Ilja Zegars

Colour Measurement

Using Mobile Phone Camera

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

5 May 2013

Helsinki
Metropolia
University of Applied Sciences

Author(s) Title Number of Pages Date	Ilja Zegars Colour Management Using Mobile Phone Camera 34 pages + 0 appendices 21 May 2013
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Olli Alm, Lecturer

The purpose of this thesis was to analyse the colour measurement on different Android based smart phones. The results are compared and presented in simple explainable form. Colour measurement strategy is described in this thesis, so the application has been developed entirely for this purpose.

An Android application was created for the thesis, which uses Android camera to pick colour from the camera view. The application is written on Android Java. Multiple colour recognition algorithms are implemented in the code and explained in the thesis.

The results show that colour recognition using smartphone camera is not reliable and can give results that are far from the actual result. The result depends on multiple factors, such as the factory settings of the camera chip, lighting of the object and colour calibration.

In future it is possible to get more reliable results, if the camera calibration is implemented in the application, requiring user to calibrate the camera for different type of lighting.

Keywords	Android, camera, colour recognition, RGB

Contents

1 Introduction

The world of digital media is constantly increasing its influence on more traditional, paper media. Everything is not digital, however. Most objects existing in the real world should be digitised in order to be manipulated digitally. The same manipulations need to be performed in order to convert the properties of the object such as density, flexibility, shape and also the colour. Colour, shape and look of the object are the first specifications of object that a person sees. In digital media the user is not always able to interact with the environment, but he is able to see it. The task of colour reproduction in digital form might be difficult and complicated due to multiple reasons.

Professional colour picking equipment exists on the market but has a large price tag compared to smartphones and is not suitable for all types of objects. It is mostly used for picking colour from the printed media, such as magazines, newspapers and posters. In a scenario where the designer is required to pick a colour of non-paper printed object like a wall, table or the car, he is stuck either with equipment that is even more expensive, or with task of guessing the colour.

This thesis describes the development of the software for simplifying the process of converting real world colour (the colour that is perceived by human eye) to its digital form. The software uses the camera of an Android based mobile phone to pick up the colour and return the value in the RGB form that can be further converted into any other digital form suitable for the user. The software is able to store the results in memory for future reference and sharing them online.

In order to get as close as possible to the real world scenario, when colours need to be picked up fast without the possibility to evaluate results, the thesis aims at comparing multiple Android based devices that have a camera with the professional equipment available in university printing lab. Real world scenario is a situation when the environment has imperfect lighting, the device has no reference colour or the reference colour sheet is not available at the moment and results are needed to be received fast.

Topic was chosen in order to simplify work for the web designers and identify devices that are able to give results close to the professional equipment in the field. The devices will be chosen from the phones currently available on the market, using cameras manufactured by different companies and having different camera resolutions. The results will be taken in pure RGB colour form allowing value comparison and analysis of colours and devices.

2 Colour measurement

2.1 Colour management

It would be perfect if we had ideal colours in real life, which can be easily imagined when a person says “red” and it will be always {255; 0; 0} in RGB form. To be precise, RGB is ideal, but it is more like that it does not match exactly with the human perception and neither for real-world colors. Unfortunately it is not possible to get these colours in real life scenario due many factors giving its influence to the colour. As it can be seen from Figure 1, the ideal colour cones have a significant difference to cones that are visible to human eye. The areas of sensitivity are overlapping each other making it more difficult to determine the colour by person himself/herself.

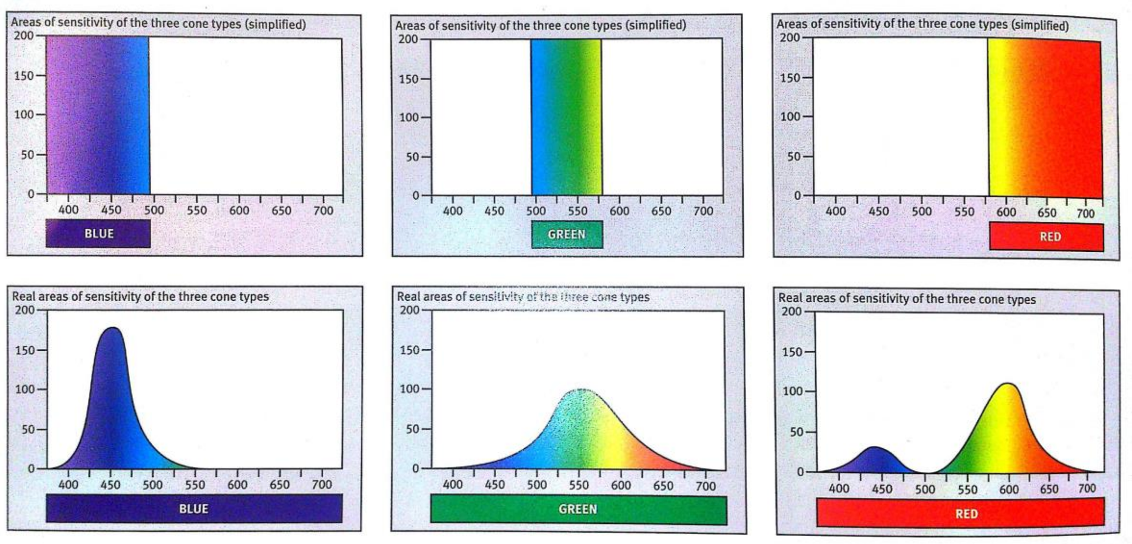

Figure 1. Comparison of sensitivity in the simplified cone model from the eye’s actual perception. [1]

Only one type of cone can be stimulated by very narrow spectrum. This phenomenon is resulting into that monitor’s primary colours look more saturated for the eye than printed ones. Light source also affects the perception of the colour in the eye.

Science that studies the connection between how a human perceives the colours and the colour stimulus spectrum is called colorimetrics [3]. The most commonly used model of colour persistence is the LCH [4] (Lightness, Chroma, Hue angle) colour space based on mathematical equations.

The LCH colour space can be divided into three attributes: Lightness, Chrome (saturation) and Hue. Different spectra could produce same LCH values, just the same way a human can perceive different spectrum with the same impression of colour. Many modern input and output devices used in the graphics industry such as monitors, printers and scanners use LCH as a basis for the colour description.

In the LCH colour space only the colours that can be presented for printing without great deviation are visible, all the others remain invisible as shown in Figure 2. The results are stored in a form of three dimensional colour space. "L" for lightness is a value changing depending on the position – the higher it is – the lighter it is. The further a colour is from central point, the more saturation it has or the "C" value. Finally, colours on the same level and at the same distance from origin can be different by the third value "H" the hue. This has become a result of thousands of experiments and research during decades. [1, 38]

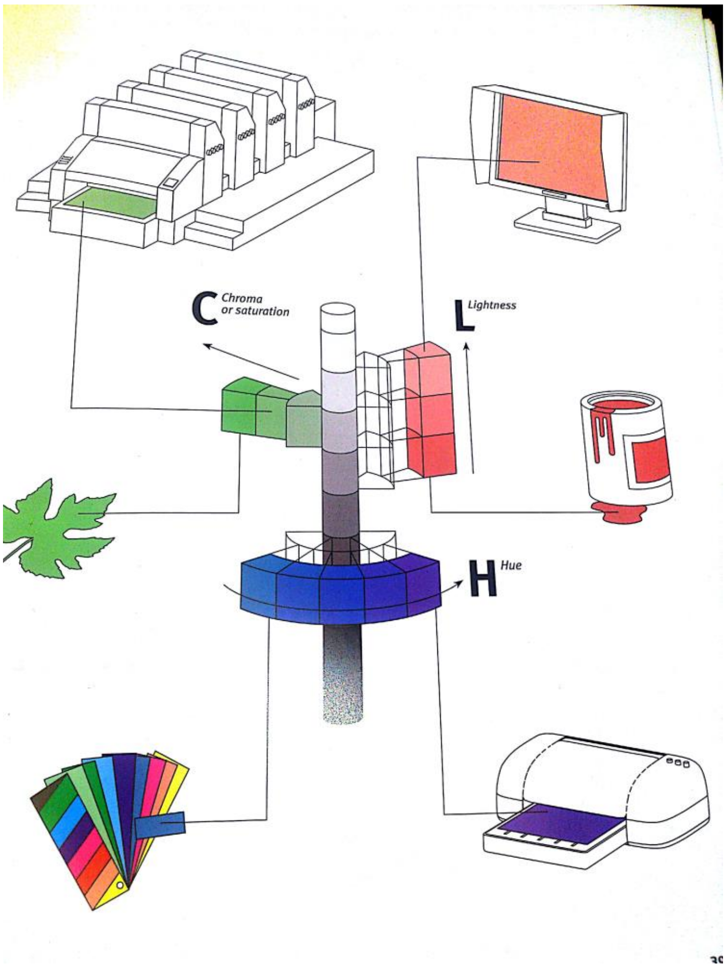

Figure 2. A colour model of the LCH colour space. [1]

LCH colour space has resulted in many variations of custom spaces, one of which, called Lab colour space [1, 54], can be found in most professional design software available on the market today. Unlike the LCH, the Lab does not use degrees for hue definition and distance from the axis for the saturation. It spreads the right-angled system of coordinates across hue and saturation. This allows simplifying the measured values, which makes this type of colour space more popular compared to other LCH spaces.

2.2 Professional measurement equipment: the Spectrophotometer

Spectrophotometers are the devices that are most commonly used for colour management of printed products and surfaces. They give the result in LCH or Lab form which can be further used for a number of tasks. Although the price went down significantly in the past few years [1, 52], it is not low enough for non-professionals to buy for one time use. Spectrophotometers are commonly used by professionals to calibrate proof system, control colour of the printing device and monitor profiling.

2.2.1 Functionality of Spectrophotometers

The spectrophotometer works by the same principle as the human brain uses the eye to get objects’ colour [5]. Light being reflected from the object gets to the sensor. The sensor analyses the spectrum of the light received checking radiant intensities for different wavelengths and translating them into the digital format as shown in Figure 3. Light reflected from the image comes to the sensor, where it is being processed and analysed. After that it will be converted to any preferred colour space form. Software provided by the equipment manufacturer uses digital values to determine LCH or Lab values.

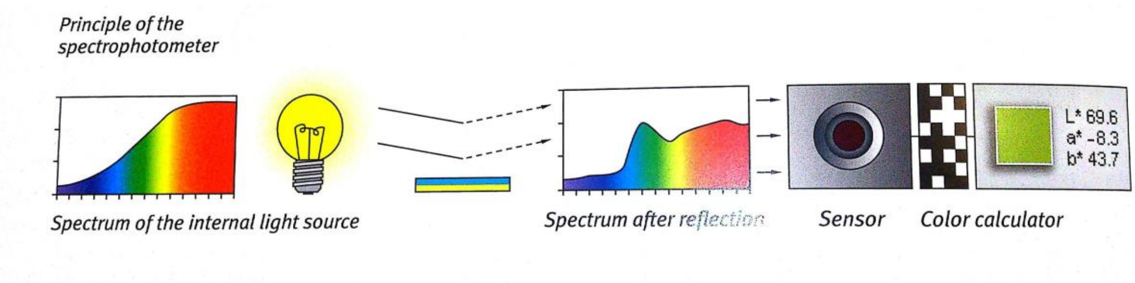

Figure 3. Principle of the spectrophotometer.

The spectrum of the lighting might change the resulting values of the device, since different spectra is reflected back to the sensor. For this case, spectrophotometers manufacturers determine standards for the light source called standard illuminants [6]. Standard or ideal light source for the colour management is “D50”. This should be specified in the spectrophotometer’s software so that the calculated results will be correct.

2.2.2 Types of Spectrophotometers

Spectrophotometers come in different sizes and serve for different purposes. Some of the devices are able to get the result not only from the prints, but from monitors as well. Those types of devices can be in different shapes and sizes, but a typical device used for picking colour from paper is shown in Figure 4. This kind of device will be used in the testing chapter of this thesis.

Figure 4. The EyeOne Pro from X-Rite is the professional spectrophotometer. [1]

The device that is shown in Figure 4 is one of the handheld measuring devices – the devices that are being directed to the source for receiving the colour. Other type of device is the Scanning measuring device – it is larger and is positioned stationary while the print is being scanned through it. Due to its design, the scanning device cannot be used for picking up the colour from monitors or other non-plane objects.

2.3 Using mobile phone camera for colour recognition

Although spectrophotometers are available today, these devices are not suitable for the “in field” work. Usually the equipment is not applicable for field jobs at all due its shape or size. Smartphones used as colour picking devices could solve that problem, but there are some limitations.

The problem with smartphones is in the unavailability of image from the camera before it has been processed. The image that can be processed is already modified and is available in JPEG form. In order to make precise calculations it is necessary to have access to raw camera data, which is not possible, because manufacturers want to keep their calculating algorithms in secret. Fortunately most of the devices save their images with sRGB (standard RGB) colour space which is a standard for displays, printers and the Internet [8]. Unfortunately sRGB color space does not take into account how much the color in the environment corresponds to the color value reproduced by the camera.

It does not give us a precise result, because the colours are being generated by the camera, but assuming that all of the devices use similar colour space, we might make a comparison with some limitations. With the sRGB comparison, it is possible to see that the results vary but not exactly how much the color reproduction varies between cameras.

In s scenario where a designer spots a sports car parked along the street and decides that this blue colour is exactly what he needs for his current project, the designer takes a picture of the car using his phone, goes to the office, downloads the photo and uses professional software for picking up a reproduced colour to use in future. In case where the designer does not have specialised software he might print out the picture taken with phone and uses a spectrophotometer on the photo to get the colour.

Modern mobile phones or so called “smartphones” have calculation power much higher than computers had in 1961 for landing the first man on the moon. [2] With this kind of performance it is just a matter of time when specialized software will be able to support designer tasks. The difficulty for this type of software is that there are multiple phone manufacturers creating different cameras, different sensors and different optics for hundreds of devices. Unlike the spectrophotometer where the user defines the standard illuminant value, in the field the light source could be any type, spectrum and colour. Sometimes there could not be a light source at all. Unfortunately end users could not rely on device built in flash light since the quality varies and some devices do not have flash at all.

2.4 Existing mobile applications for colour measurement

The idea of creating an application suitable for designers and web developers came to the mind of different people and resulted in multiple implementations. While some of the existing solutions are not free, in this thesis we will compare applications that are available without any cost / free of charge. As it happens often on Android application store called Google Play store, many developers of software or mobile games switch to free versions of their product, while implementing Google Ads inside the software itself. Previously, it was more profitable to set the price tag for the application in a range from €0.99 to €5.-, but latest research has shown that having a free application with integrated advertisement earns more money for the developer [7]. The developer and the Google Company both are earning money from users who click on ads intentionally or accidentally. This is one of the reasons why a free application was chosen – it does not affect product quality. Another reason is that the application created for this thesis is free of charge as well.

Three applications were selected for evaluation. Solutions that are on the market will be compared to each other, before the deep explanation of the application created for this thesis is introduced. Applications were picked from the Google Play store. A list of the applications is provided in Table 1.

Table 1. Comparison of existing free Android applications based on user interactions.

Application name	**Reviews**	**User rating (of 5)**
Image Color Picker	139	3.8
Color Picker	154	3.6
Rgb Color Picker	20	3.7

Table 1 also provides some information about the amount of reviews and average rating. The amount of reviews allows us to determine how many people found the application useful enough or did not find it useful to give it a review. User rating value allows us to define user satisfaction with the application in general.

None of the applications provided a real-time result while picking image from the camera. By “real-time” a feature is meant that would provide the user with value at the same time when the device is pointed towards the object, and it would change dynamically while the user moves the device around. Probably because the applications were created a while ago, the computing power of smartphones was not enough to process this kind of analysis, which was the reason for developers not to implement this feature. Nevertheless, all of the applications provided the ability to pick up the colour from the image from the gallery. The user could take a picture using an application and then determine the point on the image where he wants to pick colour from. At the same time the image would be saved to the phone memory. All these images would be stored in the device showing up in users’ personal gallery, together with their own photos of family, friends and etc. It will result in a huge garbage collection of unused and unnecessary photos that would mix up with the user’s personal photos and should be removed manually. It is assumed that the smartphone is first of all a personal device of a customer and secondly a tool for colour picking. It would be a great idea to implement the application in a way it would not interrupt with other smartphone functions and applications. In a scenario where the user has an additional smartphone for work purposes, this drawback might be ignored. Nevertheless, the device would have big collection of random object photos if the user uses the application a lot.

The **Rgb Color Picker** provides limited real time colour tracking, where the user is provided with the name of the colour in text form as seen from Figure 5. When user is informed that the colour is “dark olive green” he is supposed to tap on the small window on the bottom of the screen without shaking the device. Shaking the device would make the picture move resulting in a blurry image and the value might change or would be inaccurate, especially when the required colour source object is small or is in a significant distance from user. After taking the picture the application will switch to another window, where the values of RGB and CMYK are provided. In case when the user is not satisfied with the result he needs to tap the back button, which makes a lot of interactions from the user side to pick the right colour.

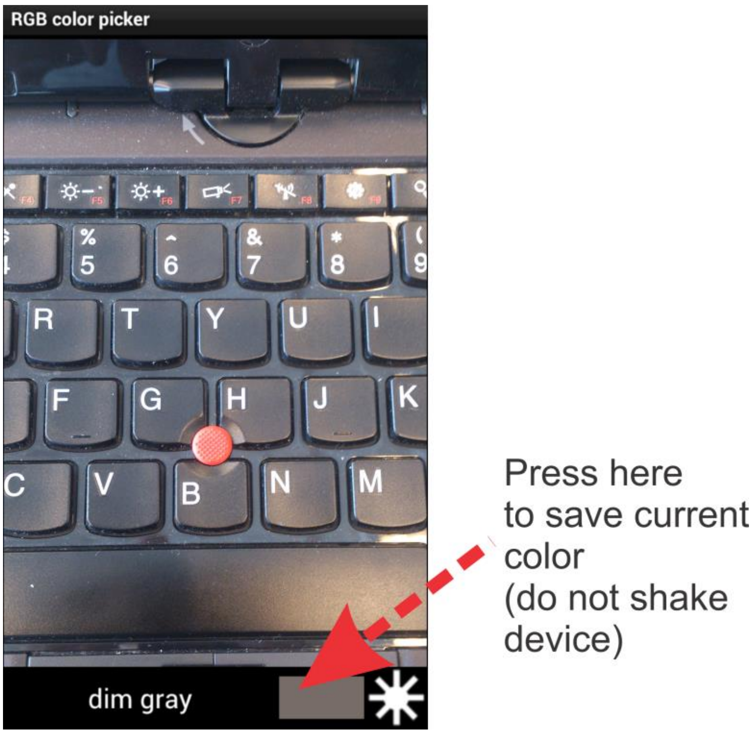

Figure 5. The EyeOne Pro from X-Rite is the professional spectrophotometer.

The **Color Picker** application has limited functionality due to inability to save, share or interact in any way with the colour selected. The user is supposed to either write colour values down on the paper, or memorise it. The user can tap on the specific point of the image and hope that the intended colour pixel will be selected, since the result is shown at the moment when the user removes the finger from the screen. The user cannot touch the screen to see dynamically colour on specific region / point of the image. This feature would be useful since sometimes specific colour area on the image could be small size and it would require pin-pointing to get the result. This feature is being called “Touch & move” in table 2. Unfortunately, “Color Picker” can provide only one result at one touch.

Table 2. Comparison of features of existing free Android applications.

	Rgb Color Picker	Color Picker	Image Color Picker
“Real time” result	Just text value	No	No
Colour from image	No	Yes	Yes
Picture storage	Not taking it	Yes	Yes
“Touch & move”	No	No	Yes
RGB values	Yes	Yes	Yes
Saving result	No	No	Yes
Zoom	No	No	No
Settings	Yes	No	No

The last application chosen for testing is called **Image Color Picker**. Table 2 shows that the Image Color Picker is more advanced compared to other ones, while still lacking some of the essential functions. It allows saving the colour, previewing it saved in visual and HEX form. It is also possible to share the colour in a text form. It has a significant advantage over previous applications, since this application supports the dynamical change of the result while the user swipes the finger over the screen. Most people would agree that the finger is not the most accurate tool for pointing pixels on the screen. Probably only children’s fingers could be used for this purpose, but let us face the truth, not everyone is carrying children or their fingers with them all the time. Some of the devices have a stylus built in for easier user interactions, but the amount of these devices is not large. Recommendations would be to use a pointer, which would be visible above the user finger, and would simplify the process of pointing.

This application is missing a real time colour picking feature as well as the “Color Picker”. Other than that, and the fact that it saves the taken camera pictures in the phone memory, it is a solid application.

3 Camera App

3.1 Interface

The application created for this thesis is called "**Camera App**". The name was chosen in order to specify that real time camera image processing is being used unlike the applications described above. This method takes larger calculation powers to analyse an image but it saves time for the user. While phone cameras are usually working on the frequency of 25-60 frames per second, "real-time" analyses need to check every separate image colour and updates the output values dynamically. It saves user from a collection of pictures in the device's library which is more convenient for the user. At the same time it allows to save the result, to be used later. It stores the colour value with graphical representation and the values are both in HEX and RGB form as shown in Figure 6, which are widely used among web designers. It has a graphical and text representation of the colour, which makes it more visual.

Figure 6. Example of the saved form of colour.

There are three graphical presentations. User interface of the main window is shown in Figure 7. As the application has two different algorithms of getting colour which will be introduced later, the current screenshot shows an estimator taking three average values from the centre of the image limited by the crosshair size. On the left side the crosshair is pointing at grey keyboard part, on the right, focused on the red mouse pointer, and correspondingly the changed colour space is shown below.

Figure 7. Screenshots of the “Camera App” main window.

A major benefit of the camera app is the ability to zoom. Application allows to zoom in to a specific area, making picking colour of small elements of the surrounding easier. There might be hardware limitation for the zooming capabilities of the Camera app on old devices with low quality cameras.

Camera App has multiple features or functions that could be stated in the list:

- Real time results;
- Result in RGB and visual forms;
- Ability to Save;
- Ability to zoom;
- Flexible and adjustable UI;
- Multiple colour picking algorithms;

In order to save the result there are multiple ways to do that. The most simple way of saving result to the memory is to tap somewhere on the screen. This gives flexibility for the user, since different people have different fingers / hands / other that might complicate the usage of application or device. Different screen sizes might create difficulties for user to reach the fixed “Save” button, which will result in user inability to use the device. Another way of saving is to use the “Settings” button of the Android device. It would open the pop-up menu of “Settings”, “Save”, and “Open Saved”. Obviously, to save current values for future use, user is required to tap the “Save” button. The third way of saving the result is to touch and hold the screen, waiting for small pop-up with “Save” option.

Three applications evaluated in Chapter 2 had none or limited settings to adjust for the UI or for the colour measurement algorithm. The “RGB Color Picker” is the only application out of three that has the “Settings” tab, where unfortunately, only camera resolution could be changed and it is not stable, making the application crash sometimes. Adjustable User Interface is highly important for the user since different people may have different requirements. User interface may be adjusted for picking just one colour or removing the pointer in the middle of the view. Settings window allows the user to modify the view as shown in Figure 8. The user can change the visibility of the crosshair and the value boxes that are showing the values on the colour bars on the main screen. It makes the application more personalised and can be adjusted for a specific situation.

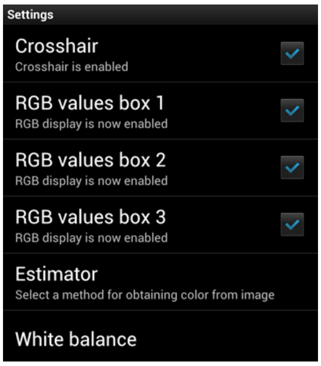

Figure 8. Settings menu of “Camera App”.

Estimator menu allows the user to choose the algorithm for estimating the colour in the view. There are two options, the “Average from central area” and the “Single pixel” will be explained in details in the next chapter of this thesis. The last option of the menu is the “White balance” that is automatic by default but can be changed manually depending on the lighting in the scene, where the application is being used. Apart from the automatic there are options of “incandescent”, “fluorescent”, “daylight” and “cloudy-daylight” which may modify the values depending on the situation.

The third option of the Android Settings button being pressed on the main screen is “Open saved”. It allows the user to see the saved results in the order, how the calculations were made. The user can delete any of the results at any time by selecting the result, pressing Android Settings button and tapping the “Delete” option. The user can return to main menu anytime by pressing the Back Android button.

3.2 Colour recognition algorithm

As it was stated previously, there are two algorithms for colour estimation implemented in this program. None of the existing solutions described in Chapter 2.4 had more than one estimation algorithm. Unfortunately we could not tell which type of analysis was used in these solutions, since the source code of the applications is not available for public. In order to give more options for the user and be more flexible and adoptable for different scenarios, the “Camera App” implements two different algorithms.

The estimation may work as the EyeDropper tool in Adobe Photoshop, getting the colour of one exact pixel that the pointer is pointing at. This gives a very precise value of the colour in case of the static image and the Photoshop. Sometimes the view from which the user wants to pick a colour may have a dynamically changing light source, such as sun or flash of the camera, which may change the exact pixel colour value making it difficult to choose acceptable result. Nevertheless, it has been mentioned before that giving user flexibility in options of the application is giving advantage comparing to other applications.
Estimation can be calculated in multiple ways. The easiest one is the single pixel estimator. It has its own drawbacks and benefits:

Pros:

- Easy implementation – for the Android based device it would be very easy to implement. This kind of function just taking the colour sample from the very centre of the image would be very simple to create.
- Low CPU consumption – logically assuming, if there are not many lines of code, it is easier for smartphone or tablet to process. That would result in fewer calculations, lower power consumption, which is good for device battery life time, and less heating.

- Faster operation – smaller programs are easier to operate on the device. Even old CPUs could manage that task without any troubles, making this algorithm universal for many devices.

Cons:

- Colour result – taking single pixel from big image, is not very effective. Image colour consists of many different shades of the same colour allocated very close to each other. Whenever the picture is taken, if user zooms in excessively, there will be some distortion on the image. . The object never has solid colour due to its imperfect surface, light sources etc. The result would be constantly changing due to the shaking device.
- One pixel result – the human eye receives the colour as a surface, as the area of pixels. Showing just one pixel to the human would not help him/her understand the colour – it is required to have at least some area that is made out of many independent pixels of different colours.

The single pixel estimator is the first algorithm that has been implemented in the "Camera app". Its code is simple and could be divided into several steps:

1. Allocate camera – usually the front camera is primary and face camera is secondary, but it is better to make sure.
2. Get the size of camera sensor width and length – these variables should be found to define the centre of the screen, from where the pixel value would be taken.
3. Get the colour result using functions `Color.red(data[y])`, `Color.green(data[y])`, `Color.blue(data[y])` and have the results.
4. Return the colour in the RGB form using `Color.rgb(r, g, b)`.

These four lines represent a very simple way of getting a colour that could be implemented in less than five lines of code. As it can be seen, Android has most of the functions built in, for picking the colour from the image. The most difficult part of that is making it able to receive the result from the camera view in real-time without taking the picture.

The second algorithm is called “Average from central area” and as the name suggests, it calculates the colours from the area in the centre of the camera view. In order to make pointing the device easier and more user friendly, the crosshair in the middle of the camera view has been added. It could be disabled from the Settings menu, however. The area of estimation is limited by 16x16 pixel square in the middle. It is defined by eight pixel step in every direction from the centre.

```
public int getAverageColor(int[] data, int width, int height) {
          int mx = width / 2;
          int my = height / 2;
          int cnt = 0:
          int r = 0, g = 0, b = 0;
          try{
          for (int x = mx - SIZE; x < mx + SIZE; x++) {
          for (int y = my - SIZE; y < my + SIZE; y++) {
          r += Color.red(data[y * width + x]);
          g += Color.green(data[y * width + x]);
          b += Color.blue(data[y * width + x]);
          cnt++;
          }}
          } catch (Exception e) {
          return Color.MAGENTA;
          }
          return Color.rgb(r / cnt, g / cnt, b / cnt);
}
```

Listing 1. The implementation of the “Average from central area” algorithm.

The code for this part of the application can be seen from Listing 1. It has mainly the same steps in implementations, as the single pixel algorithm, with different steps three and four:

3. Calculate every “r”, “g” and “b” value for every single pixel in the 16 by 16 pixel area in the centre.

3.1 Sum all the values for each of the element of “r” “g” “b”.

3.2 Divide by the amount of pixels that have been used (in our case 256).

4. Return the colour in the RGB form.

This algorithm implementation has just a couple of additional lines of code, but it increases calculations for the CPU from 1 pixel to 256 pixels. One should not forget that it means 256 calculations at the rate of around 30 times per second. This fact affects the CPU performance, but still calculations are very simple and could be performed easily even on three years old device with one core CPU.

Algorithms share the same basic functions, but behave in a different way, returning a comparable result. Single pixel algorithm returns the value that can be used only if the sampling object is allocated in a highly lightened area, with multiple light sources, no shadows and with the solid object colour. On the other hand “Average” algorithm is the one that could be used for the “in field” measurement – where the light is not perfect, colour may be gradient and having some imperfections. This is why it has been decided to make “Average” algorithm a default one for every application launch while “Single pixel” is the secondary option.

4 Evaluation of Camera App

It was decided to test the application in the field environment and compare the results. Multiple devices were used for the application: smartphones, 5-inch note, 6-inch screen tablet and 10.1-inch screen tablet. Different device manufacturers were chosen for more reliable results.

4.1 Testing environment and equipment

For the testing purposes a room was chosen with indoor light and no windows. Using the natural light source such as sun (or moon) would not give valid results. The clouds in the sky or other weather changes would change the intensiveness of the light and its power. Since we are using smartphone camera without flash, it might give significant changes in the colour saturation and in the final result itself. Although the indoor light was chosen, it was decided to use a room without any windows, since multiple light sources might affect the result. Additional light sources refer to the natural light sources.

Every device has been tested in the same conditions. Device camera was pointed directly toward the "**testing object**". Distance between the camera and the "testing object" should also be taken into consideration, since it is a variable that affects the final result. If the camera of the device is too close to the "testing object" it might create shades from the device that would drop to the "testing object" changing its colour for the camera sensor. In other words, it would change the results making analysis more difficult. Moving the camera a long way from the "testing object" would not do any good either. Long distance would result in the requirement of more precise camera pinpointing to the object, since the objects would look smaller on the camera view. Based on these observations, the optimal distance of 20 cm from the "testing object" was chosen.

The "testing object" was attached to the wall, as seen in Figure 10, for easier pointing of all devices towards it. In order to give different scenarios for the camera, and recognition algorithm, three positions were used for every device. Positions which were used in the practice for colour picking are marked with numbers 1-3 in Figure 10.

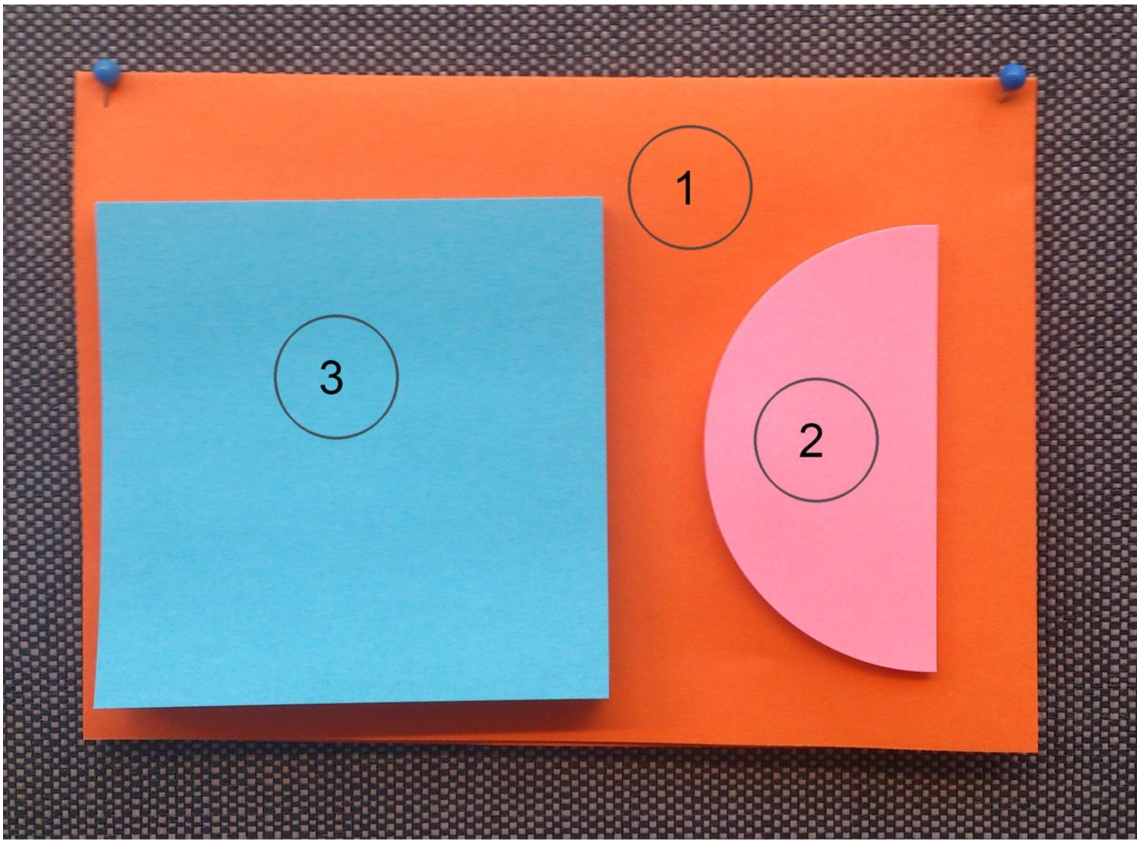

Figure 10. “Testing object”.

In order to have some ideal reference, at first it was decided to use spectrophotometer and its results. It is quite possible to get values from the spectrophotometer that would be very close to visible colour for the human eye. But there are some difficulties with this kind of colour picking. The spectrophotometer is being placed directly on the object to pick a colour, so in order to get rid of shadow and get a light source, the built-in light source is being used. This means that results would be different because light sources are different. In this case, the ideal reference was the “testing object” picture and the colours were picked using the Adobe Photoshop CS6 software. It is necessary to note, that in these tests, the average of the area algorithm has been used on all of the devices and only the first result of three has been shown. This algorithm was chosen, since single pixel algorithm may create unnecessary interference in the results.

The devices for testing were picked from multiple vendors. The smartphones were Motorola Razr XT910, Samsung Galaxy S 5.0, and Sony Xperia S. The tablets were from Samsung and HTC: Samsung Galaxy Tab 2 and HTC Flyer P512. Different devices had different Android versions, but since the application was created for API Level 8 it did work on all of the devices. API Level 8 refers to Androids version 2.2.x Froyo and up. Motorola, Sony and Galaxy Tab 2 had Ice Cream Sandwich (Android 4.0.x) while other devices were older having the latest available firmware for them, either Éclair (Android 2.1.x) or Froyo (2.2.x). Either way, the software worked on all of the devices, having slight problems with camera view window, which did not affect the functionality of the program, just made it look messier. This happened due to the different camera resolution and camera sensor. Different devices had different screen size, but it did not create any troubles since the program did not use precise dimensions in pixels. Instead, the “dp” values were used. The “dp” is an abstract unit based on the physical density of the screen. If the button is 10 dp wide, and is shown on 160 dpi (dots-per-inch) screen, it would be 10 pixels. But if it is a 320 dpi screen, the button size would be increased as well to 20 pixels. This technique was used to avoid unnecessary adjustment of UI (User Interface) for each specific device.

In order to give a fair test to the application, devices having different camera resolutions were used. Comparison of the device cameras is shown in Table 3. There are three main manufacturers of camera sensors that are developing sensors for all other smartphone developers.

Table 3. Device camera comparison.

	Camera sensors	Camera chip manufacturer
Sony Xperia S	12 mpx	Sony
Motorola Razr XT 910	8 mpx	Omnivision
Samsung Galaxy Tab 2 10.1	8 mpx	Samsung
Samsung Galaxy S 5.0	3.15 mpx	Samsung
HTC Flyer P512	5 mpx	Sony and Omnivision

The worst camera sensor is installed on the Samsung Galaxy S 5.0, so it is expected that this device would have a result with biggest difference to the ideal reference. The Sony smartphone has the largest camera sensor resolution, making it potentially the best result of all of the devices. Other devices have competitive sensors as seen in Table 3.

4.2 Testing results

As one may know, “the only difference between science and screwing around, is that in science everything is written down”[9;1]. That is the basic idea behind science. The results were taken and organised in the way that is shown in Table 4.

Table 4. Testing results for multiple devices.

		Sony Xperia S	Motorolla Razr XT 910	Samsung Galaxy Tab 2 10.1	Samsung Galaxy S 5.0	HTC Flyer P512	Photoshop CS6
Orange	R	249	217	238	200	222	254
	G	90	106	53	133	70	109
	B	11	73	8	17	8	56
Pink	R	248	228	236	210	216	255
	G	115	77	153	115	80	152
	B	143	87	156	125	112	173
Blue	R	59	125	128	115	40	88
	G	160	164	182	148	144	191
	B	160	163	193	150	180	222

Every result was written down in a form of RGB which is easier to analyse. Each colour value for Red, Green, or Blue is stored in its own row. As seen from Table 4, most values for Red in Orange and Pink columns do not differ very much. It is a good result, since Red is the main component for these colours. Obviously, for the blue colour the Blue component is the main and we can see that the results do not have a very big range. For the ideal reference, naturally values are slightly higher, since the “testing object” has more light than Android devices do.

The results have been organised in a bar chart, which gives visual understanding, on how big the difference is in the results. Figure 11 has different colour bars for different devices. The last colour bar, coloured in orange represents the values taken using Adobe Photoshop CS6 software, the ideal results. As seen from the graph, the ideal result is usually slightly higher, but as it was mentioned before, it happens due to the additional light source for the camera – camera flash.

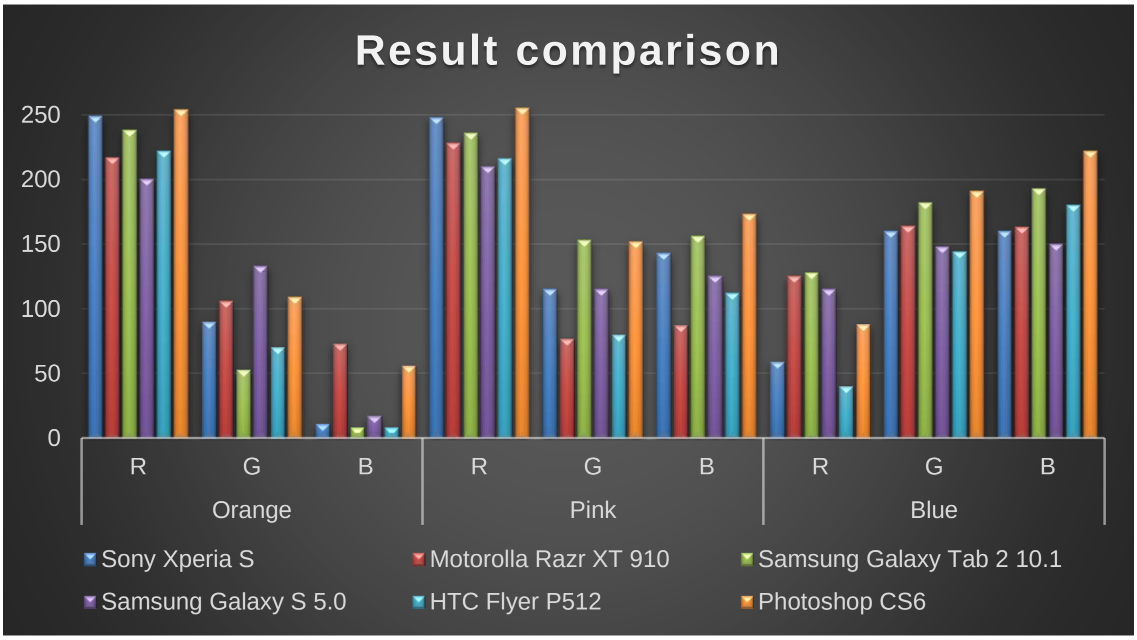

Figure 11. Testing results comparison.

Analysing the Orange colour source, it is visible that the closest result to the ideal reference, which is Adobe Photoshop column, is the result of Sony Xperia S smartphone. It is true for the R (Red) column. The second best result is the Samsung Galaxy Tab 10.1 following with HTC Flyer and Motorola Razr. All of the results have a small range from 200 to 249, which is a very good result taking into consideration different manufacturers, cameras and optics.

G (Green) value for the orange has a wider range of values which is unfortunate, but not very critical, since the main component for creating or reproducing orange colour is R value. Nevertheless, devices with better camera sensors have closest to ideal results – Xperia, Razr and Galaxy Tab 2. Flyer had a result which is slightly higher than expected and it can be explained by the low result for the R value, which should be compensated in order to get similar or close to similar colour. G values vary from 53 to 133 with the Adobe Photoshop value of 109.

The last value is the B (Blue) value, which is close to zero, since there is no or close to none B value for the orange colour. Most of the devices have a very low result of up to 17, while Razr has an unexpected value of 73. Overall image of all the results, when converted to the colour form can be seen in Figure 12.

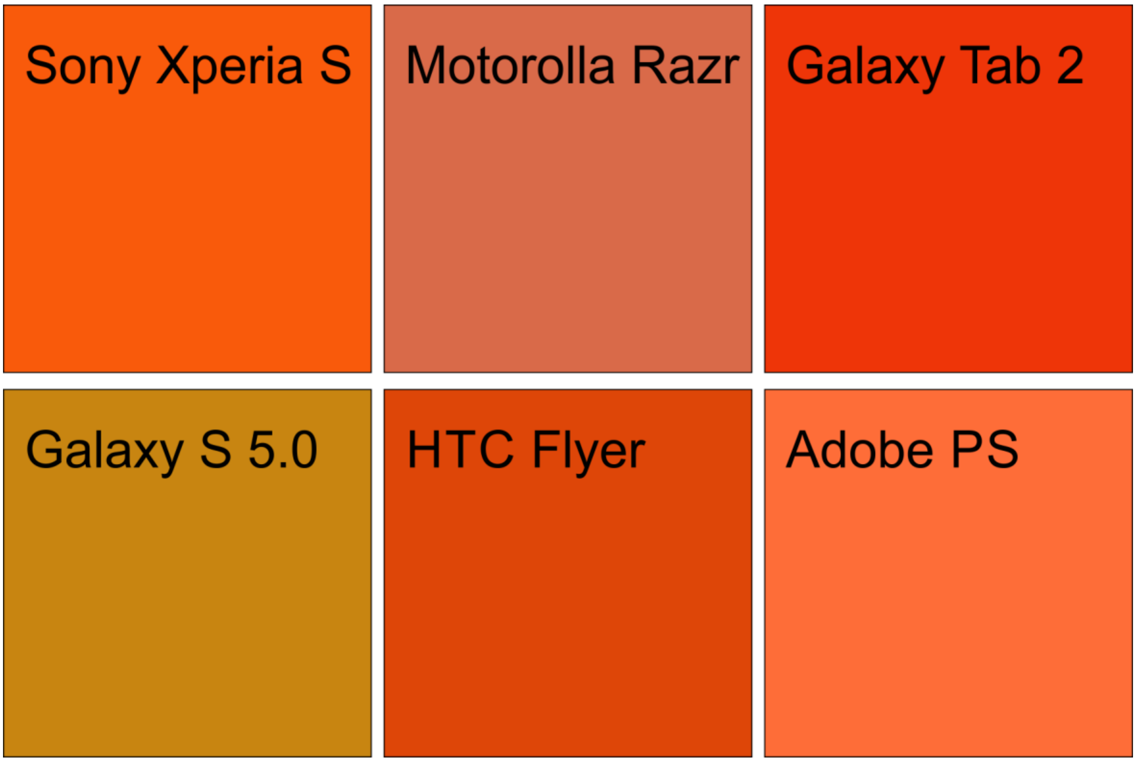

Figure 12. The results of camera recognition for the Orange colour.

As seen in Figure 12 without any special equipment, Sony device has the closest result to the ideal reference. The worst result is given by the Galaxy S device, but it was expected, since the device has the worst camera sensor, compared to the other devices. Other results are close, but the result of Galaxy Tab 2 and HTC Flyer looks more red than orange.

Analysing Pink colour results it is easy to spot that there is not much difference in R colour component compared to the Orange colour. The range in values varies just from 210 to 255. The G component has the greatest value for the Galaxy Tab 2 tablet, having the closest value to the ideal reference. The results of other devices did not change much compared to the Orange colour tab. The biggest changes happen to the B component: 173 is the ideal reference value and all of the devices have results in a range from 112 to 153, getting the best result of Galaxy Tab 2. At the same time Razr has a result, which is much lower than expected – the effect on the colour can be seen in Figure 13.

Figure 13. The results of camera recognition for the Pink colour.

Again, Figure 13 shows that the closest result is the Sony Xperia S device – the device with the best camera sensor compared to other devices. Galaxy Tab 2 also gives a very good result losing a bit of colour saturation. Other devices are providing the user with a result that is unacceptable, and could not be a good reference in case of emergency need of colour pickup. Motorola Razr and HTC Flyer have values that create colours that are too dark and faded.

The last test involves Blue or Sky Blue colour analysis. This colour does not have R as a main component, and the results would be different because of that. First of all, ideal reference value for R is slightly lower than average for the devices, it is just 88. While most of the devices have results from 115 to 128, the Sony Xperia S has an unexpectedly bad result of 59. This is unfortunate since it was expected that camera with best resolution would have the closest to ideal result, and it was true for first two tests.

The G component has values which are very close to each other, for all of the devices. Starting with 144 for HTC Flyer and finishing with 182 for Galaxy Tab 2. As it is in most cases, the ideal reference result is slightly higher – 191. There are no abnormal results, and everything is predictable.

The B component obviously is the main component for creating blue colour, and it is critical to have correct value in order to get close to perfect result. The Galaxy Tab 2 gives closest to the ideal result of 193, again. The Sony Xperia S does not do any good and shows just 160. The results of colour simulation are shown in Figure 14.

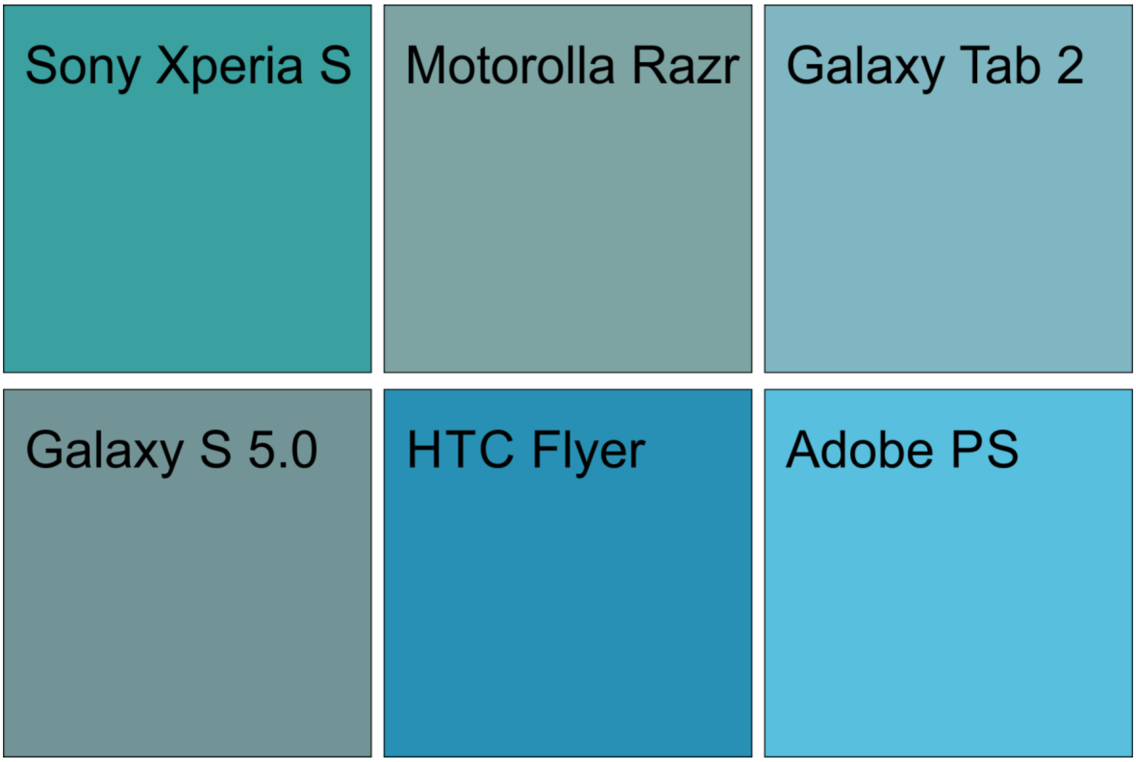

Figure 14. The results of camera recognition for the Blue colour.

Figure 14 demonstrates the results of the third test, which is different from the previous tests in the result. None of the results show the colour that is the ideal reference, but the closest results are given by HTC Flyer and Galaxy Tab 2. HTC Flyer did not show good results in any of the previous tests, but did surprisingly well for Blue colour. The Sony Xperia S was doing well in Orange and Pink tests, but for Blue – the result is closer to green colour than Sky Blue.

As it is seen from Figures 12, 13, 14 – the results often have values that are representing colours which are not precisely the ones that have been visible in real life. There are many factors that affect the final result, and should be considered before the measurement. These are the light sources, described previously and other light factors. But the main factor of having different results is the calibration.

In the theory part, it was described that spectrophotometers are being calibrated before the measurement, using special colour reference tiles. The tiles made of ceramic which makes them not to lose colour over time and they can be used as the reference for longer time. For a human the red colour has a very wide range, going from almost brown to almost pink. Device recognises it as different, it has defined colours, and the red one is exactly one colour. Calibration makes the device “understand” the main colours such as R, G, and B components. Unfortunately, smartphones do not have a built-in calibration for camera. In the end, those cameras are supposed to take pictures and record videos, not analyse the colour origin.

4.3 Possible improvements

Calibration of the mobile phone cameras is possible, it is one of the real possibilities of improving the results. Manual calibration could be implemented in the code, giving the user the possibility to get better results. It is not necessarily required to calibrate the camera every time the user wants to get colour values. More than that, it might be difficult to find perfect red, green and blue objects every time the application is being used.

Calibration could be done in a way that the user is required to point at an object with solid red, blue or green colour and touch the screen for the confirmation. After it has been done, the result of the colour is being recorded, and in future colour picking sessions it would be used as a reference for colour analysis. For example, current red has the value of {255; 0; 0} in RGB form. But it is a “perfect” colour that is very difficult to see in real life. So the user finds an object, with good lighting that is red in his/her opinion, chooses the “calibrate” option from the menu, then “red” and takes the colour sample. The smartphone or tablet analyses it and shows the result of {250; 10; 30}. This result is recorded as new – temporary “red” and would be used as a new reference instead of the old {255; 0; 0} value.

This method would make results more clear, closer to result that is visible by the human eye. This kind of calibration should be done for different type of lighting, because white colour balance is what affects the result. Calibration might be organised in the pre-sets – indoor light, sunlight, night view, etc. The pre-sets might be saved, modified and removed if necessary, giving additional flexibility for the user. It would be a good idea, to have a library of pre-sets that could be imported in the program for the use.

Also, it might be necessary to have different pre-set libraries for different devices, or at least manufacturers. A good example would be having libraries, such as: Samsung devices 5 mpx, Samsung devices 8 mpx, etc.

If this was done, the results would be better representing the same colour that is visible for the human eye. Unfortunately all these calibrations would make the application heavier and less user-friendly, since it would require more interactions from the user.

5 Conclusions

The “Camera app” has showed itself as an application that works on every device that has Android version starting from 2.2.x and ending with the latest 4.1. It works perfectly on devices with different screens and screen sizes, adjusting its user interface to any device. It is compatible with all the cameras that the devices had, independent from the manufacturer, camera sensor resolution and optics. It was able to get the result on any gadget it was launched on and provide the necessary values.

Tests were done using the same version of application on 5 different devices, from many vendors. The testing environment for all the devices was the same, with artificial light source. Samples were taken from the same distance. Nevertheless, the results were not very precise, and in some cases did not show colour that is even close to the one on the testing object. Comparing the results with reference colour, we could see a significant difference and sometimes unexpected results that are not dependent on any specific parameter of the device.

In order to avoid such results in future it has been decided that the application requires some sort of calibrating function for the device. Since the device is supposed to work as professional colour measuring equipment, it is necessary to prepare it for this purpose in the same way as commercial spectrophotometers.

The goal of the project was to compare different devices using “Camera app” with each other and with “ideal reference”. This part of the project was successfully done providing us with the ideas and information that is necessary for the future development of the application. The result is clear: Android based devices can be used for camera recognition, using current version of “Camera app” recognition algorithm, with addition of calibration functionality with user-friendly interface.

References

1 Homann JP. Digital Color Management. 3rd edition. Berlin, DE: Springler; 2009.

2 Apollo 11 Mission Report (PDF). NASA; 1969.

3 Ohno Y. CIE Fundamentals for Color Measurements. Vancouver, CA: International Conference on Digital Printing Technologies; 16 October 2000.

4 Fairchild MD. Color Appearance Models. Weinheim, Germany: Wiley; 1998.

5 Allen D, Cooksey C, & Tsai B. Spectrophotometry [online]; 2010.
URL: http://www.nist.gov/pml/div685/grp03/spectrophotometry.cfm Accessed 29.04.2013.

6 Williams B. A History of Light and Lighting [online]; 1999.
URL: http://www.mts.net/~william5/history/hol.htm. Accessed 29.04.2013.

7 Google, Google AdMob Ads SDK [online]; 2012.
URL: https://developers.google.com/mobile-ads-sdk/. Acessed 29.04.2013

8 Stokes M, Anderson M, Chandrasekar S, Motta R. A Standard Default Color Space for the Internet – sRGB, Version 1.10; November 5, 1996.

9 Savage A, MythBusters [online]; October 7, 2012.
URL: https://twitter.com/MythBusters/status/255104488661192704. Acessed 29.04.2013.