Hybrid Fuzzy Colour Processing and Learning

Daniel P. Playne, Vrushank D. Mehta,
Napoleon H. Reyes, and Andre L. C. Barczak

Massey University, Auckland, New Zealand
dan.p@dcmd.co.nz, v.mehta@direction.biz, n.h.reyes@massey.ac.nz,
a.l.barczak@massey.ac.nz

Abstract. We present a robust fuzzy colour processing system with automatic rule extraction and colour descriptors calibration for accurate colour object recognition and tracking in real-time. The system is anchored on the fusion of fuzzy colour contrast rules that operate on the red, green and blue channels independently and adaptively to compensate for the effects of glare, shadow, and illumination variations in an indoor environment. The system also utilises a pie-slice colour classification technique in a modified rg-chromaticity space. Now, colour operations can be defined linguistically to allow a vision system to discriminate between similarly coloured objects more effectively. The validity and generality of the proposed fuzzy colour processing system is analysed by examining the complete mapping of the fuzzy colour contrast rules for each target colour object under different illumination intensities with the presence of similarly coloured objects. The colour calibration algorithm is able to extract colour descriptors in a matter of seconds as compared to manual calibration usually taking hours to complete. Using the robot soccer environment as a test bed, the algorithm is able to calibrate colours with excellent accuracy.

Keywords. Computing with colours, fuzzy colour processing, computer vision, colour-object recognition.

1 Introduction

The process of characterizing a compendium of colours depicting an object in a dynamic environment for object recognition and tracking tasks needs to account for all confounding effects in the imaging system due to spatially varying illumination, presence of similarly coloured objects, lens focus, object rotation, shadows and sensitivities of the camera [1,2,3]. It is known that the colour descriptors transform non-linearly in the colour space due to these effects [1,2] and there are studies providing means of coping up with the problem [1,2,4,5,6,7]; however, the complexity of the calibration of the colour descriptors is proportional to the algorithms adaptability and robustness. In the human visual system, the qualities we assign to our perception of colour arise from our intuitive experience of colour. Colour perception underlies many complex processes that involve the photoreceptors in the retina as well as higher level processing mechanisms in the brain. Even to date, some of the intricacies in the mechanisms

M. Ishikawa et al. (Eds.): ICONIP 2007, Part II, LNCS 4985, pp. 386–395, 2008.

involved still remain to be unveiled. Nonetheless, findings in neurophysiological researches suggest that contrast computation precedes segmentation [8], and that the human colour perception system possess the ability to recognize colours adaptively and consistently despite changes in the spectral illuminant [9,1]. In this research, we mimic to a minimal extent the contrast computation mechanisms by employing the fusion of fuzzy colour contrast operations on the colour channels adaptively. Fuzzy logic is the computational paradigm of choice in this work as it lends itself amenable to solving problems involving many ambiguities and noise in the sensory inputs [10]. In addition, the system allows for the ease of use of linguistic terms in defining the colour contrast operations for the target colours at hand. As compared to other knowledge-based fuzzy colour processing systems [11,12], the proposed approach focuses on employing fuzzy colour correction steps first prior to colour classification rather than merely fuzzifying the colour sensed values to account for ambiguities in the definition of colour descriptors. Previously, in [6], the fuzzy colour contrast fusion algorithm was tested for its applicability to work in different colour spaces. It was reported that the algorithm successfully improved the colour classification task in the YUV, HSI and rg-chromaticity colour spaces. However, there is one major drawback in the system described; that is, the fuzzy colour contrast rules as well as the colour descriptors used were all derived through rigorous manual calibration, usually taking several hours to complete, especially for a wide range of target colour objects. In this research we improved and extended fuzzy colour contrast fusion by incorporating colour learning algorithms that automatically resolve the issue of finding the best combination of fuzzy colour contrast rules and fine-tuning the colour descriptors. Results show that the rules and colour descriptors extracted automatically by the system is superior to manually derived ones, and calculated only at a fraction of time of manual calibration. Lastly, the robot soccer environment can provide the ultimate test bed for the proposed algorithms as the game requires object tracking in a span of less than 33 msec., in a dynamic and adversarial environment.

2 General System Architecture

The proposed fuzzy colour processing system is comprised of a myriad of novel algorithms that are combined together. The system architecture depicted in Fig. 1 is used for the automatic fine-tuning of the colour descriptors and for the generation, evaluation and discovery of the best combination of fuzzy colour contrast rules. Once all the rules and colour descriptors are extracted and refined, the system generates a look-up table of all possible colours that can be seen by the system (16.7 million pre-classified colours) for real-time colour object recognition and tracking. An overview of the functionalities of the main components and their interdependencies is discussed in this section while the next succeeding section zeroes-in on each of the components of the system, providing more details on them.

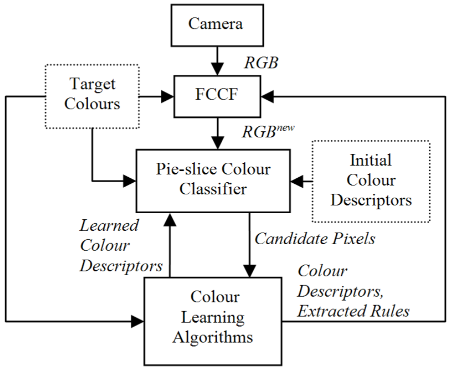

Fig. 1. General system architecture

At the top of the diagram (Fig. 1) is the camera component which returns the colour tri-stimulus in R, G and B values. The colour sensed values are then fed to the Fuzzy Colour Contrast Fusion (FCCF) algorithm which applies colour corrections on the colour tri-stimulus to allow for more accurate colour discrimination. FCCF however relies on the fuzzy colour rule-base and fine-tuned colour descriptors produced by the Colour Learning Algorithms, namely the Motion-based Predictive Colour Learning algorithm (MPCL) and the Colour Contrast Rule Extraction algorithm (CCRE). Lastly, the system employs the pie-slice colour classification technique which receives the corrected R, G, B values from the FCCF component and the refined colour descriptors from the Colour Learning Algorithms. The pie-slice colour classifier determines if the pixel being examined belongs to any of the target colour objects.

3 The Algorithms

3.1 Fuzzy Colour Contrast Fusion

It is adamant that the colours depicting an object must be adaptively corrected based on the relative illumination conditions of the environment they are exposed to. FCCF adaptively performs colour correction by either colour contrast enhancing or degrading the colour channels at different levels of intensity, prior to classifying the sensed colour tri-stimulus. For each target colour at hand (e.g. pink, orange), the RGB components will receive a unique set of fuzzy colour contrast operations.

Enhance or degrade operations are implemented via non-linear functions [3]. Figure 2 depicts the curve exhibiting the contrast enhance operator applied in different levels (1x, 2x, 3x, etc). The input signal can be any of the normalized RGB components within the range $[0, 1]$. In turn, the function amplifies input values greater than 0.5; and otherwise, attenuates it [10].

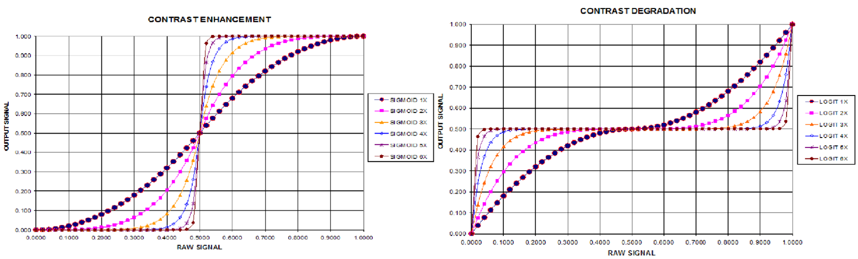

Fig. 2. On the left is the Contrast Enhance Operator, while on the right is the Contrast Degrade Operator

On the other hand, the contrast degrade operator performs the opposite fashion [1,6], as depicted in the curve in Fig. 2. It amplifies all signals less than 0.5; and otherwise, attenuates it. FCCF works on any desired colour space, provided that the colour pixels are expressed in terms of polar coordinates so that colour contrast rules can be applied selectively on colour pixels that fall within a pie-slice region classified as the general target colour region or colour contrast constraints [6].

3.2 rg Pie Slice Classifier

Colour recognition algorithms work by taking a single pixel and determining if it is of any of the colours specified by the current colour descriptors [5]. This classifier works in the rg-chromaticity colour space because it helps to reduce the effects of illumination intensity [1,6].The algorithm takes as input a pixel in RGB format and converts it into the rg colour space.

Once the pixel has been converted into rg-Hue and rg-Saturation [1,6], it can simply be checked to see if it is within the bounds of the colours as defined by the pie-sliced colour descriptors.

The algorithm does not have time to calculate the rg-hue and rg-saturation values for each pixel as the inverse tangent and square root calculations take too long, so look-up tables (LUT) were created to improve the performance. The program creates this LUT on initialization by calculating the rg-Hue and rg-Saturation values for every possible combination of RGB values. These look-up tables take several minutes to build at the beginning of the program but significantly speed up the classification process ($<$ 33msec.) [7]. When a pixel is classified, the algorithm simply has to access the look-up table and the positions of the RGB values to discover the rg-Hue and rg-Saturation values.

3.3 Motion-Based Predictive Colour-Learning Algorithm (MPCL)

The colour discrimination ability of FCCF comes with a price. It requires a rich set of colour descriptors for each target colour, namely the boundaries for rg-Hue, rg-Saturation and contrast constraint angles, and a set of colour contrast

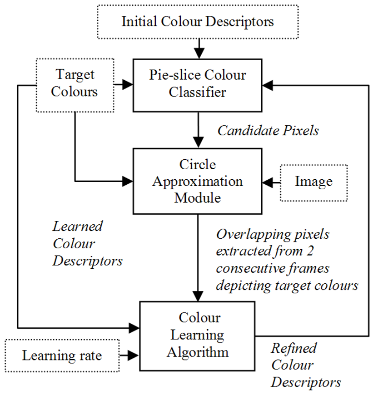

Fig. 3. The MPCL algorithm

rules. These parameters were previously extracted manually; involving an operator adjusting the values by hand until the results of the colour classification pinpoints the target colour unambiguously. However, hand calibration does not guarantee finding the optimal settings for the colour recognition system [4], and so this is the problem the MPCL algorithm is addressing. It automates the calibration process with superior calibration performance. In general, MPCL looks at two successive frames, extracting the best candidate pixels representing the object and fine-tuning the colour descriptors based on those pixels. For the purpose of easily finding the candidate pixels, a circularly shaped object was used during the calibration process. Nonetheless, after the system learns all the colour descriptors, the objects for tracking can come in any shape.

The series of steps for learning the colour descriptors are shown in Fig. 3. Initially, a broad set of colour descriptors is used by the pie-slice classifier to find the set of candidate pixels representing the target object. In turn, these pixels are fed into a circle approximation module that searches for the largest, most circular patch of colour present on the board. It calculates a formula approximating the circle by calculating the centre of the colour patch and averaging the extreme x and y values to approximate the radius of the circle. Two circle formulas will be generated for two consecutive images and the overlap of the two circles will be calculated. Once this overlap has been found the algorithm will find every pixel inside the area and filter them with the broad colour classifier to ensure that the approximated area does not include any non-colour pixels. Next, it takes all of the filtered pixels and record the extreme values for the rg-Hue and rg-Saturation values of the pixels to find the smallest possible pie-slice area that would classify every pixel inside the overlapping area. Once these extreme values have been calculated, the algorithm uses a moving average technique to adjust the actual colour descriptor parameters. The amount each set of extreme values affects the actual parameters depends on the learning rate.

Circle Generation The circle generated for each colour patch is generated by averaging the height and width of the circular patch from the centre of the circle. Once all of the pixels in the patch have been found, a centre of gravity equation is used to find the centre of the patch:

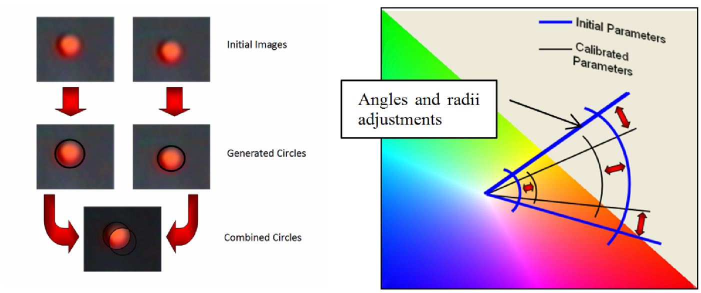

Fig. 4. On the left is the extracted object colour pixels from two consecutive frames. On the right is the calibration of colour descriptors.

$$x_{centre} = \sum_{i=0}^{n} x_i \qquad y_{centre} = \sum_{i=0}^{n} y_i \tag{1}$$

Once the centre of the patch has been located, the height and width of the patch are found:

$$height = max(x_{centre}, y) \qquad width = max(x, y_{centre}) \tag{2}$$

Then the radius is calculated with the following equation:

$$radius = \frac{height + width}{4} \tag{3}$$

The centre and radius of the circle has now been found so the next part of the algorithm can run. The learning algorithm works on a moving average system combined with a decaying learning rate algorithm. The algorithm will run for a set number of iterations and keep moving average of the maximum and minimum rg-Hue and rg-Saturation:

$$rgHue_{max} = \frac{rgHue_{max}(i-1) + max(rgHue)}{i} \tag{4}$$

$$rgHue_{min} = \frac{rgHue_{min}(i-1) + min(rgHue)}{i} \tag{5}$$

$$rgSat_{max} = \frac{rgSat_{max}(i-1) + max(rgSat)}{i} \tag{6}$$

$$rgSat_{min} = \frac{rgSat_{min}(i-1) + min(rgSat)}{i} \tag{7}$$

The idea of the algorithm is to move a robot with a colour patch or roll a ball around the board to calibrate the colour. Because the object will move through all of the different illumination conditions, the algorithm will calibrate the colour classifier to work for the entire board, accounting for all possible illumination conditions.

3.4 Colour Contrast Rule Extraction (CCRE)

Algorithm 1. $\boldsymbol{CCRE}(image, targetbounds)$

1. For each target object calculate an individual score: $score_i = \frac{hits_i}{area_i}$
   - if $hits_i < \frac{1}{n}\ area_i$ then $score_i = 0$; where $n = 4$ (empirically found)
2. Calculate average score:
   - $avescore = \frac{\sum_{i=1}^{ntargets} score_i}{ntargets}$; where: $ntargets$ is the number of targets.
3. Calculate a general score:
   - $genscore = \frac{Totalhits}{Totalhits+Totalmisses}$
4. Final score:
   - $finalscore = (0.6\, avescore) + (0.4\, genscore)$
5. Adjust score to account for misclassifications:
   - $if(Totalhits > 0)$
   - $finalscore = finalscore - (\frac{Totalmisses}{Totalhits})$

A colour contrast rule uniquely defines what combination of contrast operations and what levels of contrast operations will be applied to the red, green and blue channels. As indicated in Table 1, a light blue colour will receive a combination of contrast degrade, of level 1 on the red channel, contrast enhance, of level 1 on the green channel and no contrast operation on the blue channel. There are only 2 possible contrast operations: either to enhance or degrade. It is also possible that the colour channel does not require any contrast operation at all (i.e. no operation). Moreover, only 3 possible levels of contrast applications were considered (i.e. 1x, 2x, 3x). For example, a contrast level of three means that the contrast operator will be applied 3 times to the colour channel, using the output of each application as an input to the next. For each colour channel, there are 7 possible combinations of contrast operations: (enhance/degrade) - 3 possible levels each, no operation). Altogether, considering all 3 colour channels (RGB), there are 343 possible fuzzy colour contrast rules that can be applied for any target colour.

The algorithm hunts for the best rule by supplying the FCCF module with a generated colour contrast rule and using the pie-slice classifier for extracting the pixels representing the target colour object. It then counts the number of hits and misclassifications by examining the x and y-coordinates of those pixels if they fall within the actual boundaries of the target objects. Lastly, a formula for calculating the score for each rule is used:

The colour discrimination ability of FCCF comes with a price. It requires a rich set of colour descriptors for each target colour, namely the boundaries for rg-Hue, rg-Saturation and contrast constraint angles, and a set of colour contrast rules. These parameters were previously extracted manually; involving an operator adjusting the values by hand until the results of the colour classification pinpoints the target colour unambiguously. However, hand calibration does not guarantee finding the optimal settings for the colour recognition system [4], and so this is the problem the MPCL algorithm is addressing. It automates the calibration process with superior calibration performance

4 Experiments and Analysis

The MPCL has been tested on images of a circular colour patch in the centre of the board with promising results. MPCL was given a very broad set of parameters describing the colour and a series of images of this colour patch on the board. Several experiments using different coloured patches were run to make sure the algorithm works correctly with all kinds of colour classifier parameters. The algorithm was also tested by being set to calibrate one colour in the presence of other circular patches having relatively similar colours on the board.

The two images in Fig. 5 show a sample performance comparison results of the colour recognition algorithm using hand-calibrated settings and settings found by the MPCL algorithm. These images exhibit two circular colour patches, one orange and one green. The hand calibrated settings cause approximately 500 misclassifications whereas the MPCL algorithm settings cause 16 misclassifications. Fig. 6 shows an example of colour classification results for light blue targets. Details of classification results can be found in tables 1 and 2.

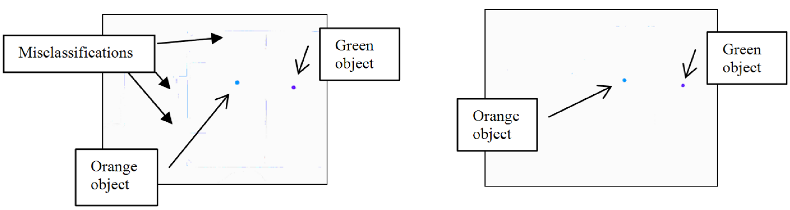

Fig. 5. MPCL results: on the left is the manual result. On the right is the system result.

Table 1. Manually derived colour contrast rules and their scores

Colour Name	Rank	Contrast Operation			Score	Hits	Misses
		R	G	B			
Yellow	0th	0	2	-2	0.48	2410	458
Green	8th	-1	2	-2	0.45	3252	608
Pink	4th	1	-1	0	0.59	1714	99
Purple	3rd	1	1	0	0.54	2629	320
Violet	0th	0	1	1	0.4	1873	415
LightBlue	15th	-1	1	0	0.63	2702	135

Table 2. System generated colour contrast rules and their scores

Colour Name	R	G	B	Score	Hits	Misses
Yellow	3	1	-2	0.65	2104	68
Green	0	-1	-3	0.55	3313	383
Pink	1	-1	0	0.59	1714	99
Purple	0	1	-3	0.57	2777	314
Violet	1	1	2	0.53	2535	497
LightBlue	0	3	1	0.67	2758	68

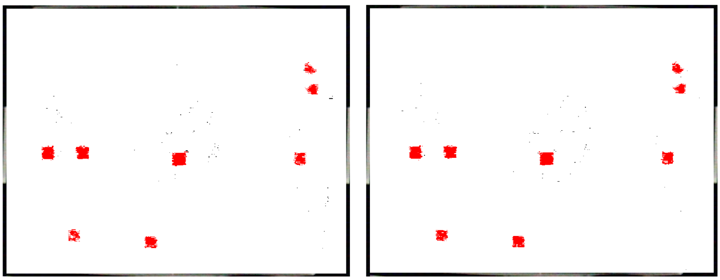

Fig. 6. Light blue targets: on the left is the result for the manual calibration, on the right is the result for the system calibration

Next, the CCRE was tested on 6 colours (i.e. pink, violet, etc.). Six colour patches per colour were placed at varying illumination intensities on the robot soccer field. The objective of the tests was to let the CCRE algorithm to extract the colour contrast rules that will accurately recognise all the patches simultaneously. The encoding of the contrast operations for Tables 1 and 2 are as follows: (+) for enhance operation, (-) for degrade operations, 0 for no operation and nonzero for any level of contrast application on the colour channel. It can be seen from Tables 1 and 2 that the system generated rules from CCRE always gives superior performance. The score and hits of the system rules were always greater than or equal to the manually generated ones. On the other hand, the misses could be greater sometimes, but we verified that such numbers never induce ambiguities during the object recognition task. Lastly, we used all the acquired colour descriptors and colour contrast rules to generate a look-up table (LUT) for real-time colour object recognition for the robot soccer game. The generated LUT guarantees that the robots can be recognised and tracked perfectly during the game without ambiguities in real-time.

5 Conclusions

We have successfully devised and tested a novel motion-based predictive colour learning algorithm (MPCL) and a colour contrast rule (CCRE) extraction algorithm that integrates with the Fuzzy Colour Contrast Fusion algorithm and pie-slice colour classifier. Results prove that the hybrid system is extremely faster and more accurate than hand-calibrated colour descriptors and colour contrast rules, while at the same time robust to changes in the illumination conditions. Lastly, by storing colour classification results in a look-up table, the hybrid vision system presented becomes very effective for the FIRA and Robocup real-time robot soccer vision systems.

References

1. Reyes, N.H.: Colour-Based Object Recognition Analysis and Application. PhD thesis, De La Salle University (2004)
2. Stonier, P.J.T.R.J., Wolfs, P.J.: Robustness of color detection for robot soccer. In: Seventh International Conference on Control, Automation, Robotics and Vision, ICARCV 2002, Singapore, pp. 1245–1249 (2002)
3. Reyes, N.H., Dadios, E.P.: Dynamic color object recognition. Journal of Advanced Computational Intelligence 8(1), 29–38 (2004)
4. Weiss, N., Jesse, N.: Towards local vision in centralized robot soccer leagues: A robust and flexible vision system also allowing varying degrees of robot autonomy. In: Proceedings of FIRA World Congress, Busan, FIRA (2004)
5. Klancar, G., Orqueda, O., Robust, D.M.R.K.: efficient vision for mobile robots control application to soccer robots. Electrotechnical Review 68(5), 305–312 (2001)
6. Reyes, N., Messom, C.: Identifying colour objects with fuzzy colour contrast fusion. In: 3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems, and FIRA RoboWorld Congress, CIRAS 2005, Singapore (2005)
7. McNaughton, M., Zhang, H.: Color vision for robocup with fast lookup tables. In: 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, Taipei, IEEE, Los Alamitos (2003)
8. Hurlbert, A., Wolf, K.: Color contrast: a contributory mechanism to color constancy. Progress in Brain Research 144, 147–160 (2004)
9. Funt, B., Barnard, K., Martin, L.: Is color constancy good enough? In: 5th European Conference on Computer Vision, pp. 445–459 (1998)
10. Ross, T.: Fuzzy Logic with Engineering Applications. McGraw-Hill, Inc., Singapore (1997)
11. Hildebrand, L., Fathi, M.: Knowledge-based fuzzy color processing. IEEE Trans. on Sys., Man. and Cybernectics-part C 34, 499–505 (2004)
12. Montecillo-Puente, F., Ayala-Ramirez, V., Perez-Garcia, A., Sanchez-Yanez, R.: Fuzzy color tracking for robotic tasks. In: IEEE International Conference on Systems, Man and Cybernetics, 2003, vol. 3, pp. 2769–2773 (2003)