

PALACKY UNIVERSITY OLMOUC
FACULTY OF SCIENCE
DEPARTMENT OF OPTICS



**Development of a Workflow
for Imaging of Works of Art
and Subsequent Image
Processing**

BACHEROL'S THESIS

Daniela Růžičková

Study programme: B1701 Physics
Field of study: Optics and Optoelectronics
Form of study: Full-time
Supervisor: Assoc. Prof. Karel Lemr, PhD
Workplace: Joint Laboratory of Optics
Thesis submitted on:

UNIVERZITA PALACKÉHO
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA OPTIKY



**Vývoj pracovního postupu
pro snímkování uměleckých
děl a následné zpracování
obrazu**

BAKALÁŘSKÁ PRÁCE

Daniela Růžičková

Studijní program:	B1701 Fyzika
Obor:	Optika a optoelektronika
Forma studia:	Prezenční
Vedoucí:	doc. Karel Lemr, Ph.D.
Školící pracoviště:	Společná laboratoř optiky
Práce odevzdána dne:

Abstract

Multispectral imaging is widely used in cultural heritage analysis by art historians and conservators. Imaging modes are divided according to imaged spectral range specified by filters. In this thesis, performed imaging modes are presented and the processing enhanced by machine learning is explored, described and discussed.

Keywords

Machine learning, Multispectral imaging, Works of art, Image processing

Abstrakt

Multispektrální snímkování je hojně využívané restaurátory a historiky umění v analýze kulturního dědictví. Jednotlivé snímkové režimy jsou rozděleny podle snímkaného rozsahu vlnových délek určeného pomocí pásmových filtrů. V rámci práce jsou představeny provedené snímkové režimy. Dále je popsáno a diskutováno zpracování obrazu pomocí strojového učení.

Keywords

Strojové učení, Multispektrální snímkování, Umělecká díla, Zpracování obrazu

Acknowledgments

I would like to sincerely express my deepest gratitude to my supervisor Assoc. Prof. Karel Lemr, PhD, for his patience, the time he devoted to our consultations and for all the advice he gave me. I also thank all the colleagues from Joint Laboratory of Optics and Department of Analytical Chemistry who are involved in multispectral imaging as well as the art historians from Palacky University and Olomouc Museum of Art. I am thankful to Petra Růžičková, MSc, and Jana Uhlová, MSc, for help with proofreading. A special thanks belong to my dear mate Anežka Dostálová, BSc, all my family and my boyfriend for the support throughout my studies.

This research was financial supported by the Arteca project (CZ.02.1.01/0.0./0.0./17_048/0007378).

Declaration

I hereby declare that I have written this Bachelor's Thesis by myself, while being supervised by Assoc. Prof. Karel Lemr, PhD.

Contents

1	Introduction	1
2	Developing Imaging Workflow	5
2.1	Scene Preparation	5
2.2	Camera Setting	7
2.3	Imaging Modes	9
2.3.1	Software Extrapolated Multispectral Imaging Mode (Using Proficolore Filters A and B)	9
2.3.2	Hardware IR Reflectance Imaging Mode (Using Filters IR1, IR2, IR3)	10
2.3.3	Hardware UV Reflectance Imaging Mode (Filter A + Flashes with UV Pass Filters)	11
2.3.4	Ordinary Visible Imaging Mode (Filter A + UV-IR Cut)	11
2.3.5	UV Induced VIS Luminescence Imaging Mode (Flashes with UV Filters + UV-IR Cut + Filter A)	11
3	Processing	13
3.1	Software Extrapolated Multispectral Imaging Mode Processing	15
3.1.1	Cross-validation	16
3.2	Ordinary Visible Imaging Mode Processing	19
3.2.1	Cross-validation	21
4	Case Study	25
4.1	Initial Processing Steps	26
4.2	ANN Enhanced Modes	29
5	Conclusion	33

Chapter 1

Introduction

Multispectral imaging (MSI) is a process of visualising imaged objects in multiple spectral bands even outside the visible range of the electromagnetic spectrum. These bands can be selected by band-pass filters on either the detector or illumination source. Using a set of single-band filters, one can obtain useful results. Such implementation is, however, very time consuming. An alternative strategy is to reduce the number of filters to only a few with distinct spectral transmissivities across the entire imaged spectral range. The signal obtained through these filters together with the RGB channels defined by the Bayer mask on the detector can be numerically processed to estimate reflectivity in individual spectral bands. Whatever strategy is chosen, MSI allows to extract additional information about the imaged object namely at wavelengths inaccessible to the human eye. That is why it has found its practical usage in cultural heritage analysis by art historians. [1, 2]

Cultural heritage defined us, defines us and will define us in the future, so it is of paramount importance to learn about it as much as possible. MSI methods are nowadays widely used by art historians and conservators [3]. Although MSI methods do not replace the capability of spectroscopic methods in general or chemical analyses, MSI has the advantage of being a non-invasive and affordable method for analysing important works of art without the necessity of taking samples and thus without damaging the artwork as invasive methods usually do. Using MSI we can map and identify historical pigments, retouches and underpaintings, which cannot be normally observed, because it is beyond the capabilities of the human eye [4]. It does not always allow to identify all different pigments, because they are mixed and overlapped in multiple layers, but it is successful for those, which present particular behaviour in

the imaged spectral range (most commonly from ultraviolet to infrared parts of the electromagnetic spectrum). The possibility of imaging underpaintings depends on the upper layer of the painting (whether the layer absorbs, reflects or transmits infrared radiation) and on the base layer, which can also absorb or reflect infrared radiation. It is also of a significant benefit to have the artwork digitized, because cracks in the layers of the painting and retouches are inimitable and thanks to that, historians of art can determine for example its authenticity, i.e. at what time the artwork was created, who created it and whether the artwork is original or fake [5].

There are multiple studies on the applications of the ideas mentioned above and some of the famous artworks have already been imaged using MSI. Let's mention for instance research and conservation project called Operation Night Watch (Rembrandt's The Night Watch painting), which started in 2019. The artwork measures 379.5 cm in height and 454.5 cm in width and two platform lifts made possible to study the entire canvas. 56 scans were taken using macro-XRF scanner to analyse the chemical elements in the painting such as iron, potassium, cobalt and calcium. Each scan took 24 hours. Imaging of The Night Watch led to some remarkable conclusions such as that Rembrandt first painted feathers in the canvas, but later he repainted them. Another discovery suggests that he used a pigment containing toxic arsenic or cobalt (smalt pigment), which comes from a blue glass that he often used. It gives an interesting insight into his painting process. [6]

As briefly mentioned above, MSI methods have also been used to identify historical pigments, which is the goal of many research articles. For example the article from Heritage science journal by Antonio Cosentino illustrates a flowchart method based on a collection of 56 historical pigments, which was applied with a special medium used in watercolor, so pigments are most likely not overlapped. MSI images were then taken in ultraviolet (360–400 nm), visible (400–780 nm) and infrared (780–1700 nm) spectral bands and certain pigments were reliably identified. [7]

The main purpose of this thesis is to image a work of art using specially modified digital single-lens reflex camera in the range from 300 nm to 1000 nm while using various filters and to process the obtained photos using machine learning (artificial neural network) for further usage in cultural heritage analysis. In the first chapter, the imaging workflow and the setup for imaging is introduced. Then all performed imaging modes are specified and discussed including namely specific filters used. The

subsequent software processing of all imaging modes is described step by step in the second chapter. Finally, the whole process is demonstrated on an example work of art (The Death of Sophonisba by an unknown artist) and the results are summarized in the last chapter.

Chapter 2

Developing Imaging Workflow

Procedures described below are common for all imaging modes. The imaged object was in our case a painting on canvas, but it could be other 2D object of art.

2.1 Scene Preparation

Preferably, the imaged object should be carefully removed from its frame if possible. If the imaged object is protected by a climaframe, which ensures the isolation of the painting from the outside air and prevents damage of the object by temperature changes, it should be removed as well. The object should be placed on an easel and attached to it if needed. Any suitable tool such as a thread can be used.

Illumination sources are deployed homogeneously as depicted in Figure 2.1, typically under 45 degrees with respect to the object's plane and it should be checked that the illumination does not cast shadows or cause specular reflections on the imaged object at least from camera's perspective. Using a power meter with a set of optical filters one can verify that the illumination contains all required spectral bands. If a combination of several types of light sources is needed to cover the required spectral width, one also needs to check that the imaged object is covered spectrally homogeneously. There are several types of illumination sources. For instance incandescent light bulbs are used to provide infrared illumination, because their emission peak is situated at 1300 nm. Light tube panels with white light tubes and UV light tubes provide visible and ultraviolet lighting. A set of remotely controlled flashes can be used as well, especially when bandpass filters are needed to be placed on the

sources.

Despite all measures taken, the illumination is never perfectly homogeneous. To compensate in post-processing, a set of 94% reflectance white cards is placed in the scene around the object. The distance between the object and a camera has to be the same as the distance between the cards and the camera, which means that surfaces of these cards should lie in the same plane defined by the object. At least one card should be placed near each corner of the object, so that imaginary lines connecting neighbouring cards do not crop the imaged object. In our experience, fastening these cards using tape or clothes pegs that are placed on the object is a practical solution. A black background can be placed behind the object if better contrast with the white cards is required. A 99% white reflectance standard (in our case by Labsphere [8]) and a black hole (in our case by Thorlabs [9]) should be placed near the imaged object. All this equipment should be visible in all photographs, and shall be within the confines of imaginary lines, connecting neighbouring white cards.



Figure 2.1: 3D model of the scene preparation.

A camera (in our case Nikon D850) is placed on a tripod with its

lens optical axis perpendicular to the canvas and the canvas should be in the centre of the camera's field of view encompassing the white cards, white standard and the black hole. Thermometer should be placed near the imaged object too to monitor temperature and to prevent damage by overheating by illumination sources.

2.2 Camera Setting

The camera is specially modified for multispectral imaging. Its inner UV-IR cut filter is removed. We managed to get the camera spectral sensitivity of RGB pixels similar to our device given by the manufacturer [10] in dependence on wavelength (see Figure 2.2). We suppose there is not any noticeable difference in sensitivity compared to our camera.

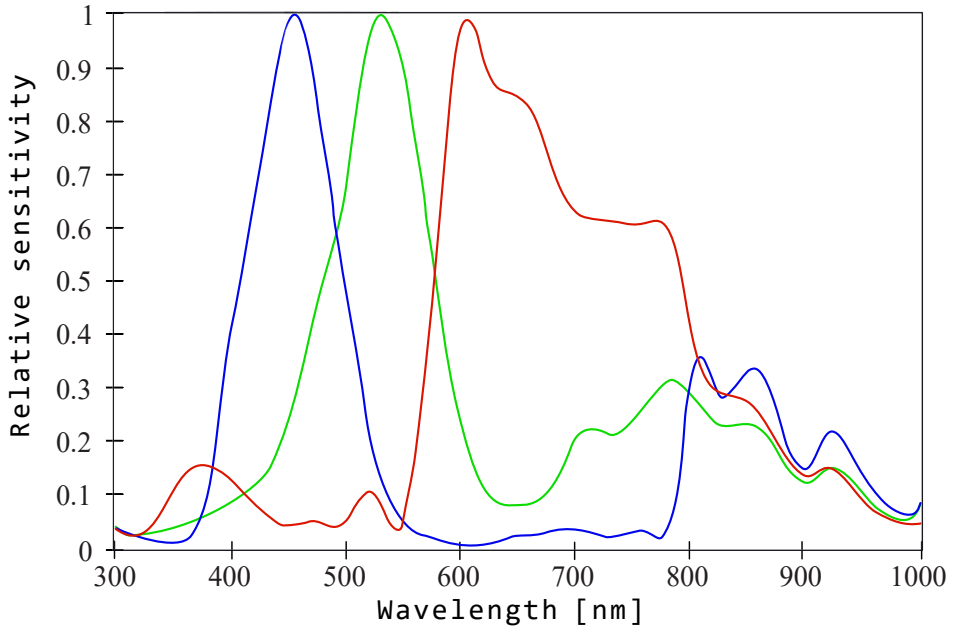


Figure 2.2: Camera sensitivity of RGB pixels as given by Profilocore [10].

All optical components have to be checked and cleaned using a professional cleaning kit or compressed air if needed to get rid of impurities.

Then setting of the camera must be checked¹. The images should be stored as a 14-bit RAW file format (or better if available). Because

¹Note that some of the settings that follow are specific to our camera.

this is an uncompressed version of the image file where the camera saves directly the image data from the sensor in an unedited and uncompressed format to the memory card. We thus retain all information present in the data.

Active D-lightning must be set OFF and *Set picture control* set to *Standard (SD)*. *Silent mode* (silent shutter) is turned on except when using flashes, where it has to be turned OFF (specifically in case of our camera by using the “i” button).

The camera is connected to a computer using a USB cable. A wireless way through a Wi-Fi was not as convenient as expected. Camera’s access point was constantly turning off after ten minutes of inactivity and sometimes the connection failed, so turning it on again several times in a row manually was impractical.

When the camera is successfully connected to a computer, it is time to start Python² program `cancli.py` (code attached in electronic version). The program is started in a Linux terminal using command

```
python3 -i cancli.py -p,
```

where the parameter `-p` activates the camera and the `-i` parameter opens python in an interactive mode. This program allows to issue commands such as *setIso*, *setSpeed*, *setFnumber*, *focus*. These functions allow to set ISO, shutter speed, aperture number (typically f/11) and even drive the focusing motors by a precise amount of steps according to conditions as needed. Focusing is much easier and more precise with the function *previewHR* of the `cancli.py` compared to manual focusing and using either the viewer or built-in camera display. The *previewHR* command captures a jpeg image in full resolution and transfers it to the computer where it gets immediately displayed on the screen. Iterating the *previewHR* and the *focus* command allows to establish best focus.

Finally, after camera setting, the photo is taken using `cacli.py`’s command *capture*. When the image is captured, proper exposure is visualised and displayed on a computer screen. Pixels with values too close to exposure saturation (greater than 94 % of dynamic range) are depicted red while pixels in range of 82–94 % of dynamic range are depicted green. Observing the white cards as well as the white standard coloured in green assures the best exposure setting providing some reasonable margin. Manual inspection instead of machine-only processing is needed because of overexposed pixels corresponding to direct reflections

²<https://www.python.org/>

from metallic objects (e.g. screws) are present in the image, but do not represent a problem.

2.3 Imaging Modes

2.3.1 Software Extrapolated Multispectral Imaging Mode (Using Profilocore Filters A and B)

The colorchecker we use is a chart consisting of patches of colour samples arranged in three rows and 12 columns³. Spectral reflectances of each colour sample are given by the manufacturer in the range from 300 to 1000 nm with a spectral resolution of 1 nm.

Two camera lens-mounted filters are used in this imaging mode, those are filter A and filter B by Profilocore (for spectral transmissivities see Figure 2.4). Using these two filters one can obtain six independent channels (2 filters \times 3 RGB channels) that can further be used to estimate spectral reflectivities in seven partially overlapping bands [Profilocore hyperspectral matching functions [10] (see Figure 2.3)]. The exact process of this estimation is described in Chapter 3.

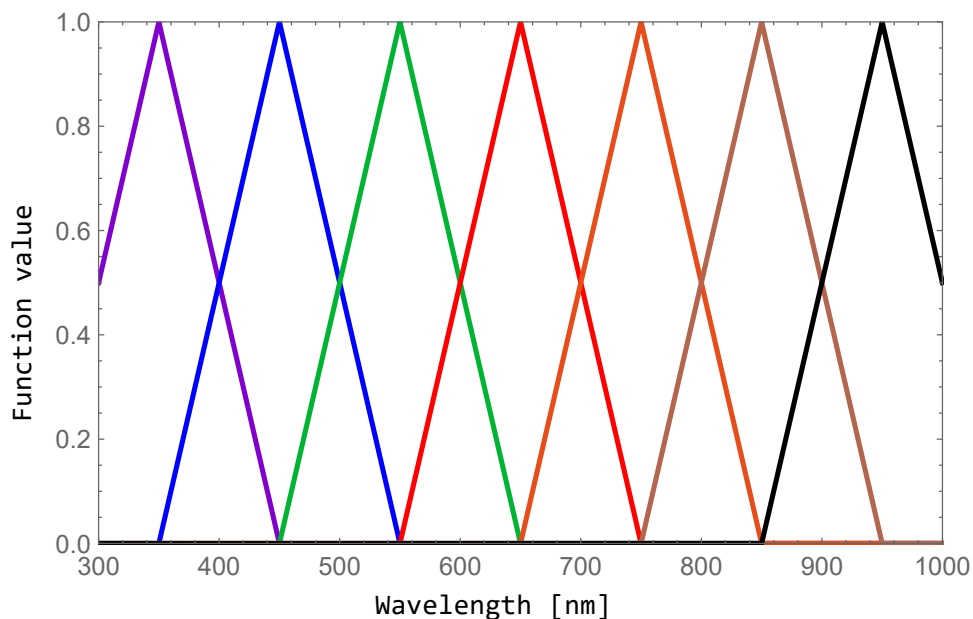


Figure 2.3: Hyperspectral matching functions by Profilocore [10].

³Note that the arrangement is specific to our colorchecker by Profilocore.

The estimation described above requires images taken with the two filters that contain a well specified colorchecker. Unless the colorchecker fits into the view alongside the imaged object, additional images of the colorchecker need to be taken under the same conditions as the object images.

The colorchecker should be placed on an easel in front of the imaged object. As well as the white cards, the surface of the colorchecker has to lie in the plane defined by the object perpendicular to the camera lens optical axis (see Figure 2.1).

Whenever a filter is changed, camera setting, mainly focus, must be repeated. If it were not refocused, the final image would be blurred. The focal length of the camera lens depends on the wavelength of the incident light passing through the filter especially outside of the visible range. Similarly, the exposure time needs to be adjusted for any combination of filter and illumination.

One of the problems of this imaging mode is the fact, that the filter B transmits UV and IR part of electromagnetic spectrum at the same time and the ratios of RGB values in portions of these regions are similar, so one can object, that it is not clear which part of the spectrum caused the detection event. We, thus, believe that the estimation may be ambiguous, unstable or noisy. Further to that, it is impossible to fully focus in UV and IR bands simultaneously at least with our lens. A third problem arises from the fact that the amount of detection events corresponding to IR radiation around 950 nm and to UV radiation around 300 nm is comparable to noise level (while simultaneously preventing overexposure in the entire captured spectral range). That is why another imaging modes (we call them “hardware modes”) are realized with bandpass filters specific to each part of the spectrum for improved accuracy.

2.3.2 Hardware IR Reflectance Imaging Mode (Using Filters IR1, IR2, IR3)

Infrared bandpass filters (for spectral transmissivities see Figure 2.4), that let infrared radiation pass through to the camera lens and block most of the visible light spectrum are used, specifically filters IR1, IR2 and IR3 (transmitting wavelengths from about and above 750, 850 and 950 nm respectively). Three photos are taken gradually with these filters attached to a camera lens. Combination of edge filters with decreasing detection efficiency creates an effective band-pass filter.

2.3.3 Hardware UV Reflectance Imaging Mode (Filter A + Flashes with UV Pass Filters)

Using remotely controlled flashes with UV filter attached to them, the operator captures an image is captured with filter A attached to the camera lens (for spectral transmissivities see Figure 2.4). Using filter A may seem unnecessary, but transmissivity of flashes with UV filters at 350 nm is 0.72 and at 750 nm it is 0.12, which gives the ratio of transmissivity at 350 nm vs at 750 nm 6:1 and the data may be affected by red/IR contribution. Better results are obtained when filter A is attached. Although transmissivity of the filter A at 350 nm is 0.64 and, thus, overall transmissivity of the filter A and UV flash filters at 350 nm falls down to 0.46, which may seem worse than without filter A. At 750 nm transmissivity of filter A is 0.01 and that means that overall transmissivity of the filter A and UV flash filters is only 0.001. The final ratio of transmitted light is about 460:1, so photograph in ultraviolet spectral band is uncontaminated by other parts of the spectrum.

2.3.4 Ordinary Visible Imaging Mode (Filter A + UV-IR Cut)

Combination of filter A and UV-IR cut (for spectral transmissivities see Figure 2.4) filter is used, which blocks the wavelengths outside the visible range of the spectrum. Usually two photos are taken, with and without colorchecker (unless the colorchecker fits into the view alongside the imaged object).

2.3.5 UV Induced VIS Luminescence Imaging Mode (Flashes with UV Filters + UV-IR Cut + Filter A)

The same filters as in the Ordinary visible imaging mode (i.e. filter A and UV-IR cut) plus remote controlled flashes with UV filters are used (for spectral transmissivities see Figure 2.4). The imaged object absorbs UV radiation from the flashes and by a process of luminescence emits at longer wavelengths. This imaging mode requires complete darkness in the room to ensure that luminescence is not contaminated by any other external illumination sources. Filters attached to a lens cause camera to detect emitted light only in the visible range.

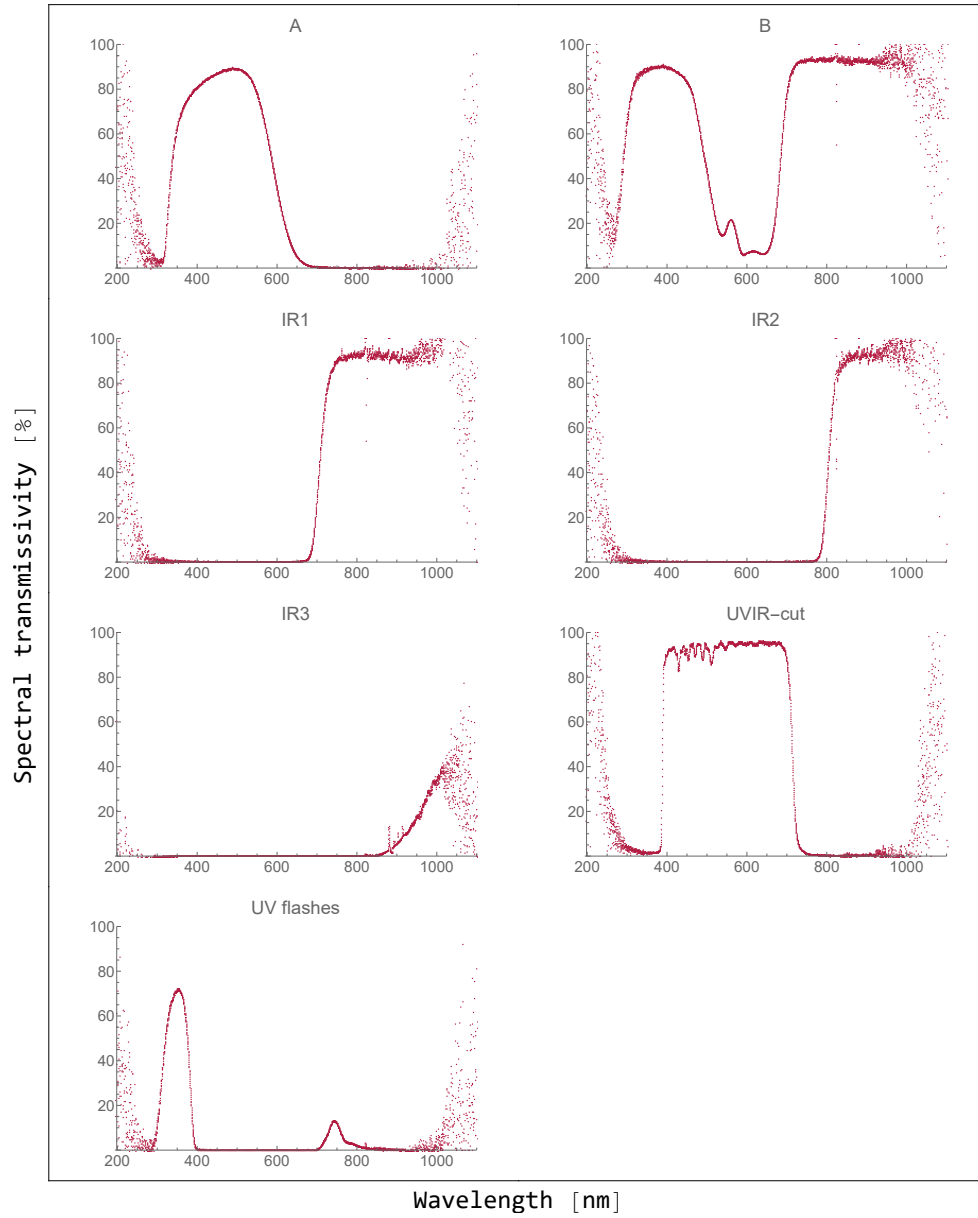


Figure 2.4: Spectral transmissivities of all filters used.

Chapter 3

Processing

Processing described below is common for all imaging modes. If any imaging mode needs an additional processing step, it is specified.

A Bayer mask (RGB colour filters) is used in digital image sensors, so that each pixel is filtered to record with different spectral sensitivity. Raw data from the sensor need to be converted from RAW file format into RGB coordinates. A demosaicing algorithm is used to interpolate RGB values for each pixel. In our workflow, we use a free software called Rawtherapee¹. Rawtherapee is set to perform only the following image processing tasks: expected noise level subtraction, demosaicing, approximative white balancing (multiplication coefficients for RGB values all close to 1) and saving RGB data with Adobe RGB profile in a 16-bit TIFF file.

While changing filters, we may accidentally move the camera even without noticing so the same pixel coordinates in particular photos do not correspond to each other. That is why image registration (alignment) has to be done. To align overlapping images we use Hugin² (free software) package command-line tool called `align_image_stack` with command

```
align_image_stack -a reg_ -v -m -i --use-given-order  
    <image_file_names>,
```

where `-a` causes the output images to be saved with `reg_` prefix, `-v` (means verbose) prints progress messages, `-m` optimizes field of view for all images, except for first, `-i` optimizes image center shift for all images, except for first and using `--use-given-order` the image order as given

¹<https://www.rawtherapee.com/>

²<https://sourceforge.net/p/hugin/hugin/ci/default/tree/>

on the command line is preserved. In grayscale regimes (where it is not necessary to have information of all RGB values), only one channel is extracted. For example in UV reflectance regime only the red channel is extracted using command

```
mogrify -separate -channel R <image_file_name>,
```

where `mogrify` command-line tool is a part of `ImageMagick` utility software. Separating an image channel is achieved using `-separate` and applying option to select only red channel using `-channel R`.

To compensate for non-homogeneous illumination, points of interest in the image containing the white cards, white 99% and 96% reflectance standards and the black hole using `markItems.py` are manually marked, which is started through the command such as

```
python3 markItems.py <image_file_name>.
```

This process opens a preview on display and one can move a mouse cursor to the required area and press a hotkey to mark the point. The hotkeys are *b* to mark 94% reflectance white cards, *y* to mark the black hole, *n* to mark 96% white reflectance standard and *m* to mark 99% white reflectance standard. Upon closing the `markItems.py`, program saves the points of interest coordinates into an extra file, because white cards or other reflective standard might have different reflectivity for different spectral bands it is recommended to use only one type of reflective objects (for example only white cards) for this procedure. In the next step, images are processed using a program `processTiff.py`, which is started using command

```
python3 processTiff.py <image_file_name> -H -i -w.
```

Using various command line arguments we are able to carry out following steps: black hole subtraction (`-H`); flat-field interpolation based on the estimation of the illumination intensity using white cards (`-i`); exposure setting so that white reflectance standard has the right RGB value (`-w`). Make sure that photos that are taken with the same filter (with and without colorchecker) are processed in the same way. The `processTiff.py` program outputs images using Adobe RGB color space.

Following steps need to be done only for the software extrapolated multispectral imaging mode and ordinary visible imaging mode.

Using our program `findColorchecker.py`³, started through command

```
python3 findColorChecker.py <image_file_name>,
```

the colorchecker is machine detected in each image containing the colorchecker. The program also returns RGB values of each patch and saves it into a data file `colorchecker_output.dat` (when there are more photos containing colorchecker, the RGB values are written in extra columns). These RGB values of the patches are further important for both multispectral (software extrapolated multispectral imaging mode) and color processing (ordinary visible imaging mode). In combination with colorchecker definition data files it is possible to train an artificial neural network (ANN).

3.1 Software Extrapolated Multispectral Imaging Mode Processing

As mentioned in Chapter 2.3.1, multispectral processing requires images taken using filters A and B. Our program `processMultispectral.py`³ is started through the command line similarly to the previous programs

```
python3 processMultispectral.py <image_file name>.
```

The industry standard by Profilocolore defines the Hyperspectral Matching Function $HMF_i(\lambda)$, where $i \in (1, 7)$, as depicted in Figure 2.3. From spectral reflectivity $R_j(\lambda)$ for each of the $j \in (1, 36)$ patches, the 252 hyperspectral reflectivities HR_{ij} can be calculated

$$HR_{ij} = \sum_{\lambda=300}^{1000} R_j(\lambda) \cdot HMF_i(\lambda). \quad (3.1)$$

These values are saved into a definition file `colorchecker.refl`. The ANN program takes the colorchecker photo data file (the measured $RGB_{A,B}$ values of the patches obtained from photos taken using filters A and B) `colorchecker_output.dat` and a colorchecker definition file `colorchecker.refl` and trains to predict of hyperspectral responses

³This program was developed by the author of this thesis.

$\overline{\text{HR}}_{ij}$ based on the $\text{RGB}_{A,B}$ values. The $\overline{\text{HR}}_{ij}$ are then compared to the definition HR_{ij} through a mean squared error

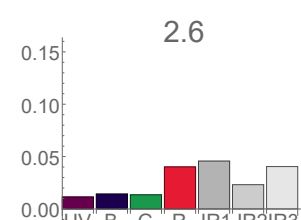
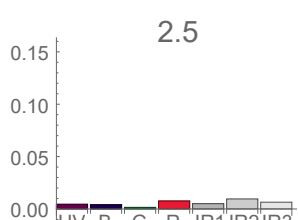
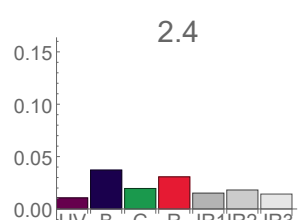
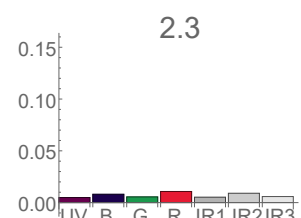
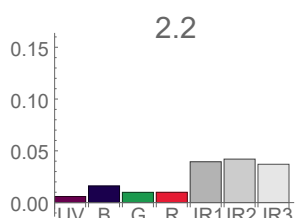
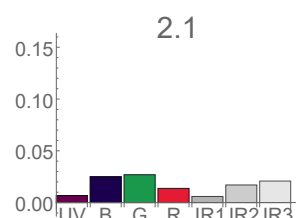
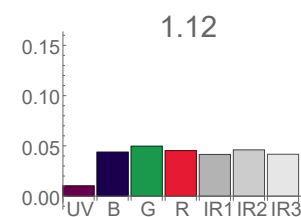
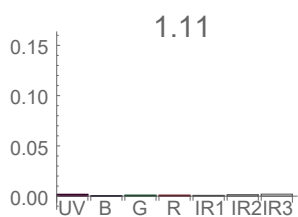
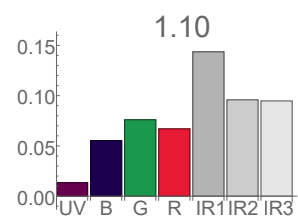
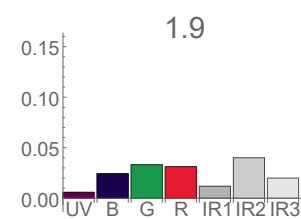
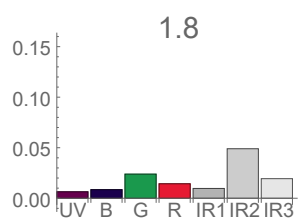
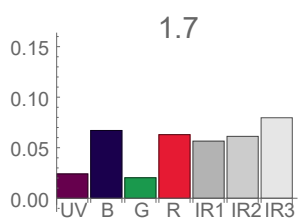
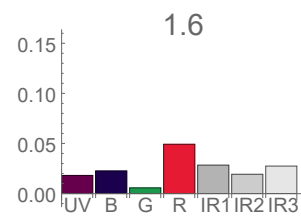
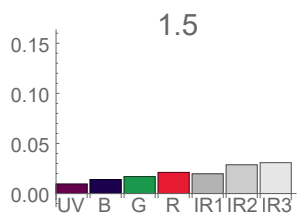
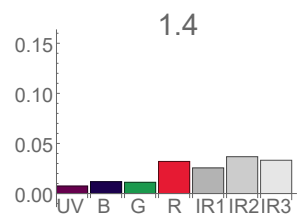
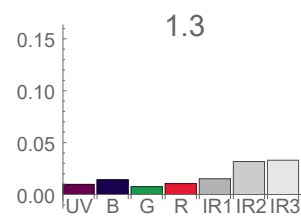
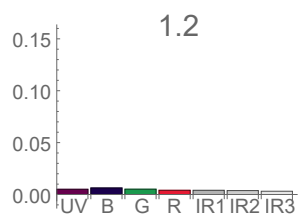
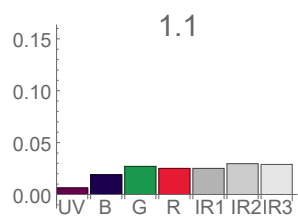
$$\Delta R^2 = \sum_{i,j} (\overline{\text{HR}}_{ij} - \text{HR}_{ij})^2. \quad (3.2)$$

The ANN is trained to minimize ΔR difference between the predicted and the expected (definition) values. The error cannot be completely eliminated, but we are trying to minimize it as much as possible. We can save the ANN parameters for the future usage without the necessity of repeating the ANN training. Then the prediction is applied to all pixels of the actual image. Result of multispectral processing is in our case seven photos in seven spectral bands. Those are ultraviolet (UV), blue (B), green (G), red (R) and three infrared (IR – 1, 2, 3), corresponding to the seven hyperspectral matching functions.

3.1.1 Cross-validation

Because a small training dataset is available, cross-validation technique (holdout method), is performed to establish how well the ANN was trained. Holdout method works out by removing part of the data set (in our case one patch of the colorchecker) and letting the rest of data to be used for training to get the predictions. The prediction is applied to the dataset, which was removed from the training and we can evaluate how the ANN training went. This is repeated several times for each patch on the colorchecker and the data set containing the differences ΔR between the predicted and the definition values is obtained. The results for the Software extrapolated multispectral imaging mode processing are depicted in Figure 3.1 as bar charts, where there are the ΔR values for each patch and the seven columns correspond to the seven hyperspectral matching functions (see Figure 2.3). As we can see in Figure 3.1, the largest difference, that occurs in four cases out of total 36, is close to 0.15.

But if we take a look at Figure 3.2, the average values for each of the spectral band do not exceed 0.05, so we can say that the ANN training was successful. The average difference increases as we go from UV to IR3 and as we stated in Chapter 2.3.1, there are reservations, that this estimation may be ambiguous and affected by noise also due to decreasing detection efficiency, so it is not surprising that the ambiguity has to show somewhere, in this case mostly at parts IR2 and IR3.



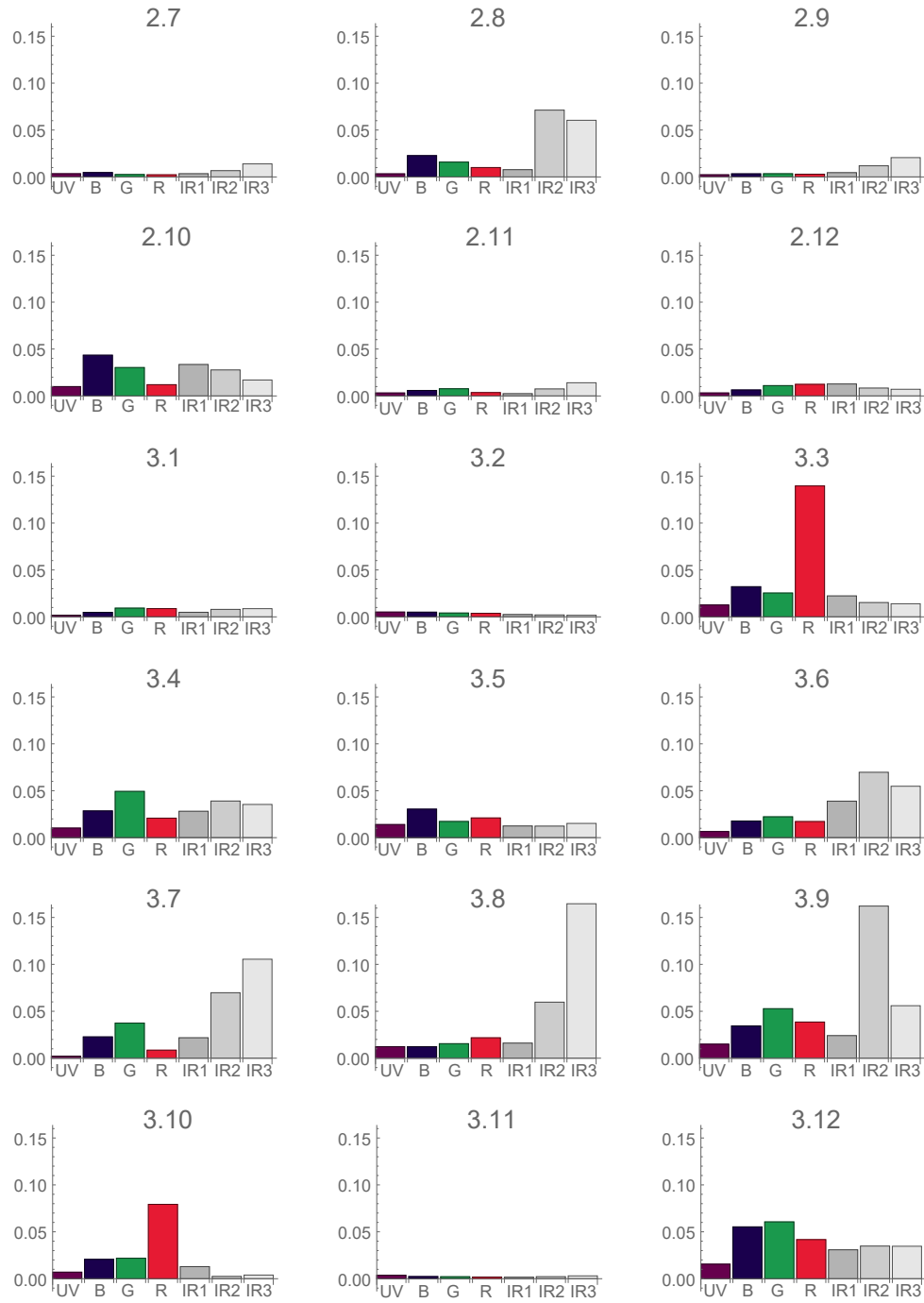


Figure 3.1: Results of cross-validation holdout method for Software extrapolated multispectral imaging mode. ΔR values are depicted as functions of seven hyperspectral bands.

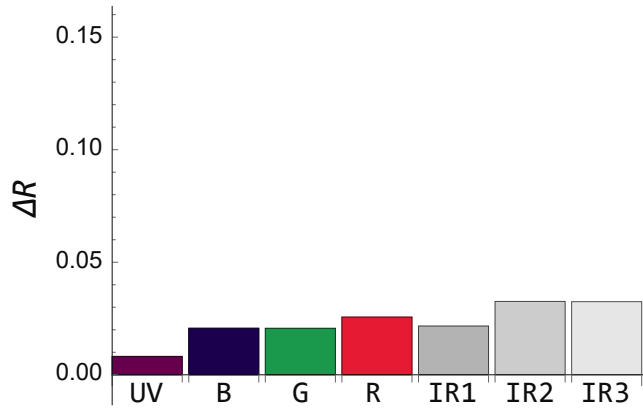


Figure 3.2: The average values of the cross-validation for Software extrapolated multispectral imaging mode. ΔR value is depicted as a function of seven hyperspectral bands.

3.2 Ordinary Visible Imaging Mode Processing

To estimate a colorimetrically correct photograph (in the visible part of the spectrum) of the imaged object, a photograph taken with filter A and UV-IR cut (ordinary visible mode) is processed using `processColors.py`⁴ started on a command line

```
python3 processColors.py <image_file name>.
```

Similarly to multispectral processing above, the ANN makes use the file `colorchecker_output.dat`, but this time with a definition data file `colorchecker.dat`. The `colorchecker.dat` contains factory measured XYZ coordinates of each patch, calculated as

$$[x, y, z]_j = \sum_{380}^{780} R_j(\lambda) \cdot [\bar{x}, \bar{y}, \bar{z}](\lambda), \quad (3.3)$$

where $[\bar{x}, \bar{y}, \bar{z}](\lambda)$ stands for CIE XYZ Color Matching Functions (see Figure 3.3), which are the numerical description representing chromatic response of the “standard observer” (representing an average human’s eye chromatic response).

⁴This program was developed by the author of this thesis.

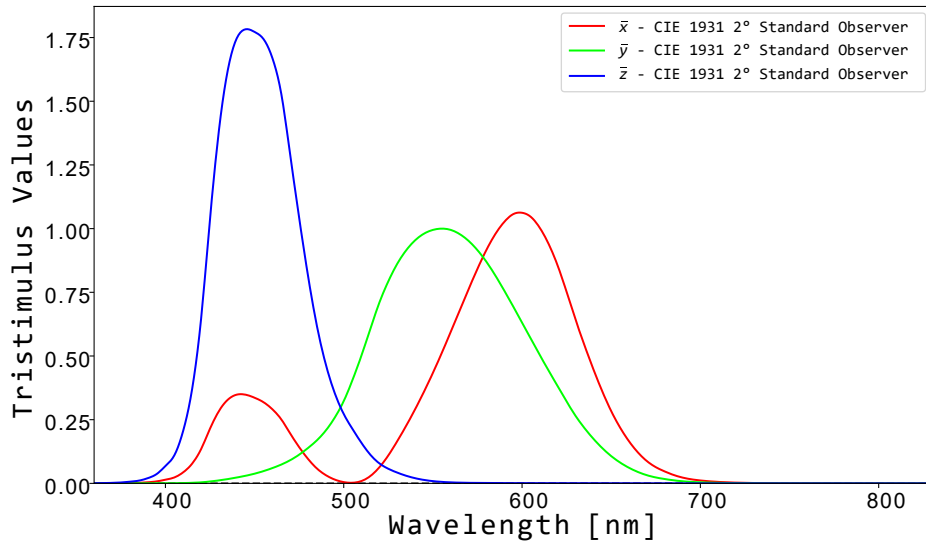


Figure 3.3: CIE 1931 Standard Observer - Color matching functions obtained using Colour Science open-source Python package [11].

Unfortunately, RGB or even XYZ coordinates cannot be used to directly determining color differences. That is why we have to convert our data into another color space. In our case, we use the $L^*a^*b^*$ (CIELAB) color space, which is the color space defined by the International Commission on Illumination (CIE) that encompasses quantitative relationship of colors. The $L^*a^*b^*$ coordinates are computed as [12]

$$L^* = 116f\left(\frac{Y}{Y_n}\right) - 16, \quad (3.4)$$

$$a^* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right), \quad (3.5)$$

$$b^* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right), \quad (3.6)$$

where X_n, Y_n and Z_n are the values of X, Y, Z for the chosen reference white ($X_n = 95.0489, Y_n = 100$ and $Z_n = 108.8840$ for Standard Illuminant D65). If we consider a variable $t = \frac{X}{X_n}, \frac{Y}{Y_n}, \frac{Z}{Z_n}$, then $f(t)$ equals

$$f(t) = \sqrt[3]{t} \text{ if } t > \left(\frac{6}{29}\right)^3 \text{ or } f(t) = \frac{841}{108}t + \frac{4}{29} \text{ otherwise.}$$

First of all, measured (obtained) RGB values of the colorchecker have to be converted into XYZ coordinates and then (while assuming of D65 illuminant and Adobe RGB color space) into L*a*b* coordinates. Definition data file is given in XYZ coordinates, so it has to be converted into L*a*b* as well. Then it is possible to measure the “distance” between the actual Lab coordinates and predicted Lab coordinates using Keras enabled⁵ formula for CIE color difference CIEDE2000 squared [13]. The ANN is considered to be successfully trained, when the average CIEDE2000 becomes less than just noticeable difference value corresponding to 2.3. Then, after learning is performed and ANN is successfully trained, prediction (transformation) is applied to all pixels of the actual image. It cannot be forgotten to convert output (predicted) L*a*b* coordinates through XYZ back to the required Adobe RGB values. Adobe RGB 1998 profile is inserted using ImageMagick. The result of the color processing is a colorimetrically correct photograph of the imaged object.

3.2.1 Cross-validation

Similarly to multispectral processing, cross-validation is performed to check the quality of the ANN with respect to color processing and the results are depicted as bar charts. Using holdout method several times for each patch on the colorchecker we obtain a dataset containing the differences between the measured (before ANN processing) and the definition data and between the predicted (after ANN processing) and the definition data. See, in Figure 3.4, columns “R” (raw = before processing) and ”P” (predicted = after processing) showing the CIEDE2000 differences for each colorchecker patch. If the CIEDE2000 difference decreases at least to about 2.3, it means that the ANN was trained successfully. The opposite situation occurs, when the difference after performed training is worse than before the training. As we can see in Figure 3.4, the worst difference after performed training is close to 7 (7.127 on the patch 3.3 exactly), which is more than three times the just noticeable difference. It is the proof, that the ANN is not 100% reliable and sometimes the training of the ANN fails.

⁵Keras enabled means that the formula can be calculated efficiently on a graphics card.

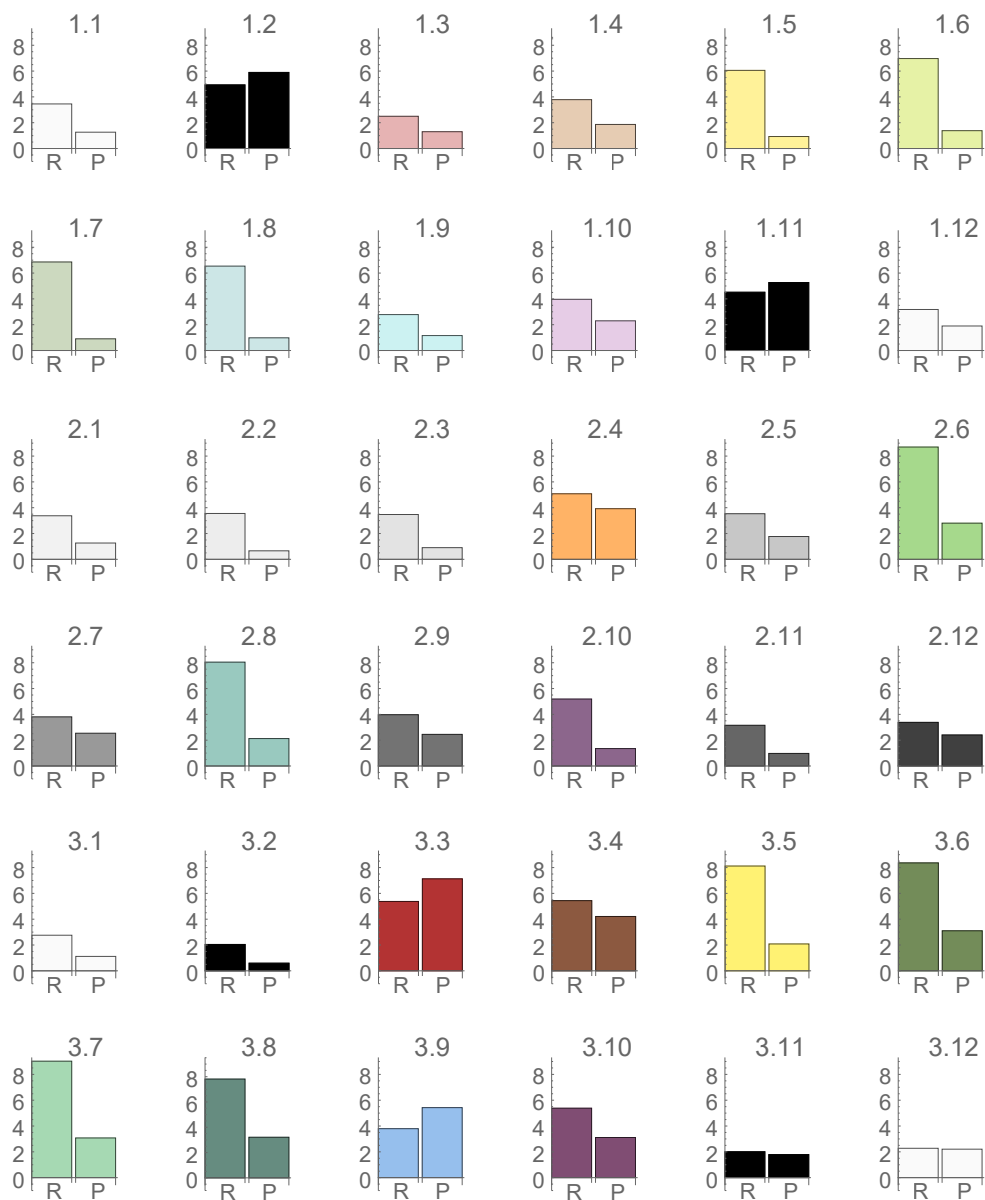


Figure 3.4: Results of cross-validation holdout method for Ordinary visible mode.

For comparison, the average difference before and after the ANN training see Figure 3.5. As obvious from the Figure 3.5, the average color difference after the ANN training is smaller than before the training and the CIEDE2000 value of 2.37 is close to the just noticeable difference. We can thus consider the ANN training as successful.

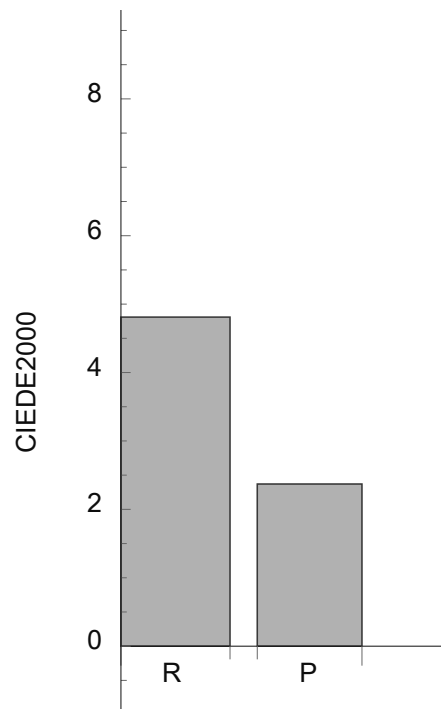


Figure 3.5: Average differences of colors before and after performed training for the Ordinary visible mode.

Chapter 4

Case Study

In this Chapter, we go over the whole process from raw image capture up to processing in practice. The imaged object was in our case the painting named The death of Sofonisba made by an unknown artist. Being the property of the archbishops of Olomouc, it was photographed at the Archbishop's Chateau Kroměříž.

We proceed step by step according to Chapter 3. For reader's better understanding of the procedure, diagrams in Fig. 4.1 and 4.2 are attached.

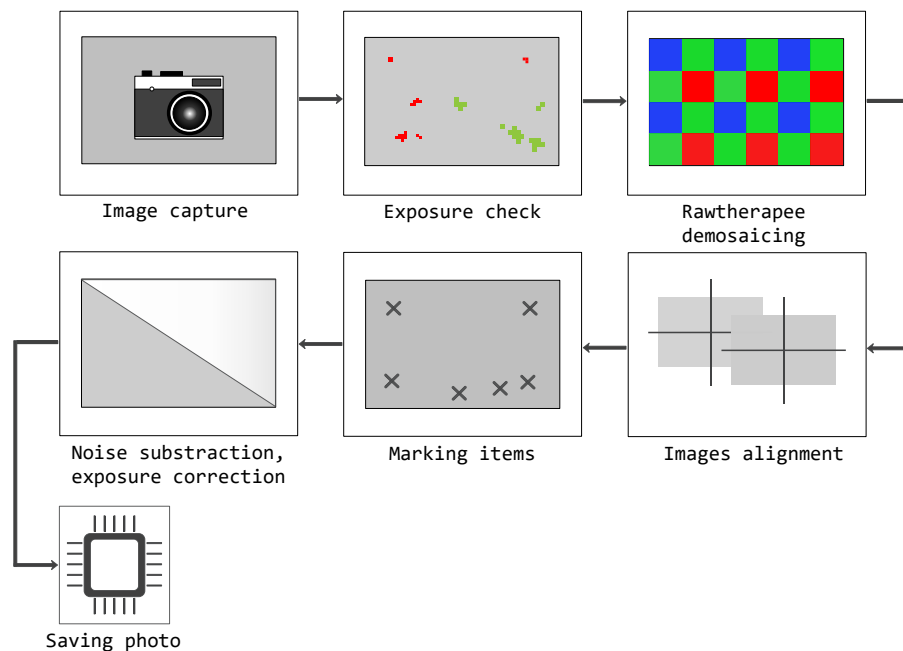


Figure 4.1: Initial processing diagram.

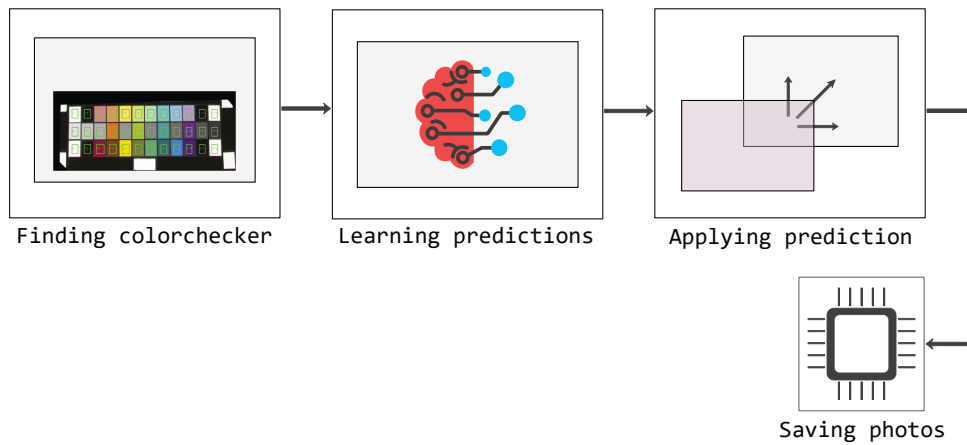


Figure 4.2: ANN enhanced modes diagram.

4.1 Initial Processing Steps



Figure 4.3: Exposure checking. Red color stands for overexposed pixels.

After capture, images in RAW file format are checked for overexposure and then, if fine, opened in Rawtherapee. For an example of overexposed

pixels inspection see Fig. 4.3. Then, a set of tasks is performed such as demosaicing and saving with Adobe RGB profile as a 16-bit TIFF file. In Fig. 4.4 one can see raw data opened in Rawtherapee, where each pixel corresponds only to one of the three (red, green, blue) colors. The resulting TIFF file is already standard color image. Note, that only approximative white balancing is implemented in this step to make sure that no channel is overexposed.



Figure 4.4: Raw data opened in Rawtherapee.

Because the same painting is imaged in multiple regimes, the TIFF files output from Rawtherapee are aligned.

Then, the 94% reflectance white cards (`card94`), the black hole (`hole`) and the 99 % white reflectance standard (`white99`) are marked in each photo taken with a different filter. The Figure 4.5 shows the preview, where all keypoints are marked. Images can now be processed using `processTiff.py` (black hole subtraction, flat-field interpolation and exposure correction) and saved into a 16-bit TIFF file with Adobe RGB profile. In Figure 4.6, there are photos taken with filters A and B before and after processing with `processTiff.py`, where photos in the first row are those one obtained from Rawtherapee in the first step and photos in the second row correspond to the processed by `processTiff.py`. Figure 4.7 shows the photo taken with filter A plus filter UV-IR cut before (1) and after (2) processing with `processTiff.py`. All steps mentioned so far concern images for all imaging modes. For images corresponding to Hardware IR reflectance, UV reflectance and UV induced VIS luminiscence

imaging modes, the workflow is now terminated. The subsequent steps do only concern image processing in Software extrapolated multispectral and Ordinary visible imaging modes.



Figure 4.5: Marking the black hole, the white standard and the white cards.

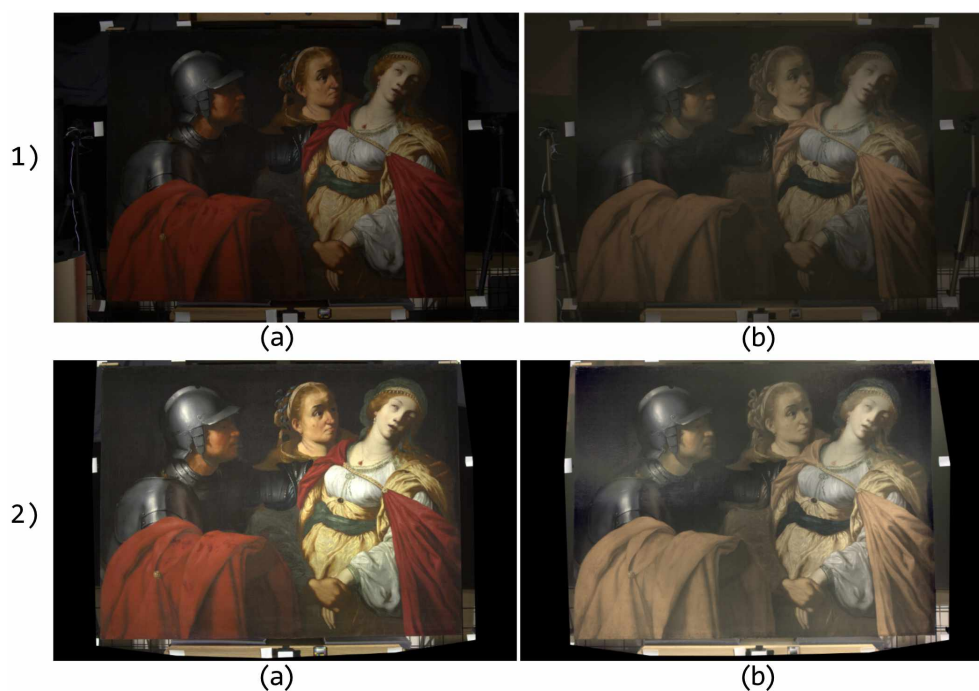


Figure 4.6: Images before (1) and after (2) performance `processtiff.py` taken with filter A (a) and filter B (b) attached to a camera lens.



Figure 4.7: Images before (1) and after (2) performance `processtiff.py` taken with a combination of filters A and UV-IR cut attached to a camera lens.

4.2 ANN Enhanced Modes

The colorchecker is found in all images containing colorchecker. In Fig. 4.8 and 4.9, there are photos of colorchecker taken with filters A, B (i.e. Software extrapolated multispectral imaging mode) and filter A in combina-

tion with filter UV-IR cut (i.e. Ordinary visible imaging mode), where each patch in each photo is framed and labeled to be able to make sure that colorchecker was found properly.

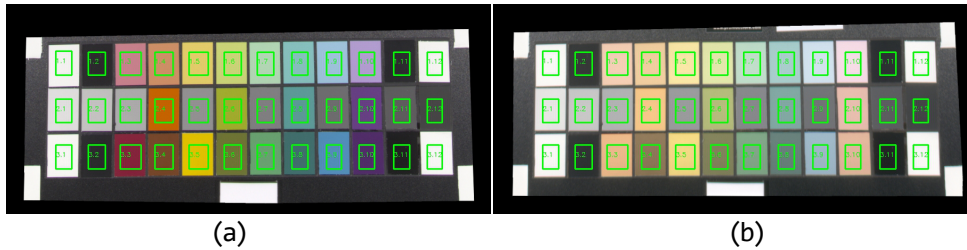


Figure 4.8: Diagnosed patches in the photos containing colorchecker taken with filter A (a) and filter B (b) attached to a camera lens.

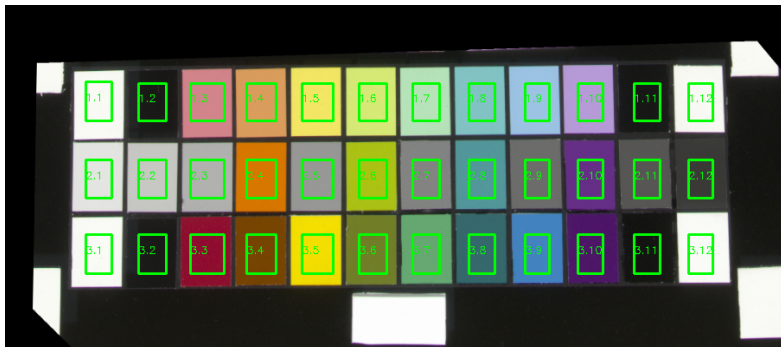


Figure 4.9: Diagnosed patches in the photo taken with filter A and UV-IR cut attached to a camera lens.

The RGB values of each patch from the photos are obtained. Finally, for Software extrapolated multispectral imaging mode, prediction of reflectancies are learned and then applied to all pixels of the photo using `processMultispectral.py`. The result of this imaging mode is a set of multispectral images from ultraviolet to infrared spectral band summarized in Figure 4.10. For Ordinary visible imaging mode colors are corrected using `processColours.py` and the result of processing is a colorimetrically correct photograph of the painting, depicted in Figure 4.11.



Figure 4.10: Result of Software extrapolated multispectral imaging mode processing - multispectral images in the range from UV to IR.



Figure 4.11: Result of Ordinary visible imaging mode processing.

Chapter 5

Conclusion

The goal of this thesis was to introduce usage of multispectral imaging (MSI) methods, which have an essential application in cultural heritage analysis and conservation, and to develop a workflow for imaging works of art, especially ANN enhanced multispectral imaging. We used a camera, which is modified to record spectral responses in multiple spectral bands ranging from 300 to 1000 nm while using only two special filters. It is thus an extremely time saving method for multispectral imaging instead of using particular band-pass filters. The main result presented in this thesis is our own processing software using enhanced by artificial intelligence.

In Chapter 2, we described the scene setup, depicted its 3D model and introduced each imaging mode. For all these modes, a detailed explanation of filters used is given. The two imaging modes involving artificial intelligence are Software extrapolated multispectral imaging mode and Ordinary visible imaging mode, which are further processed by our program using a neural network. We conclude that the Software extrapolated multispectral imaging mode has an advantage of being a much quicker method for multispectral imaging, but there are also disadvantages such as that filter B transmits UV and IR parts of spectrum at the same time and in relation with detection efficiency of the camera at about 300 and 950 nm the final estimation may be noisy and ambiguous.

In Chapter 3, the software processing of all imaging modes is introduced. Moreover, the performance in the two last mentioned imaging modes is analysed and the results of the cross-validation method for both of them are discussed. We found out, that the color processing worked well and that the colors were corrected successfully with the error corresponding to the just noticeable difference. For multispectral processing, the ANN training was successful as well, but we observed, that the as-

sumption of the ambiguity of the estimation was confirmed in spectral regions IR2 and IR3 and we, thus, conclude that the accuracy of the resulting images is limited.

In Chapter 4, a case study was performed, with each processing step commented and visualized.

Bibliography

- [1] Susana Del Pozo, Pablo Rodríguez-Gonzálvez, Luis Sánchez-Aparicio, A. Muñoz-Nieto, David Hernandez, Beatriz Felipe, and Diego González-Aguilera. Multispectral imaging in cultural heritage conservation. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W5:155–162, 08 2017.
- [2] Cerys Jones, Christina Duffy, Adam Gibson, and Melissa Terras. Understanding multispectral imaging of cultural heritage: Determining best practice in msi analysis of historical artefacts. *Journal of Cultural Heritage*, 45:339–350, 2020.
- [3] Antonino Cosentino. Multispectral imaging and the art expert. *Spectroscopy Europe*, 27, 01 2015.
- [4] Joanne Dyer, Giovanni Verri, and John Cupitt. *Multispectral Imaging in Reflectance and Photo-induced Luminescence modes: A User Manual*. 10 2013.
- [5] Anna Pelagotti, Andrea Del Mastio, Alessia De Rosa, and Alessandro Piva. Multispectral imaging of paintings. *Signal Processing Magazine, IEEE*, 25:27 – 36, 08 2008.
- [6] Rijksmuseum. The initial results of operation night watch. <https://www.rijksmuseum.nl/en/the-initial-results-of-operation-night-watch>, 2020.
- [7] Antonino Cosentino. Identification of pigments by multispectral imaging: A flowchart method. *Heritage Science*, 2:12, 04 2014.
- [8] Labsphere. Spectralon diffuse reflectance targets. https://www.labsphere.com/site/assets/files/1828/pb-13025rev02_spectralontargets.pdf, 2020. SRT-99-020.

- [9] Thorlabs. Beam trap. <https://www.thorlabs.com/drawings/6c53f5d6482ca528-E8DF2A6D-C526-70AE-378C186B869AD3AC/BT600-SpecSheet.pdf>, 2013. BT600.
- [10] Marcello Melis, Matteo Miccoli, and Donato Quarta. Multispectral hypercolorimetry and automatic guided pigment identification: some masterpieces case studies. *Proc SPIE*, 05 2013.
- [11] Thomas Mansencal, Michael Mauderer, Michael Parsons, Nick Shaw, Kevin Wheatley, Sean Cooper, Jean D. Vandenberg, Luke Canavan, Katherine Crowson, Ofek Lev, Katrin Leinweber, Shrirama Sharma, Troy James Sobotka, Matt Moritz, Chinmay Rane, Pavithra Eswaramoorthy, John Mertic, Ben Pearlstine, Manuel Leonhardt, Olli Niemitalo, Marek Szymanski, Maximilian Schambach, Sianyi Huang, Mike Wei, Nishant Joywardhan, Omar Wagih, Pawel Redman, Joseph Goldstone, and Stephen Hill. Colour 0.3.16. <https://doi.org/10.5281/zenodo.3757045>, January 2020.
- [12] R.W.G. Hunt and M. Pointer. Measuring colour: Fourth edition. *Measuring Colour: Fourth Edition*, 07 2011.
- [13] Ming Luo, Guihua Cui, and B. Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application*, 26:340 – 350, 10 2001.

List of Figures

2.1	3D model of the scene preparation.	6
2.2	Camera sensitivity of RGB pixels as given by Profilocolore [10].	7
2.3	Hyperspectral matching functions by Profilocolore [10].	9
2.4	Spectral transmissivities of all filters used.	12
3.1	Results of cross-validation holdout method for Software extrapolated multispectral imaging mode. ΔR values are depicted as functions of seven hypespectral bands.	18
3.2	The average values of the cross-validation for Software extrapolated multispectral imaging mode. ΔR value is depicted as a function of seven hypespectral bands.	19
3.3	CIE 1931 Standard Observer - Color matching functions obtained using Colour Science open-source Python package [11].	20
3.4	Results of cross-validation holdout method for Ordinary visible mode.	22
3.5	Average differences of colors before and after performed training for the Ordinary visible mode.	23
4.1	Initial processing diagram.	25
4.2	ANN enhanced modes diagram.	26
4.3	Exposure checking. Red color stands for overexposed pixels.	26
4.4	Raw data opened in Rawtherapee.	27
4.5	Marking the black hole, the white standard and the white cards.	28
4.6	Images before (1) and after (2) performance <code>processTiff.py</code> taken with filter A (a) and filter B (b) attached to a camera lens.	29

4.7	Images before (1) and after (2) performance <code>processTiff.py</code> taken with a combination of filters A and UV-IR cut attached to a camera lens.	29
4.8	Diagnosed patches in the photos containing colorchecker taken with filter A (a) and filter B (b) attached to a camera lens.	30
4.9	Diagnosed patches in the photo taken with filter A and UV-IR cut attached to a camera lens.	30
4.10	Result of Software extrapolated multispectral imaging mode processing - multispectral images in the range from UV to IR.	31
4.11	Result of Ordinary visible imaging mode processing.	32