

Real-Time Automatic Colour Calibration for NAO Humanoids

Anastasia BOLOTNIKOVA, Gholamreza ANBARJAFARI

Abstract: A challenge in NAO soccer robots is colour calibration. Good colour calibration can result in robust and accurate self-localization of the robot. Currently manual calibration is the only solution, which is used. In this paper, we are proposing an automatic real-time, accurate YUV colour space based colour calibration technique. In order to define average values for the desired colour classes namely orange, white, green and purple, a specified set of frames from the NAO camera are analysed. These average values are corrected by luminance analysis of a new frame and are passed to the K-means clustering algorithm as a set of initial means. In addition to these four values, a set of initial means of the K-means algorithm contains 16 values that are calculated in the following manner: the frame being processed is divided into 4 by 4 grids and the average value from every grid serves as an initial mean for K-means clustering. Consequently, colours of a similar type are combined into clusters. The final step of the proposed technique is cluster classification in which the average values of the desired colour classes are corrected by luminance analysis. As NAO cameras provide video streams in YUV format and the proposed algorithm uses this format there is no need for additional computational steps for conversation between the colour spaces. As a result, computational process is reduced compared to current techniques.

Keywords: automatic colour calibration; K-means clustering; localisation; NAO Robots; RoboCup

1 INTRODUCTION

NAO robots have been used in many real-world applications, as a consequence of which real-time processing is of importance [1-3]. One of the uses of NAO robots is in the Robot Soccer World Cup Standard Platform League (RoboCup SPL) [4, 5]. In this competition object recognition is highly dependent on real-time colour information processing. In order for a NAO robot to distinguish between different objects such as a ball, goal posts and field lines, it is necessary that similar colours within a frame are clustered into blobs. Future steps such as detection of important objects and localization are benefiting from these blobs. In order to accurately form these blobs in real-time it is necessary for NAO robot's camera to be colour calibrated.

The classical way to perform colour calibration is manually, when a human defines which pixel values belong to which colour class before every match. The advantages of the manual approach are its level of accuracy and low computation complexity. Retrieving classified colour values manually produces a look-up table that can quickly be read by a robot during the game. It is important to note that NAO robots have limited processing capabilities and their response time is also lower than many commonly used processing units, hence many pre-processing applications cannot be directly implemented on the robot during a match [6-8]. The main disadvantage of the manual approach is that it is very time consuming. This provides a motivation for RoboCup teams to develop automatic colour calibration as a part of their soccer software.

The NUBots team from the University of Newcastle, Australia has proposed HSI colour space based automatic colour calibration technique [9]. In this approach, the NAO camera acquires video frames and then they are converted from YUV to HSI colour space to retrieve Hue component. Because this component represents the angular colour value of the pixels. After the conversion of colour spaces, the hue component values are used in labelling the pixels, for instance, all the pixels whose hue component falls in the range of 100-140 degrees are clustered into so-called

green, resulting in reduction of dimensionality used for clustering into a single channel.

In the work of the NUBots team, the expectation maximization algorithm was introduced and applied to the histogram of the images in order to compute the parameters of multivariate Gaussian distributions of each colour class [10]. Then the HSI colour table is generated based on the standard deviations of the distributions of each colour class. They also suggested definition of so called "soft colour" classes, which represented the pixels that according to Gaussian distribution's standard deviation were equally probable to be classified as two different colour classes. Soft colour class blobs of pixels were only processed when the blob was sufficiently large.

The last step of this technique was the computation of a YUV colour table corresponding to the HSI colour table. The YUV colour table took approximately 10 minutes to be generated. The experimental results presented in this paper stated that it provided 100% blob formation rate and 83-94% object recognition rate depending on the type of the object. Its main drawback is the complexity introduced in order to convert each frame from YUV colour space into HSI colour space.

In addition to the expectation maximization algorithm, David M. Budden describes several other unsupervised colour clustering techniques using automatically generated colour tables [11]. His work focuses on segmentation methods based on mean shift and mode finding. He presents a detailed description of the algorithms such as K-means clustering, expectation maximization and mean shift along with the performance results for each method [12-14].

In addition to aforementioned clustering techniques, colour table generalization method based on Support Vector Machines (SVM) classification model was suggested in order to organize the clusters. This is achieved by removing extreme value, filling the holes within the clusters and making the shape and boundaries of a cluster smoother [15-18]. Budden's work concludes that the optimal method for automatic pixels clustering is the K-means algorithm without adopting SVM. K-means

clustering produces colour tables providing the best object recognition rates that it satisfies:

- classification sensitivity;
- the success of detecting of an object on the segmented frame;
- and the success of not detecting an object when it is not present in the frame.

Finding of the shortest distance between the cluster values and values of the corners of the RGB cube has been used for cluster classification. The main disadvantage of this method is that the unrealistic toxic colour values corresponding to the RGB colour space cube corners or regions close to the corners are making the classification biased in favour of more realistic colour shades.

The main contribution of this paper is to introduce real-time colour calibration for NAO robots and test it in the real-world scenarios. The rest of this paper is organized as follows: In section 2, the logic behind the selection of sample frames for data analysis and algorithm testing is explained, then the detailed explanation of luminance analysis and its application is presented, following the explanations of the clustering algorithm. Finally, the cluster classification methodology is explained and visual results of the algorithm output are presented in section 3.

2 PROPOSED REAL-TIME AUTOMATIC COLOUR CALIBRATION

A set of NAO camera frames were analysed in order to extract average colour values of the desired colour class in an environment under approximately average lightning. The proposed method implements luminance analysis of a frame to correct the Y component of the average colour values of the desired colour classes. The corrected values of the desired colour classes were used as a set of initial means for the K-means clustering algorithm along with 16 colour values extracted from the frame. Colour clusters were classified using average values of the desired colour classes corrected by luminance analysis.

The main requirement of the set of sample frames is that they should contain those objects of interest such as soccer field, goal posts, and ball and in addition, some undesired image that will serve as data that does not need to be classified at all.

Another restriction concerns the average luminance values, being shown by the Y color channel: Since the average luminance value of each frame is considered as the initial cluster center for the K-means clustering algorithm, the changes happening to it in the course of the iterations performed to optimize the within-class and between-class distances relates proportionally to the differences between the average luminance value of the individual frame being processed and that of the whole set. Thus, the average luminance value of the sequence should be as close as possible to that of the YUV representation used by default on NAO robots' cameras [19], which is 128, since the darkest and brightest colours lead to luminance values of 0 and 255, respectively.

Noticing the above considerations, 33 images captured from a soccer field with average luminance values ranging from 119 to 139 have been picked up as the test set, where the average luminance of the whole sequence is 130, which

is acceptably close to the aforementioned desired value, i.e. 128. It should be noted that apart from the luminance channel, the blue- and red-light contributions to the pixel values are born by the YUV colour coding system, which are represented by the U and V channels, respectively. Fig. 1 shows an example of the colour components in the YUV system, where the distribution of the luminance values can be illustrated through drawing a histogram of the Y channel, as shown in Fig. 2 for sample dark, normal and bright images.

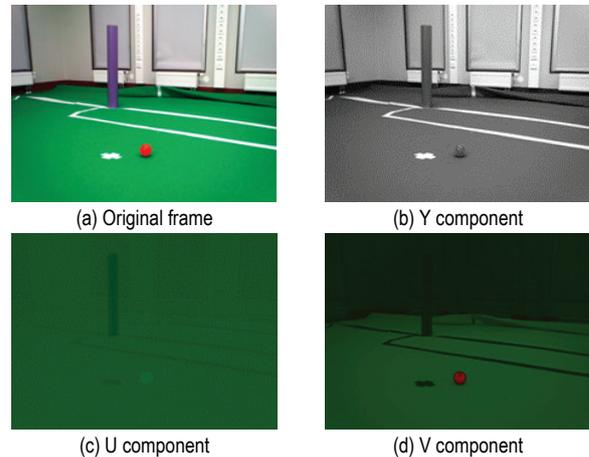


Figure 1 Examples of all components of a frame in the YUV colour coding

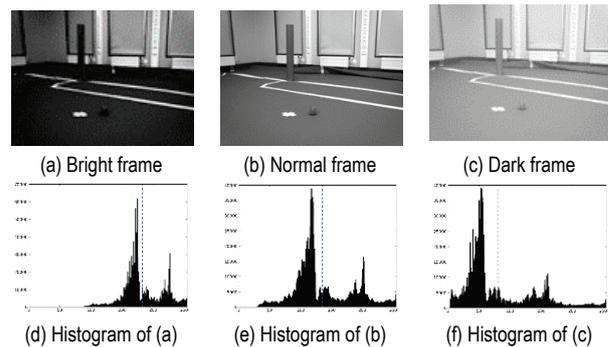


Figure 2 Frames, histograms and average values (dashed line) of Y component

In the proposed method, the histogram of a frame was used in calculating the average value of the luminance intensity. The histogram was chosen simply because looking at it, the luminosity of a frame can quickly be estimated. The computed average value was compared with the average Y value component and the difference between the two was generated. This difference was later used for computing the Y components of the four initially assigned means. The initial values for the four assigned means themselves were calculated using the average pixel values of the test set. Both the difference and the means were used for the K-means clustering algorithm.

In order to correct for the luminance of a frame, the difference between the average luminosity and that of the Y component was added to the Y components initially assigned means. The result of this operation shifted the initial mean values of brighter frames to the brighter side and darker ones to the darker side of the YUV colour cube.

For the K-means algorithm, a lot of care was taken into considering the k value, which determines the number of expected clusters. If the k value was set too high, then the algorithm would classify noise into several different

clusters, essentially assigning very similar colours into separate clusters. On the other hand, if the value was set too low, then not all of the desired clusters would be in the output.

There are many different approaches to calculate the optimal number for k in K-means clustering algorithm [20, 21]. One commonly used assumption is that the number of clusters should be approximately equal to half of the square root of number of data points. Given that, the number data points for frame acquired by NAO camera is 1280×960 then the number of the clusters should equal 784. Knowing that we only require 20 colours, the aforementioned number of clusters is an unnecessary large number.

Alternatively, in order to determine the optimal number of unique clusters, one may apply the Elbow method. The main idea behind this approach is to apply K-means clustering on the dataset for a range of k values, where k represents the total number of desired cluster. Next, a plot of sum of squared-error (SSE) is depicted versus each k value to produce an arm-shape graph for which the elbow of the plot corresponds to the best value of k . In other word, we want to maintain SSE as low as possible while increasing the number of clusters (k), but not to zero. Hence, the elbow on the plot usually represents where we start to have diminishing returns by increasing k . However, in the particular case of automatic colour calibration the Elbow method cannot be properly applied due to the wide variability of the input data. In practice, determining such optimum cluster number varies significantly for each different input frame. In our proposed method, the optimum number of clusters is experimentally shown to be 20.

The initial colour clustering of our proposed method can be summarized as follows: using luminance analysis to first enhance the sample test images of ‘soccer field’ object, 4 mean clustering that corresponds to the average colour classes in the YUV cube model is assigned. The remaining 16 clusters are obtained from the input data by dividing each frame into 4×4 grid and then calculating each grid’s mean respectively.

In case where the input image is contaminated with a noise, the input databased clustering is very useful to suppress all noise samples into the same cluster.

The algorithm in Python 2.7 uses the built-in Scikitlearn library function K-means [22]. At the beginning of original k-means algorithm, the means are selected randomly to produce initial SSE. As the algorithm continues to run, in each iteration a lower SSE is expected to achieve and hence the means are updated. This process continues until both the convergence and minimum SSE is reached.

However, in the special case, when the values of the initial means are fixed for every frame and there is no randomness in calculating the values for initial means, the algorithm will be deterministic and one iteration of running algorithm from the start until convergence is sufficient.

After the clustering is done by K-means algorithm, the clusters are then classified in to five classes. They are namely orange, white, purple, green and undefined. For this purpose, the initial four means, which were obtained by the use of YUV colour model, are used to serve the centroid of each corresponding cluster. Next, the Euclidean distance of each data point within the cluster is calculated

with respect to its corresponding centroid. The shortest distance is then compared to a threshold value. The value of this threshold is varying based on the colour class. If the value of the shortest distance is less than the threshold value, the cluster is labelled as a corresponding colour class. Otherwise, cluster is labelled as undefined.

Aforementioned threshold values are determined by conducting several tests on a set of sample frames. The best results were achieved when the values of the thresholds were set at 40, 60, 35, and 50 for orange, white, purple and green colour classes respectively. The block diagram of the proposed real-time colour calibration technique is shown in Fig. 3.

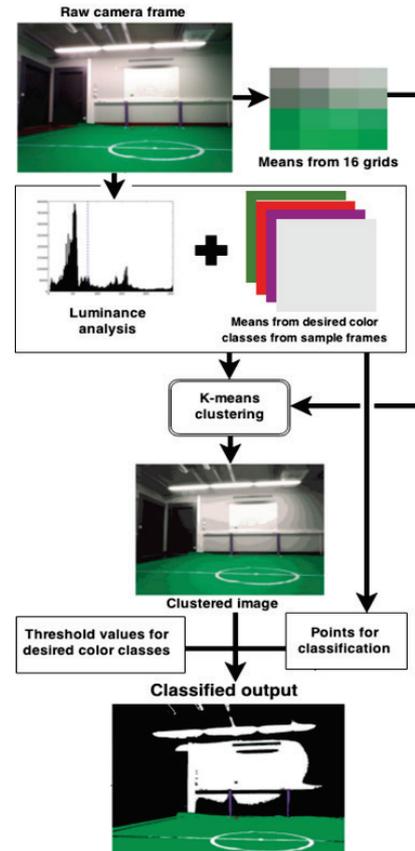


Figure 3 Automatic colour calibration algorithm scheme

The proposed algorithm is optimized by taking the number of means in K-means rule to be 20. This number was obtained by conducting many experiments in real-time scenarios. Due to implementation of the algorithm on the default colour space (YUV), there is no additional computational cost for the conversation between the YUV colour space and other colour spaces. This important feature of the proposed method reduces the computational cost compared to manual techniques used for colour calibration.

3 EXPERIMENTAL RESULTS AND DISCUSSIONS

Fig. 4 shows a few views of input and the respective output of the proposed automatic colour calibration algorithm. As can be seen, all objects of interest such as field, ball, goal posts and lines have been assigned to correct colour class and noise such as ceiling and table is mainly assigned to the undefined colour class.

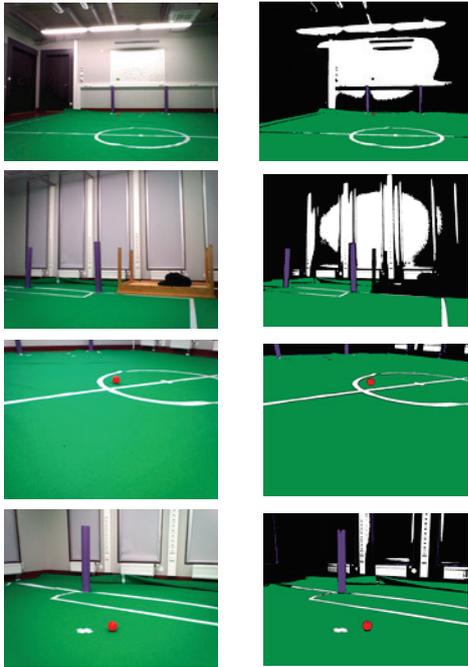


Figure 4 Input raw camera frames on the left and output of automatic colour classification algorithm on the right

Prior to the soccer match, several images of the field objects were taken and the proposed automatic colour calibration algorithm described in this paper was applied to them. The aim of techniques used in this algorithm is to minimize the probability of incorrect classification, however this remains a possibility (>10%). Hence, the results obtained were confirmed by human observation to eliminate the misclassification of the set of frames that were used to form the colour look-up table.

The complete automatic colour calibration algorithm is presented schematically in Fig. 5.

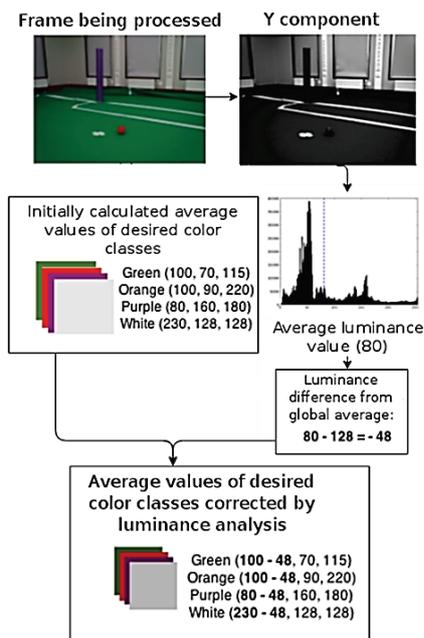


Figure 5 Examples of luminance analysis and its influence of the value of the initial means and the points for classification

4 CONCLUSION

In this paper, a fast and accurate automatic calibration framework is proposed and validated based on the YUV colour coding system. The clustering of the colour codes relies upon the guidelines of the K-means algorithm, with the average luminance values of individual frames as the initial cluster centroids, where the iterations aim at optimizing the within-class and between-class distances taking the differences between the average luminance values of the colour clusters with the associated centroids into account as criteria. The main contribution of the proposed method lies in its capability of leading to noticeably precise collections of colour codes resulting in clear distinctions between the actual colours sought from the outset and the ones corresponding to noisy points.

In future work additional techniques may be added to the proposed algorithm in order to minimize the probability of misclassification to a lower value in a way that there will be no need for any manual confirmation.

Acknowledgements

This work has been partially supported by Estonian Information Technology Foundation, Skype Technologies and Estonian Research Council Grant (PUT638). The authors would like to thank Dr. John Willcox for editing the English of this paper.

5 REFERENCES

- [1] Jokinen, K. & Wilcock, G. (2014). *Multimodal Open-Domain Conversations with the NAO Robot*. Natural Interaction with Robots, Knowbots and Smartphones, Springer New York, 213-224. https://doi.org/10.1007/978-1-4614-8280-2_19
- [2] Huang, S. J., Liu, S., & Wu, C. H. (2015). Intelligent Humanoid Mobile Robot with Embedded Control and Stereo Visual Feedback. *Journal of Mechanical Science and Technology*, 29(9), 3919-3931. <https://doi.org/10.1007/s12206-015-0838-y>
- [3] Bolotnikova, A., Demirel, H., & Anbarjafari, G. (2017). Real-time ensemble based face recognition system for NAO humanoids using local binary pattern. *Analog Integrated Circuits and Signal Processing*, 92(3), 467-475. <https://doi.org/10.1007/s10470-017-1006-3>
- [4] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., & Osawa, E. (1997). RoboCup: The Robot World Cup Initiative. *Proceedings of the first international conference on Autonomous agents / California*, 340-347. <https://doi.org/10.1145/267658.267738>
- [5] Kitano, H., Asada, M., Noda, I., & Matsubara, H. (1998). RoboCup: Robot World Cup. *IEEE Robotics & Automation Magazine*, 5(3), 30-36. <https://doi.org/10.1109/100.728221>
- [6] Anbarjafari, G., Ozcinar, C., & Demirel, H. (2015). Human Vision Inspired Based Image Illumination Enhancement by Using Local Singular Value Decomposition and Discrete Wavelet Transform. *International Journal of Electronics Communication and Computer Engineering*, 6(1), 14-19.
- [7] Chikane, V. & Fuh, C. S. (2006). Automatic White Balance for Digital Still Cameras. *Journal of Information Science and Engineering*, 22(3), 497-509.
- [8] Demirel, J., Anbarjafari, G., & Jahromi, M. N. S. (2008). Image Equalization Based on Singular Value Decomposition. *23rd IEEE International Symposium on Computer and Information Sciences / Istanbul*, 1-5.

<https://doi.org/10.1109/ISCIS.2008.4717878>

- [9] Henderson, N., King, R., & Chalup, S. K. (2008). An Automated Colour Calibration System Using Multivariate Gaussian Mixtures to Segment HSI Colour Space. *Proc. of the Australasian Conference on Robotics and Automation / Canberra*, 1-7.
- [10] Schiele, B. & Crowley, J. L. (2000). Recognition without Correspondence Using Multidimensional Receptive Field Histograms. *International Journal of Computer Vision*, 36(1), 31-50. <https://doi.org/10.1023/A:1008120406972>
- [11] Budden, D. M. (2012). Applications of Machine Learning Techniques to Humanoid Robot Platforms. *Doctoral dissertation*, University of Newcastle, Australia.
- [12] Hartigan, J. A. & Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. *Applied statistics*, 100-108. <https://doi.org/10.2307/2346830>
- [13] Cheng, Y. (1995). Mean Shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), 790-799. <https://doi.org/10.1109/34.400568>
- [14] Tzortzis, G. & Likas, A. (2014). The MinMax K-Means Clustering Algorithm. *Pattern Recognition*, 47(7), 2505-2516. <https://doi.org/10.1016/j.patcog.2014.01.015>
- [15] Cusano, C., Ciocca, G., & Schettini, R. (2003). Image Annotation Using SVM. *Electronic Imaging, International Society for Optics and Photonics*, 330-338.
- [16] Tong, S. & Koller, D. (2002). Support Vector Machine Active Learning with Applications to Text Classification. *The Journal of Machine Learning Research*, 2, 45-66.
- [17] Razmjooy, N., Mousavi, B. S., Soleymani, F., Khotbesara, M. H. (2013). A Computer-Aided Diagnosis System for Malignant Melanomas. *Neural Computing and Applications*, 23(7-8), 2059-2071. <https://doi.org/10.1007/s00521-012-1149-1>
- [18] Schlipfing, M., Salmen, J., Tschentscher, M., & Igel, C. (2014). Adaptive Pattern Recognition in Real-Time Video-Based Soccer Analysis. *Journal of Real-Time Image Processing*, 1-17.
- [19] A. Robotics, NAO software 1.14.5 documentation, <http://www.aldebaran-robotics.com> (07/02/2016).
- [20] Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). Constrained K-Means Clustering with Background Knowledge. *8th International Conference on Machine Learning / Williamstown*, Vol. 1, 577-584.
- [21] Ganganath, N., Cheng, C. T., & Tse, C. K. (2014). Data Clustering with Cluster Size Constraints Using a Modified K-Means Algorithm. *IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery / Shanghai*, 158-161. <https://doi.org/10.1109/CyberC.2014.36>
- [22] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, O., Weiss, R., Dubourg, V., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12, 2825-2830.

Contact information:

Anastasia BOLOTNIKOVA, Res. Asst.

iCV Research Lab, University of Tartu

Also at University of Montpellier

Institute of Technology, Nooruse 1, 50411 Tartu, Estonia

E-mail: nana@icv.tuit.ut.ee

Gholamreza ANBARJAFARI, Assoc. Prof.

iCV Research Lab, University of Tartu

Institute of Technology, Nooruse 1, 50411 Tartu, Estonia

Also at Department of Electrical and Electronic Engineering,

Hasan Kalyoncu University, Gaziantep, Turkey

E-mail: shb@icv.tuit.ut.ee