given case, are located in the 2 by 2 upper left part of a covariance matrix, which represent the variances and covariances of Gaussian distributions of X and Y coordinates of a robot's position on the field. In case of robot's position, the error of coordinate prediction can be represented by an area in the field around the means of predicted values for X and Y coordinates. This area is computed in the following manner:

1. The elements of interest are extracted from the covariance matrix in order to form a covariance sub-matrix;

2. Eigen decomposition is performed on a matrix obtained after step 1 in order to obtain magnitude (eigenvalues) and direction (eigenvectors) of the variances of X and Y coordinate estimates;

3. In order to make sure that the eigenvalues calculated in the previous step are real numbers, the complex magnitude of the obtained eigenvalues is calculated and assign to represent the eigenvalues in the further computational steps;

4. Standard deviation is computed by taking a square root of eigenvalues obtained in step 3;

5. As shown in the Figure 3.2, the two values of eigenvectors form the direction of variances and standard deviations show the magnitude of the error in the current estimate.

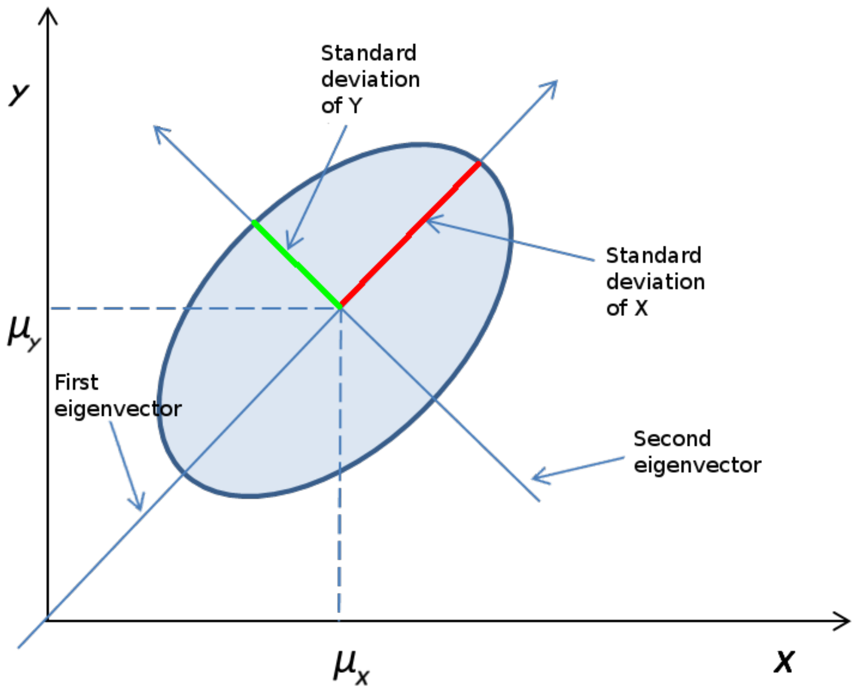


Figure 3.2: Area of robot's possible positions represented by standard deviations of coordinate estimates.

This way a standard deviation is used in order to estimate the possible error of an output of Kalman Filter for estimation position of the robot on the field. The error can be interpreted as a set of robot possible positions. Elements of this set will cover some particular area in the soccer field. This area represents 68.2% of most probable positions of the robot at the current moment[RCA93]. The size of the area can now be easily analysed with the purpose of defining the threshold value, which will indicate that the set of possible positions, which encodes the standard deviation of prediction, is large enough to consider that the robot estimation of it's own position on the field is not reliable enough.

In the currently utilized soccer software, the uncertainty of robot position is expressed by multiplication of standard deviation values of Gaussian distributions that model X and Y coordinate possible value sets. Robot position uncertainty value, in this case, can be interpreted as one fourth of the area of robot's most probable positions. Experimental results has shown, that when robot position uncertainty reaches value of 500*500 mm, it starts heavily influencing soccer game performance. That value has been set as threshold in a function that controls weather robot is "lost" during the game. When the threshold is reached the behaviour of the robot will change in attempt to stabilize the localization and decrease the position uncertainty value. Changes in robot's behaviour and detailed implementation is described in Section 3.3.

3.3 Implementation of localization enhancement technique

In order to allow the robot to increase the number of observed field landmarks scanning skill has been developed. When the necessity of increasing the number of observations occurs, scanning skill will be activated.

As has been mentioned in the Introduction part of this thesis, NAO robots have 25 degrees of freedom. 24 joints are controlled by a separate motors and 2 hip joints share one motor. For the scanning skill, two head joint values are controlled and updated. The figure illustrating position and full angular range of the joints is shown in the Figure 3.3

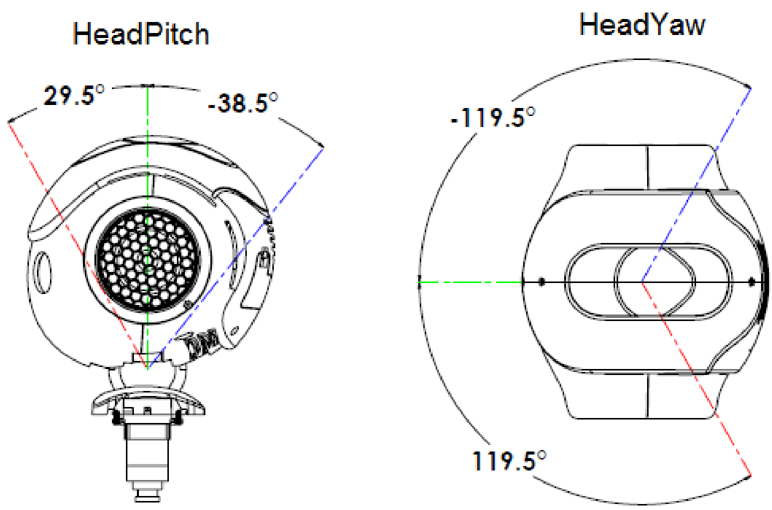


Figure 3.3: Head joints used in scanning skill [ald].

As can be seen from the Figure 3.3, the total range of head yaw is 239°. Angular value of X axes of NAO robot video camera field of view is equal to 60.97°. Together aforementioned values allow robot to observe a space of nearly 300°.

Scanning skill has been implemented in Python behaviour script. One fifth of a total head yaw joint range is added to the current value of this joint. The value of the head yaw joint in being updated not constantly (i.e. 30 times per second), but only every second, in order to give the robot some time to process new observations. When the head yaw joint reaches one of it's limits the sign of update operation changes. Head pitch joint is being updated each time when head yaw joint reaches it's maximum or minimum value. There are only two possible value for a head pitch joint in the scanning skill, being $0°$and $5°$, as introducing values smaller of greater than that will not contribute significantly to the increase of useful space for the search of field landmarks.

In parallel with scanning, the value of standard deviations of X and Y coordinate estimates is being checked. After the localization stabilizes and the area of robot possible locations is less than a threshold value, scanning skill will be disabled and robot will continue to execute other components of soccer behaviour.

Conclusion

This thesis gave an overview of some of the main topics that play important role in the autonomous robot soccer software.

A robust and accurate YUV color space based automatic color calibration has been proposed and described in details. Proposed technique uses K-means algorithm for color clustering, with the implementation of luminance analysis of sample data for the correct assignment of initial means. Cluster classification is performed in the YUV color space by comparing distances of cluster centroids to average values of desired color classes, corrected by previously performer luminance analysis. Propose method shown good results in distinguishing between the colors of interest and noise data, by providing robust and accurate look-up table.

Work, presented in this thesis, also explained the solution to white goal detection challenge of RoboCup 2015. RoboCup team Philosopher has implemented the proposed algorithm and will use it in action during the competition. Overview of the old and updated algorithms have been presented along with results.

Overview of the utilised localization system has been presented. Explanation of such concepts as Kalman Filter and eigen decomposition of covariance matrix have been provided. Localization enhancement approach, that aims to increase the number of observed field landmaks, has been proposed and explained.

Bibliography

[ald] Aldebaran nao documentation. `http://doc.aldebaran.com/1-14/family/robots/joints_robot.html`. Accessed: 12-05-2015.

[AMFM11] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011.

[BASK08] Morten Rufus Blas, Motilal Agrawal, Aravind Sundaresan, and Kurt Konolige. Fast color/texture segmentation for outdoor robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 4078–4085. IEEE, 2008.

[Bud12] David M Budden. *Applications of Machine Learning Techniques to Humanoid Robot Platforms*. PhD thesis, University of Newcastle, Australia, 2012.

[CCS03] Claudio Cusano, Gianluigi Ciocca, and Raimondo Schettini. Image annotation using svm. In *Electronic Imaging 2004*, pages 330–338. International Society for Optics and Photonics, 2003.

[Cel90] Mehmet Celenk. A color clustering technique for image segmentation. *Computer Vision, Graphics, and Image Processing*, 52(2):145–170, 1990.

[CKV13] M Emre Celebi, Hassan A Kingravi, and Patricio A Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1):200–210, 2013.

[DAJ08] Hasan Demirel, Gholamreza Anbarjafari, and Mohammad N Sabet Jahromi. Image equalization based on singular value decomposition. In *Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on*, pages 1–5. IEEE, 2008.

[Dav03] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410. IEEE, 2003.

[DH14] Kenneth Dawson-Howe. *A Practical Introduction to Computer Vision with OpenCV.* John Wiley & Sons, 2014.

[DOA10] Hasan Demirel, Cagri Ozcinar, and Gholamreza Anbarjafari. Satellite image contrast enhancement using discrete wavelet transform and singular value decomposition. *Geoscience and Remote Sensing Letters, IEEE*, 7(2):333–337, 2010.

[Gon09] Rafael C Gonzalez. *Digital image processing.* Pearson Education India, 2009.

[GWR06] Ketill Gunnarsson, Fabian Wiesel, and Raúl Rojas. The color and the shape: Automatic on-line color calibration for autonomous robots. In *RoboCup 2005: Robot Soccer World Cup IX*, pages 347–358. Springer, 2006.

[HKC08] Naomi Henderson, Robert King, and Stephan K Chalup. An automated colour calibration system using multivariate gaussian mixtures to segment hsi colour space. In *Proceedings of the 2008 Australasian Conference on Robotics & Automation (ACRA'2008)*, volume 6, 2008.

[JU97] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

[KMN+02] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, 2002.

[LLX+13] Xun Li, Huimin Lu, Dan Xiong, Hui Zhang, and Zhiqiang Zheng. A survey on visual perception for robocup msl soccer robots. *International Journal of Advanced Robotic Systems*, 1, 2013.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RCA93] Phillip Bording Richard C. Allen, Chris Bottcher. *E-book: Computational Science Education Project.* 1993. `http://www.phy.ornl.gov/csep/mc/node19.html`.

[RMSK13] Navid Razmjooy, B Somayeh Mousavi, Fazlollah Soleymani, and M Hosseini Khotbesara. A computer-aided diagnosis system for malignant melanomas. *Neural Computing and Applications*, 23(7-8):2059–2071, 2013.

[rob] RoboCup standard platform league. https://www.informatik.uni-bremen.de/spl/bin/view/Website/WebHome.

[RT99] Siddheswar Ray and Rose H Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. In *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques*, pages 137–143. India, 1999.

[SU07] Oleg Sushkov and William Uther. *Robot Localisation Using a Distributed Multi-Modal Kalman Filter*. INTECH Open Access Publisher, 2007.

[TC01] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118. ACM, 2001.

[WCR+01] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

Internet URLs were valid on 14.05.2015

License

Non-exclusive license to reproduce thesis and make thesis public

I, Anastasia Bolotnikova (date of birth: 12.12.1993),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   "Melioration of color calibration, goal detection and self-localization systems of NAO humanoid robots", supervised by Gholamreza Anbarjafari,

2. am aware of the fact that the author retains these rights.

3. certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 14.05.2015