



Universitat de Lleida

Document downloaded from:

<http://hdl.handle.net/10459.1/62843>

The final publication is available at:

https://doi.org/10.1007/978-3-319-00563-8_4

Copyright

© Springer International Publishing Switzerland, 2013

Evaluation of the color-based image segmentation capabilities of a compact mobile robot agent based on Google Android Smartphone

Dani Martínez, Javier Moreno, Davinia Font, Marcel Tresanchez, Tomàs Pallejà, Mercè Teixidó, and Jordi Palacín

Computer Science and Industrial Engineering Department, University of Lleida, 25001 Lleida, Spain

{dmartinez, jmoreno, dfont, mtresanchez, tpalleja, mteixido, palacin}@diei.udl.cat

Abstract. This paper presents the evaluation of the image segmentation capabilities of a mobile robot agent based on a Google Android Smartphone. The mobile agent was designed to operate autonomously in an environment inspired in a soccer game. The pixel-based color image segmentation capabilities was performed by testing different two-dimensional look up tables created from two-dimensional color histograms of the objects appearing in the scenario. The best segmentation alternative was obtained when using of a two-dimensional look up table based on the H (hue) and V (value) color image description. The final conclusion is that a Google Android Smartphone has enough potential to define an autonomous mobile robot agent and play a game that requires color-based image segmentation capabilities.

Keywords. segmentation, mobile robot agent, Android, Smartphone.

1 Introduction

The development of a mobile robot ready to play a game is an application that requires the application of challenging technologies. The main goal of this work is the evaluation of the color-based image segmentation capabilities of a compact mobile robot based on Google Android Smartphone. The application scenario is inspired in the common game called soccer (in USA), football (in Europe), or futbol (in Spanish). This paper proposes the development of a comparative study

on fast color-based image segmentation performed in a Google Android Smartphone. This paper has been developed in the direction of the creation of a full autonomous mobile robot agent based on this platform and with different potential game capabilities. In this paper, the mobile robot has to operate in a predefined scenario with the following color-based objects (Figure 1): floor, in green color; the field line limits, in white color; the ball, in a yellow main color; the goals, in red (red team) and blue (blue team) colors; and the mobile robots, also in red and blue colors. The main hypothesis is that current Smartphones have enough capabilities to perform fast color-based image segmentation and autonomously control the evolutions of a game oriented mobile robot.



Fig. 1. Image showing the ground field, the ball, one goal and two mobile robot agents.

2 Background

The Android operating system is based on the Linux core and was developed by the Open Handset Alliance, led by Google. The creation of applications for Android based Smartphones and Tablets can be performed with the free Android SDK [1] that is based on the high level Java language with other specialized functions optimized for the specific hardware written in C. The main difference with the standard Java programming style is the use of events instead of a classic main function. The Integrated Development Environment (IDE) used in this work was the Eclipse 3.7.1 Indigo [2], the most popular IDE for Android which also includes the Java developer package.

2.1 Soccer mobile robots

The Robocup [3] domain and their leagues are the most popular reference for mobile robots which are designed to play a team game. This game domain define robots sizes and designs and up to 8 colors to define objects. Also it has different leagues; some of them are Small Size league, Medium Size league and Humanoid

league. Nevertheless, the fulfillment of the rules defined by the Robocup domain is not the main goal of this work. Instead, we have defined a simplified game environment (Figure 1) containing one ball, two robots and two goals.

2.2 Fast color-based image segmentation

The image segmentation techniques applied in this paper are inspired in [4] which proposed an optimized threshold algorithm applied to the YUV color space to perform image segmentation and object color classification with a limited hardware; and in [5] and [6] which proposed the use of a YUV/RGB LUTs or other derived color spaces for color indexing based on specific calibration techniques to improve color segmentation and avoid the problems caused by lightning conditions. In this direction, this paper proposes the use of a two-dimensional LUT to define a fast segmentation methodology in order to take advantage of the limited number of colors defined in the proposed game.

3 Mobile robot implementation

Figure 1 shows an image of the mobile robot implementation based on a Google Android Smartphone. The mobile robot is a cylindrical box that supports the Smartphone and an additional battery and also contains the wheels, DC motors, and one electronic board to control the motors. The rear camera of the Smartphone is used to obtain images of the soccer playfield in low resolution (320x240 pixels) and high color quality. In this mobile robot the Smartphone analyzes the images acquired and sends basic commands: forward, back, turn left, turn right to the electronic board to drive the evolutions of the DC motors of the mobile robot. The additional battery is used to power directly the DC motors, the electronic board, and the Smartphone trough the USB connection. Then, the Smartphone can operate even when the external battery of the mobile robot is fully discharged, for example, to send a message to request a battery change.

The Smartphone used in this proposal is the HTC Sensation powered by a dual-core 1.2 GHz processor, 768 MB of RAM memory, and Android 4.0.3. The Smartphone has several embedded sensors: GPS, ambient light sensor, digital compass, three-axial accelerometer, gyroscope, multi-touch capacitive touch screen, proximity sensor, microphone, and a frontal and a rear camera. An important feature of one Smartphone is its high resolution screen that will be very useful to show some real-time information of mobile robot internal operation. The Smartphone device runs an agent APP which includes three basic capabilities: the main functional threads, the graphical interface, and the sensor managers. There are many specific threads defined, one thread has to receive the camera images and implement different color-based segmentation functions, two threads sends

and receives high-level movement orders with the electronic board that control the motors of the mobile robot. The definition of different threads in Android requires the initialization of an instance of extended thread class and override the `run()` function which contains the code that the thread will execute. Normally this function contains the main code with the computational load of the application.

3.1 Image processing implementation

Once the camera of the Smartphone is initialized a preview callback function is called each time that a new image is generated. This function is used to implement the color-based image segmentation capabilities of the mobile robot agent. Figure 2 shows the execution timeline required by the preview callback function to process the current image. The blue line in the figure shows an irregular behavior for the frame processing time. This effect is caused by the internal implementation of the Android camera functions and probably by other background applications running simultaneously in the Smartphone. After a careful analysis, the cause of the problem was identified as the Android's garbage collector module that operates very frequently to dynamically allocate different memory requested.

This problem can be solved on Android version 2.2 (or later) by using a new callback implementation `setPreviewCallbackWithBuffer` that avoids memory demands for each frame using the same static buffer to allocate image data, and thus, preventing frequent calling to the garbage collector (see Fig. 2 red line).

3.2 Influence of the image color space

The development of this work requires the use of different color spaces to evaluate with one has the best segmentation performances when applying a two dimensional LUT to classify the colors of the images. Figure 3 shows the histograms of the time lapse between consecutive calls to the preview callback function (with static buffer) for different color spaces. The red histogram shows the time lapse values when applying a LUT by using the original YUV color space image. This process is performed in most of the cases by spending 30 ms. The green histogram shows the time lapse spend when converting the original YUV (equivalent to YCbCr) color image into the RGB color space and then applying a LUT to classify the pixels of the image. This process is performed in most of the cases by spending 40 ms. Finally, the blue histograms showing the callback processing time when converting the original YUV image into the HSV color space and then apply a LUT to individually classify the pixels of the images. This process is performed in most of the cases by spending 67 ms. These poor results are caused by the difficult conversion between of the YUV and HSV image color spaces using high level procedures.

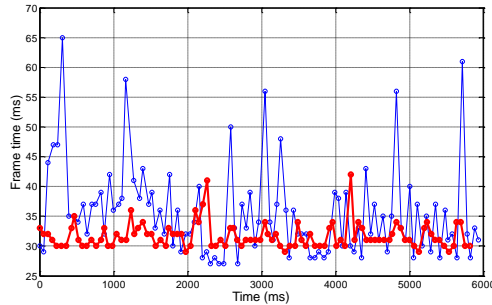


Fig. 2. Time lapse between consecutive calls to the preview callback function (blue line) and to the preview callback function with static buffer (red line).

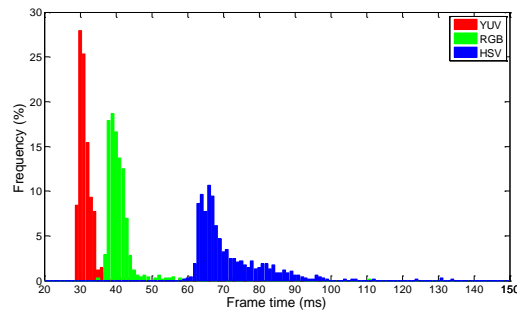


Fig. 3. Histograms of time lapse between consecutive calls to the preview callback function when classifying images in the YUV, RGB, and HSV color spaces.

4 Fast image segmentation

The proposed methodology to perform fast image segmentation is the definition of a LUT for each specific color space used. In this analysis, the LUTs are computed off-line and the evaluation of the classification performances requires the development of the following steps: 1) acquire several images of the game playfield with the rear camera of the Smartphone, 2) manually label the different elements and objects of the images, 3) compute a two-dimensional histogram of all pixels of the same color, and 4) combine the histograms of the different objects in a two-dimensional LUT. Figure 4 also shows three examples of two-dimensional histograms corresponding to the ball and to blue and red color. These histograms have been computed by using the RG color spaces with 256 bins in each axis. These histograms can be joined in a unique LUT according to the different pertinence probabilities of each object.

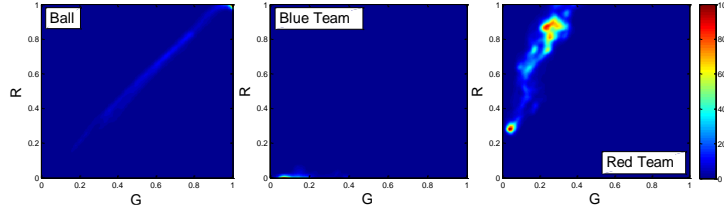


Fig. 4. Example RG histograms computed with the pixels corresponding to the ball, blue robot and goal team, and red robot and goal team (see figure ;Error! No se encuentra el origen de la referencia.).

Table 1 shows the results of a complete analysis performed to evaluate different color space combinations in the definition of a two-dimensional LUT optimized for the high quality images obtained with the Smartphone. The number of bins was limited to 32 after a trial and error test to simplify the final implementation in the Smartphone. The relative error shown in Table I is computed as the difference between a manually labeled image and the same image automatically classified with the LUT which was automatically obtained from the selected pixels of different images. The results in the column labeled as H have been obtained by creating the LUT from the conventional histograms and the results in the columns labeled as \hat{H} have been obtained after convolving the histogram with a disk of 2 pixels radius (to expand the color-space area covered by each object in the histogram).

The analysis performed in this work has been developed by considering all combinations between the color spaces RGB, Gray, HSV, and YUV (labeled as Y-Cb-Cr). Additionally, the areas outside the game playfield field have been discarded and not compared. Figure 5-a shows the best LUT automatically obtained in this experiment, corresponding to the HV color spaces (case: histogram not convolved), and figure 5-b shows the image classification results obtained with this LUT. The average segmentation error obtained after comparing pixel by pixel the manual and automatic classification performed was 0.87%. The next best alternatives were based on the use of the HY, and the HG color spaces. Probably these results are strongly correlated with the high quality of the image acquired with the camera and all auto-adjusted photo-image features included in the Smartphone. The LUT shown in figure 5-a is very small (32x32 bins) and can be easily incorporated into the application of the Smartphone.

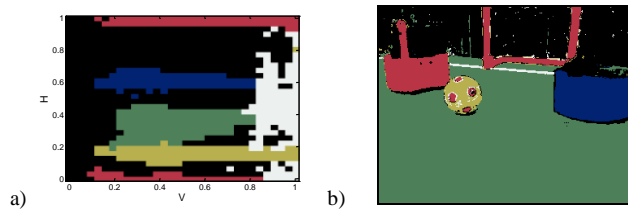


Fig. 5. Best segmentation LUT obtained in the HV color space (a) and its image classification results (b).

Table 1. Automatic image classification results.

| Color Spaces | Object segmentation relative error (%) | | | | | | | | | | Average error (%) | |
|--------------|--|-----------|--------------|-----------|--------------|-----------|-----------------|-----------|------------------|-----------|-------------------|-----------|
| | <i>Ball</i> | | <i>Field</i> | | <i>Lines</i> | | <i>Red Team</i> | | <i>Blue Team</i> | | <i>H</i> | <i>H̃</i> |
| | <i>H</i> | <i>H̃</i> | <i>H</i> | <i>H̃</i> | <i>H</i> | <i>H̃</i> | <i>H</i> | <i>H̃</i> | <i>H</i> | <i>H̃</i> | <i>H</i> | <i>H̃</i> |
| R-G | 1.18 | 1.38 | 3.34 | 3.43 | 0.91 | 0.92 | 1.15 | 0.73 | 1.63 | 1.68 | 1.64 | 1.63 |
| R-B | 5.35 | 7.55 | 2.33 | 2.20 | 0.69 | 0.65 | 5.74 | 7.56 | 0.94 | 0.85 | 3.01 | 3.76 |
| R-H | 0.41 | 0.77 | 2.19 | 2.08 | 0.79 | 0.73 | 0.56 | 0.54 | 0.57 | 0.51 | 0.90 | 0.93 |
| R-S | 2.84 | 4.41 | 2.67 | 2.45 | 0.71 | 0.66 | 3.35 | 4.25 | 1.26 | 1.13 | 2.17 | 2.58 |
| R-V | 2.16 | 2.48 | 2.97 | 3.07 | 0.90 | 0.93 | 1.88 | 2.42 | 1.26 | 1.26 | 1.83 | 2.03 |
| R-Y | 1.09 | 1.49 | 3.42 | 3.31 | 0.72 | 0.76 | 1.00 | 0.79 | 1.67 | 1.62 | 1.58 | 1.59 |
| R-Cb | 0.46 | 0.22 | 2.29 | 1.91 | 0.89 | 1.07 | 1.17 | 1.45 | 0.70 | 0.65 | 1.10 | 1.06 |
| R-Cr | 1.03 | 1.38 | 3.21 | 3.21 | 0.76 | 0.83 | 0.94 | 0.73 | 1.54 | 1.65 | 1.50 | 1.56 |
| R-Gray | 1.00 | 1.42 | 3.28 | 3.31 | 0.71 | 0.76 | 1.02 | 0.76 | 1.61 | 1.63 | 1.52 | 1.58 |
| G-B | 0.95 | 0.75 | 2.32 | 2.55 | 0.83 | 0.71 | 1.96 | 2.12 | 0.91 | 0.78 | 1.39 | 1.38 |
| G-H | 0.42 | 0.80 | 2.19 | 2.08 | 0.65 | 0.63 | 0.75 | 0.60 | 0.45 | 0.45 | 0.89 | 0.91 |
| G-S | 6.41 | 4.20 | 14.46 | 13.11 | 1.43 | 1.41 | 8.76 | 9.68 | 1.57 | 1.59 | 6.53 | 6.00 |
| G-V | 3.31 | 3.64 | 3.04 | 2.83 | 1.70 | 1.89 | 3.21 | 2.54 | 3.24 | 2.69 | 2.90 | 2.72 |
| G-Y | 3.26 | 2.75 | 3.61 | 3.73 | 1.38 | 1.39 | 0.91 | 0.73 | 2.31 | 2.80 | 2.29 | 2.28 |
| G-Cb | 1.08 | 0.27 | 5.50 | 6.26 | 0.63 | 0.60 | 4.94 | 5.13 | 0.72 | 0.54 | 2.57 | 2.56 |
| G-Cr | 1.00 | 1.46 | 3.03 | 3.19 | 0.78 | 0.73 | 1.01 | 0.73 | 1.33 | 1.43 | 1.43 | 1.51 |
| G-Gray | 4.40 | 3.04 | 3.97 | 3.80 | 1.95 | 1.97 | 1.00 | 0.72 | 2.21 | 2.80 | 2.71 | 2.47 |
| B-H | 0.41 | 0.77 | 2.25 | 2.04 | 0.66 | 0.56 | 0.75 | 0.61 | 0.41 | 0.45 | 0.90 | 0.89 |
| B-S | 17.20 | 20.73 | 17.28 | 19.03 | 0.77 | 0.93 | 5.34 | 5.54 | 0.81 | 0.68 | 8.28 | 9.38 |
| B-V | 5.61 | 7.37 | 2.07 | 2.92 | 0.83 | 0.72 | 5.78 | 7.34 | 0.58 | 1.62 | 2.97 | 3.99 |
| B-Y | 0.52 | 0.35 | 6.95 | 8.34 | 0.79 | 0.61 | 5.82 | 7.16 | 0.68 | 0.57 | 2.95 | 3.41 |
| B-Cb | 0.43 | 0.32 | 13.58 | 10.51 | 0.75 | 0.59 | 12.27 | 9.26 | 0.71 | 0.53 | 5.55 | 4.24 |
| B-Cr | 1.16 | 1.73 | 5.17 | 4.96 | 0.69 | 0.61 | 0.95 | 0.78 | 3.59 | 3.61 | 2.31 | 2.34 |
| B-Gray | 0.36 | 0.32 | 8.60 | 8.13 | 0.65 | 0.65 | 7.44 | 6.95 | 0.71 | 0.54 | 3.55 | 3.32 |
| H-S | 0.42 | 0.74 | 2.30 | 2.10 | 0.63 | 0.60 | 0.72 | 0.66 | 0.61 | 0.53 | 0.94 | 0.93 |
| H-V | 0.42 | 0.69 | 2.18 | 2.10 | 0.77 | 0.72 | 0.56 | 0.57 | 0.41 | 0.46 | 0.87 | 0.91 |
| H-Y | 0.42 | 0.78 | 2.20 | 2.08 | 0.68 | 0.67 | 0.71 | 0.61 | 0.45 | 0.47 | 0.89 | 0.92 |
| H-Cb | 0.39 | 0.29 | 14.85 | 14.95 | 13.78 | 14.94 | 0.48 | 0.65 | 0.48 | 0.44 | 6.00 | 6.25 |
| H-Cr | 0.36 | 0.51 | 2.86 | 2.51 | 1.74 | 1.79 | 0.57 | 0.53 | 0.44 | 0.50 | 1.19 | 1.17 |
| H-Gray | 0.43 | 0.73 | 2.24 | 2.10 | 0.67 | 0.68 | 0.72 | 0.61 | 0.48 | 0.47 | 0.91 | 0.92 |
| S-V | 5.09 | 7.44 | 4.24 | 5.17 | 1.43 | 1.12 | 4.75 | 5.43 | 1.30 | 1.34 | 3.36 | 4.10 |
| S-Y | 3.84 | 3.94 | 4.55 | 5.89 | 1.29 | 1.04 | 5.02 | 6.82 | 1.76 | 1.69 | 3.29 | 3.88 |
| S-Cb | 1.25 | 0.37 | 12.36 | 22.54 | 0.70 | 0.69 | 11.69 | 21.26 | 0.65 | 0.57 | 5.33 | 9.09 |
| S-Cr | 0.99 | 1.43 | 2.86 | 2.78 | 0.67 | 0.69 | 0.98 | 0.83 | 1.20 | 1.05 | 1.34 | 1.36 |
| S-Gray | 3.77 | 3.38 | 4.97 | 5.96 | 0.83 | 0.79 | 4.98 | 5.73 | 1.72 | 1.67 | 3.25 | 3.51 |
| V-Y | 2.58 | 2.51 | 3.30 | 3.03 | 0.91 | 1.17 | 2.57 | 2.17 | 1.76 | 1.56 | 2.22 | 2.09 |
| V-Cb | 0.55 | 0.51 | 5.68 | 5.92 | 0.85 | 0.86 | 4.75 | 5.11 | 0.71 | 0.58 | 2.51 | 2.60 |
| V-Cr | 1.11 | 1.36 | 7.85 | 13.43 | 0.83 | 0.80 | 0.94 | 0.73 | 6.18 | 11.60 | 3.38 | 5.58 |
| V-Gray | 3.64 | 3.16 | 3.91 | 3.03 | 1.53 | 1.76 | 2.57 | 2.13 | 1.64 | 1.52 | 2.66 | 2.32 |
| Y-Cb | 0.27 | 0.29 | 14.10 | 13.19 | 0.69 | 0.64 | 12.78 | 11.96 | 0.73 | 0.55 | 5.71 | 5.33 |
| Y-Cr | 1.07 | 1.42 | 3.31 | 3.68 | 0.73 | 0.74 | 0.91 | 0.75 | 1.53 | 2.01 | 1.51 | 1.72 |
| Y-Gray | 2.28 | 1.78 | 12.73 | 12.63 | 1.93 | 1.33 | 10.88 | 10.88 | 2.88 | 2.88 | 6.14 | 5.90 |
| Cb-Cr | 0.39 | 0.49 | 2.54 | 2.33 | 1.34 | 1.46 | 0.53 | 0.49 | 0.49 | 0.44 | 1.06 | 1.04 |
| Cb-Gray | 0.31 | 0.27 | 13.12 | 11.14 | 0.66 | 0.64 | 11.82 | 9.91 | 0.71 | 0.54 | 5.32 | 4.50 |

| | | | | | | | | | | | | |
|---------|------|------|------|------|------|------|------|------|------|------|------|------|
| Cr-Gray | 1.08 | 1.64 | 3.32 | 3.78 | 0.76 | 0.80 | 0.96 | 0.89 | 1.53 | 2.18 | 1.53 | 1.86 |
|---------|------|------|------|------|------|------|------|------|------|------|------|------|

Figure 5-b also shows that there is an empty area in the LUT ($0.7 < H < 0.95$) that can be used, if required, to define a new color-based object (this area corresponds to the violet or purple color). Additionally, the results of figure 5-a also suggest that the basic classification information of the LUT could be expressed directly as different interval rules (as proposed in [4]) but it must be noticed that this alternative color classification will have always lower time-classification performances than the use of a LUT.

5 Conclusion

This paper presents the evaluation of the image segmentation capabilities of a compact mobile robot based on Google Android Smartphone. This proposal of a mobile robot takes advantage of the quality of the images provided by such devices. The time required to apply an image processing algorithm to the image acquired by the Smartphone has been analyzed and the effect of using a buffer in the preview callback function has been evaluated. Results show that the original YUV image can be classified in 30 ms whereas the converted RGB and HSV images in 40 and 67 ms respectively.

The best classification LUT was obtained from the two dimensional histograms of the H and V color spaces with an average error in the classification of the pixels of only 0.87%. The objection of using the HSV color space is that the conversion of the original images in the YUV color space to the HSV color space doubles the time required to classify an image. The time results obtained must depend largely on the Smartphone used but the time ratio is expected to be similar.

Future work will be focused in image calibration, algorithm optimization, and in additional image analysis procedures designed to estimate the relative distance to the different objects located in the playfield. Then the next efforts will be focused in the definition of the basic operational agent required to convert a Google Android Smartphone into a gaming mobile robot.

References

1. Android Developers, <http://developer.android.com>
2. The Eclipse Foundation open source community website, <http://www.eclipse.org>
3. Robocup, <http://www.robocup.org>
4. Bruce, J., Balch, T., Veloso, M.: Fast and inexpensive color image segmentation for interactive robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), vol.3, pp. 2061-2066. (2000).

5. Neves, A.J.R., Pinho, A.J., Martins, D.A., Cunha, B.: An efficient omnidirectional vision system for soccer robots: From calibration to object detection. In: *Mechatronics*, Vol. 21, Issue 2, pp. 399-410. Elsevier, Amsterdam (2011).
6. Jamzad, M., Sadjad, B.S., Mirrokni, V.S., Kazemi, M., Chitsaz, H., Heydarnoori, A., Hajiaghaj, M.T., Chiniforooshan, E.: A Fast Vision System for Middle Size Robots. In: *5th International Workshop on RoboCup 2001*, pp. 159-203. Springer, Heidelberg (2002).