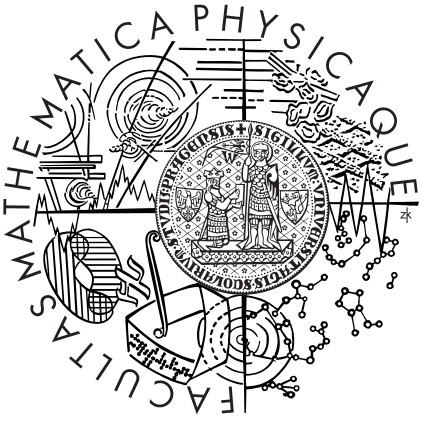
Charles University in Prague

Faculty of Mathematics and Physics

DIPLOMA THESIS



Bc. Jan Blažek

DIGITAL RESTORATION OF FRESCO PAINTINGS

Department of Theoretical Computer Science and Mathematical Logic

Supervisor: RNDr. Barbara Zitová, Ph.D.

Study program: I1 – Theoretical Informatics

2009

I would like to express a great gratitude to my supervisor RNDr. Barbara Zitová, PhD. and thank her for her professional feedback and help during my work. I would also like to thank RNDr. Janka Hradilová who has introduced the problem of fresco restoration to me.

I am much obliged to Institute of Archaeology of the Academy of Sciences of the Czech Republic and Academy of Fine Arts in Prague for providing us with all materials necessary for our work.

Last, but not least, I am very grateful to all my linguistic advisors for their help with the final stylistic polishing of this thesis.

I declare that I wrote this diploma thesis independently using exclusively sources cited herein. I agree with lending of this thesis.

Prague, 30th July 2009

Jan Blažek

Contents

Title: Digital restoration of fresco paintings
Author: Jan Blažek

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: RNDr. Barbara Zitová, Ph.D.,
Institute of Information Theory and Automation

Supervisor's e-mail address: zitova@utia.cas.cz

Abstract: In this thesis we present the application of digital image processing algorithms for the process of fresco restoration such as image registration, image fusion, and image segmentation. We have worked with images of various modalities (visual and ultraviolet spectra) and at different times. Moreover, during the image analysis we also have taken the local chemical analysis into account. The robustness of proposed algorithms is required to be high with respect to the bad state of the fresco. The achieved results provide a better insight into the evolution of the fresco aging to the art conservators and show the way how a proper conservation method can be chosen. All developed methods are illustrated by generated output images.

Keywords: virtual restoration, wall painting in the fresh plaster, fresco painting, image analysis, image fusion, image registration

Název práce: Digitální restaurování nástěnných maleb
Autor: Jan Blažek

Katedra (ústav): Katedra teoretické informatiky a matematické logiky

Vedoucí diplomové práce: RNDr. Barbara Zitová, Ph.D.,
Ústav teorie informace a automatizace AV ČR

E-mail vedoucího: zitova@utia.cas.cz

Abstrakt: Diplomová práce zpracovává nástěnné malby za použití algoritmů pro image processing. Součástí jsou algoritmy pro registraci, fúzi a segmentaci obrazu. Vstupními daty jsou snímky v různých modalitách (viditelné a ultrafialové spektrum) a staré snímky z předchozí doby. V úvahu byla vzata i možnost použít chemické vzorky získané z maleb. Kvalita vstupních dat vyžaduje robustní metodu, která zároveň nabízí lepší představu o vývoji fresky, pomáhá při jejím datování a poskytuje lepší podklady pro její restaurování. Výsledky všech použitých metod jsou ilustrovány na obrázcích.

Klíčová slova: virtual restoration, nástěnná malba do vlhké omítky, freska, analýza obrazu, image fusion, image registration

Chapter 1: Introduction

Computer technology is so advanced nowadays, that even such a complicated task as the image processing or the restoration of old paintings can be solved virtually on your monitor. This virtual restoring is a very useful and powerful tool especially in the field of culture heritage. Every real interference with the work of art is very risky and simulations can increase the quality of restorers' final product. In addition to great usability of such virtual processing, virtual restoration is also cheaper than other types of extraneous analysis.

Despite the obvious utility of such virtual restoration there is still no specialised software suited to that purpose, even though there is a very long list of image processing algorithms already available. That includes algorithms for image recognition and algorithms for processing of multimodal data as well as plenty of algorithms for improving visualization of noisy images especially from CT or MR medical devices. At first glance there seems to be no connection between ancient art and CT images, but on closer look we find many mutual issues such as image registration, contrast enhancing, segmentation, or texture synthesis. Of course there is a lot of differences as well, e.g. dissimilar erosion of images (lacunas and scratches on the paintings compared to the noise on CT images). So even though there are many specialised algorithms for image processing, they are not exactly what we need.

Figure 1.1: The church of st. George in Kostolany pod Tríbečom (Slovakia)

1.1 Motivation

As our first motivation we gained the images from a church in Kostolany pod Tríbečom in Slovakia (see Figure 1.1). The church of st. George is pre–Romanesque firstly mentioned in 1113 A.D. It is decorated with a very large set of wall paintings on the fresh plaster. Our approach to virtual restoring is based on those paintings and all examples mentioned in this thesis shows results obtained on Kostolany paintings. Presented methods were developed

in close cooperation with conservator scientists from ALMA[1], therefore we propose methods which facilitate work of conservators. The presented method would be useful also for another freco paintings.

1.2 Idea of restoration

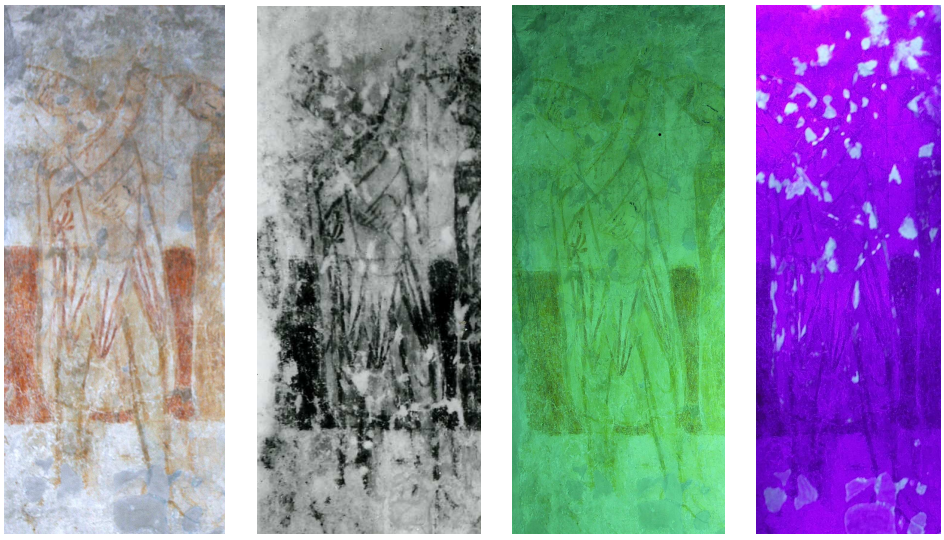
However, our goal is not that simple for sake of the restoration rules. In virtual restoration we can use a lot of techniques which are for restoration of real art work prohibited. The goal of restoration process is to present the original look of the piece of art. It ought to be as precise as possible, without any synthesized regions or putative information. All regions filled in with such a method, especially the holes in walls, must be distinguishable at first glance (different material, different colour). On the other hand the virtual image can be used to fill in some synthetic parts to the real work without any damage and can offer a "better" visualization of original state (before many years) to the laics.

At the beginning we have to understand how is our painting created. Hence we use the restoring techniques which give us necessary information. We obtained a lot of different types of input data of different characteristics. The simplest type of input data (already accessible) are the classic photos (wave–length 380–780nm, see Figure 1.2), which depict actual state of the art work. Besides that we use also very common photos in UV spectra (see Figure 1.3 (c),(d)) or different wave–length, which can penetrate through the first layer and can uncover images under surface. Our input images are very often affected by humidity or by a new layer of painting made over the old one. We say that those images have nonzero depth. Classic photos and photos in UV spectrum are often supplemented with photos in infra–red spectrum and point–based chemical samples from colour layers.

Figure 1.2: Photo of a wall painting on the fresh plaster in visible light spectrum

This thesis is based on the data collected by the Institute of Archaeology of the Academy of Sciences of the Czech Republic and the Academy of Fine Arts. The supplied data consists of the images in broad-band UV spectrum (UVB) see Figure 1.3 (c), narrow-band UV spectrum with maximum at $369nm$ (UVN) see Figure 1.3 (d), photos in visible spectrum see Figure 1.3 (a) and old photos from '1960s 1.3 (b).

[1]Academic Materials Research Laboratory for Painted Artworks – joined laboratory of the Academy of Fine Arts, in Prague, and The Institute of Inorganic Chemistry, The Academy of Sciences of the Czech Republic



(a) (b) (c) (d)

Figure 1.3: Original input data set. Photos of wall painting on the fresh plaster.
Image in visible spectrum (a), old photo from ‘1960s (b), wide–band UV spectrum (UVN) (c), narrow–band UV spectrum with maximum at $369nm$ (d). All can be seen different photographic angle and also different quality.

1.3 Environmental effects

The degradation process of wall paintings is inflicted by humidity and light or any mechanical damage. Under those influences the dyes fade away or change their original colours. The countours, primarily sharp, are blurred into different sides or disappear completely i.e.: with the march of time a dye can be found under the point where it was placed originally while the surface layer of the image can fade away to deeper layers. This process is very hard to simulate or estimate because it is also necessary to take many different influences into account. After a long degradation process the edges of the image vanish, which causes that different objects can be fused together and interpreted as a single one.

Better results can be achieved by developping a more proper physical model of diffusion including dyes chemical degradation caused by light and substances from the atmosphere. From this model we need to develop the inverse function, if it is possible. The inverse function can then enhance original contours but it can be difficult to obtain all parameters of the inverse function such as granularity of plaster or the effect of the water viscosity. Also mechanical damage, which is unpredictable and can affect paintings more than water diffusion or light effect, can be a serious problem.

Our thesis presents an approach without such degradation model. Therefore the results correspond mainly with the current quality of the piece of art and on quality of input data obtained from original wall paintings in the fresh plaster.

Chapter 2: Goals

The main purpose of this work is to propose a set of methods suitable for virtual restoration of old paintings and helpful in the analysis procedure. We want to enable more effective object identification in wall painting, to simplify orientation between different modalities of the same painting and show differences between layers in the sense of different modalities. This goal essentially contains the testing of known methods and their improvement as well as development of new ones, that should react better within this type of work.

From the view of great complexity of such goal we propose systematic consecution of digital image processing which contains image enhancement (noise reducing, contrast enhancement, edge sharpening), image fusion (especially multimodal data) and segmentation. For every part of this model we suggest methods that are suitable foremost for our type of input data.

Chapter 3: Methodology

Work with old paintings requires many different points of view, because the information embedded in the image can be split in three parts. First part includes visible information such as colours and textures; second part consists of information that is not visible (colours in UV spectra, chemical structure of colour layers) and the last part implies the information lost in time (holes or scratches). Our goal is to find dependencies among the data sets and to extract the invisible information of the currently visible state or to define state of painting before 1000 years.

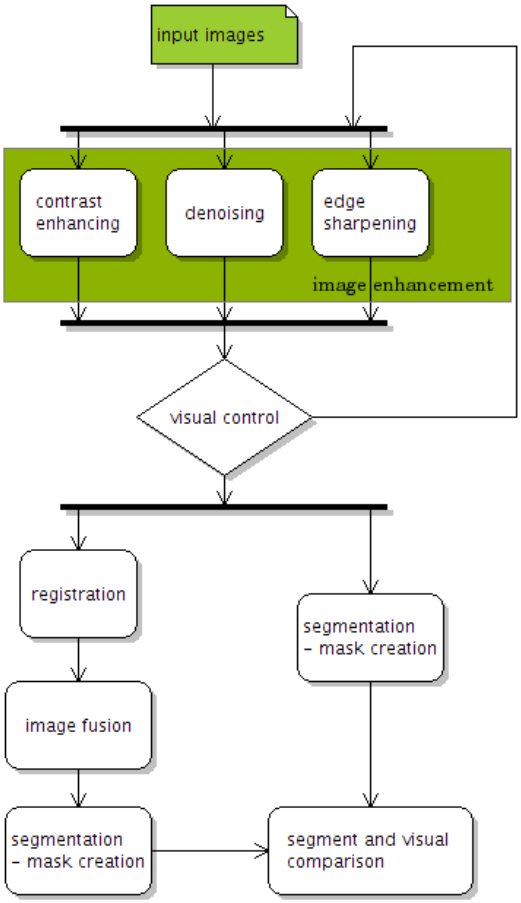


Figure 3.1: Image processing algorithm – UML diagram (unified model)

3.1 Process overview

Virtual restoration process covers many of known image processing algorithms, which should help us considerably reach our goal. Whole image correction consists of such algorithms ordered as follows. First we try to improve our input data (Chapter 4), which consists of using appropriate denoising filter (Section 4.1), edge detection and sharpening (Section 4.4) and contrast enhancement (Section 4.3). These algorithms form pre–processing phase, which has to be human–controlled. Following is image processing phase we have to register images (Chapter 5). Image registration is necessary for image fusion (Chapter 6) that comes next. Each layer of fused image is then segmented (Chapter 7). Segment masks in different modalities can be compared and output image is prepared for any of image inpainting methods. Results of those methods can be found in Chapter 9. Whole image processing diagram is shown in Flow chart 3.1. We recommend to substitute all segments without relevant in-

formation by the colour mean or synthesized by using the artificial texture for creation final virtual look of the art work.

3.2 Colour Spaces

When we use different colour spaces, we can extract information about edges and regions from the visible parts. This transformation from one three-dimensional space to another can simplify edge sharpening and contrast enhancement (greater part of colour range is used) as well as improving visual evaluation. For demonstration of this effect we can use a photograph of wall painting on fresh plaster shown in Figure 3.2. Differences between histograms are presented in Figure 3.3.

Figure 3.2: Example of wall painting on the fresh plaster

We can see on the hue graph more than one peak and the curve is more complicated than in other histograms. Peaks in this colour band come from combining peaks in red, green and blue bands. From the changes of derivation values we can mark out single segments and better quantize the values of coloured pixels. Such distribution is in harmony with human visual system and therefore useable without being disruptive.

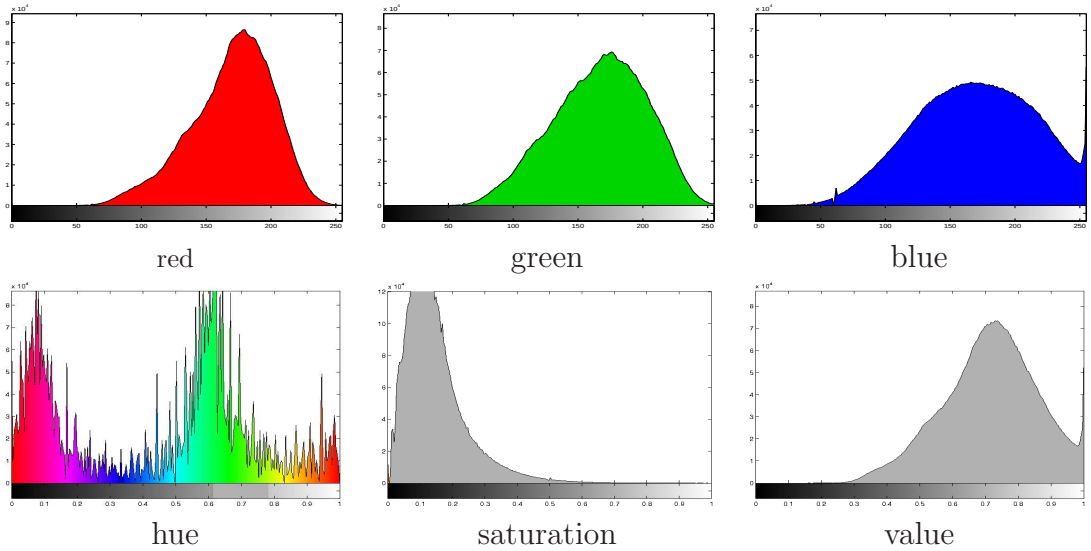


Figure 3.3: Comparsion of RGB histogram and HSV histogram
Histograms obtained from Figure 3.2. Concentration of peaks in hue histogram are created by mixing peaks in RGB

For our purpose we often employ HSV, HSL, Yuv, or CIElab colour space [1]. Broadly used are also images in gray spectra. Among these spectra we can simply convert our image using functions mentioned in Chapter 6 in [1].

Chapter 4: Image enhancement

First we concentrate on methods that can improve our input image before further processing as well as emphasise differences between measured values. All these methods scale measured values to possible maximum of defined range so that we can better recognise objects and also make the input smoother. Such operations do not need to show us any "real" image but can positively affect all other algorithms.

Ideally we use data that can be described by a continuous two-dimensional function, where the function range is the pixel colour. Unfortunately, reality is different. Images are affected by omnipresent noise and truncation error and are completely discrete due to splitting into pixels. Our goal is to reduce noise by smoothing the image (Section 4.1). Then we can use an interpolation function for subpixel points (see Section 4.2). This process guarantees that the image function behaves as continuous in spite of being based on discrete values. Finally we equalize histogram of the image to increase contrast of input data (Section 4.3).

4.1 Denoising

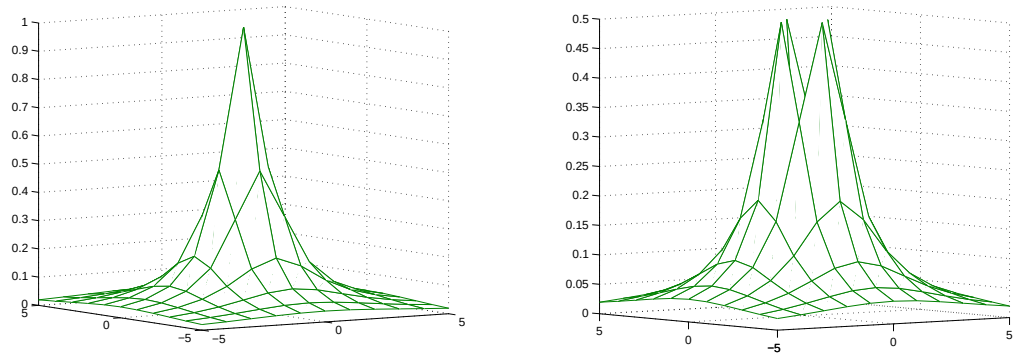
Denoising is standard image processing method which clears off high–frequency information with or without respect to edges. Most common and used approximations of noise are white Gaussian noise and salt and pepper noise. For our input data we assume white Gaussian noise approximation, which means gauss function added to image function f. Image obtained after digitalization process is then f':

$$f' = f + n, \tag{4.1}$$

where $n = N(0, \sigma^2)$. For reducing noise we can use many different filters (see [1]). Modern denoising methods are using the decorrelation properties of wavelet decomposition [2] or pleasible beavior of dictionary-based approaches [3]. The result depends mainly on the quality of images. In our case we can expect that the purchase of high quality images is possible, because the creation of fair input data is a part of the conservation process. Using this assumption we can use arbitrary and a very simple filter with variable size and weights.

The formula of denoising filter is as follows:

$$M[x, y] = \frac{1}{(x^2 + y^2 + 1)}, \tag{4.2}$$



(a) (b)

Figure 4.1: The shape of smoothing function M

Second version (b) of smoothing function M is independent on the de–noised pixel - the weight of point $[0, 0]$ is set to 0.

where $x, y \in \{-R, \ldots, -1, 0, 1, \ldots, +R\}$ and $M[x, y]$ represents weight of pixel value at a distance $\sqrt{x^2 + y^2}$ from the denoised pixel. Graphically shown in Figure 4.1. After convolution of pixel neighbourhood with the weight matrix M we have to normalize obtained value by $\sum M[x, y]$.

In some cases we need a filter independent on the original pixel value and so we use the same formula as in equation 4.2 but with $M[0, 0] = 0$. Surface of this function is shown in Figure 4.1(b).

Described pre–processing should improve efficiency of algorithms sensitive to noise i.e. algorithms which mostly run only on local range (region growing methods in segmentation). We choose radius size for the smoothing filter with respect to actual noise in input data.

4.2 Subpixel Interpolation

To set the subpixel values we use simple bilinear interpolation and each band is counted separately. For bilinear interpolation we need four neighbouring pixels

$$\begin{aligned} P_{1,1} &= (x_1, y_1) = colour[\lfloor x \rfloor, \lfloor y \rfloor], \\ P_{1,2} &= (x_1, y_2) = colour[\lfloor x \rfloor, \lceil y \rceil], \\ P_{2,1} &= (x_2, y_1) = colour[\lceil x \rceil, \lfloor y \rfloor], \\ P_{2,2} &= (x_2, y_2) = colour[\lceil x \rceil, \lceil y \rceil]. \end{aligned} \tag{4.3}$$

where $\lfloor a \rfloor$ is a value of the floor function of a and $\lceil a \rceil$ is a value of the ceiling function.

There we substitute $\alpha = \frac{x - x_1}{x_2 - x_1}$ and $\beta = \frac{y - y_1}{y_2 - y_1}$. Colour of the subpixel can then be counted as follows:

$$colour[x, y] = \alpha\beta P_{1,1} + (1 - \alpha)\beta P_{2,1} + \alpha(1 - \beta)P_{1,2} + (1 - \alpha)(1 - \beta)P_{2,2} \tag{4.4}$$

Multidimensional colours are substituted in every single band.

4.3 Contrast Enhancing

As we can see in Figure 1.3 our images are very fuzzy and blurred and have very low contrast. This defect can be improved by enhancing contrast and sharpening edges (see Figure 4.4). We choose algorithm for adaptive histogram equalization [4], specifically its improved version with saturation and desaturation effect [5]. We enhance contrast in HSV spectrum (as in [5]) where contrast is represented by the value of V component.

The principle of these algorithms is histogram stretching [1]. We distribute the input image into square–shaped fragments of uniform size with a centroid. For each of these fragments we then compute a histogram and we stretch it. Value in each pixel is received as bilinear interpolation of the values from stretched histograms of neighbouring centroids.

For example we have pixel $A = [x, y]$ and its neighboring centroids C_1, C_2, C_3, C_4. We look at their stretched histograms $hist_{C_1}$, $hist_{C_2}$, $hist_{C_3}$, $hist_{C_4}$ for the colour value of A. We sum up these values as in equation 4.4, where P_i are the transformed values of our pixel A by the histograms $hist_{C_i}$, α and β are relative distances from the centroids (with respect to the distance of the centroids from each other).

The effect of adaptive histogram enhancing method depends mostly on the size of fragments and on given borders for histogram stretching, which both affect the ratio of contrast enhancing. The histogram is stretched in such a way that the first colour with non–zero value in the histogram (i.e. gray level 58) is transformed to zero colour (to gray level 0) and the last colour with non–zero value is changed to maximum colour value (for one byte colour is set to 255). If we use concrete borders for this stretching, then the first colour is set to the lower border and the last to the higher border. Colour values between these new borders are distributed by their original distance ratio in the new range (which means mostly uniform distribution). Effect of such stretching can be seen on histograms in Figure 4.2.

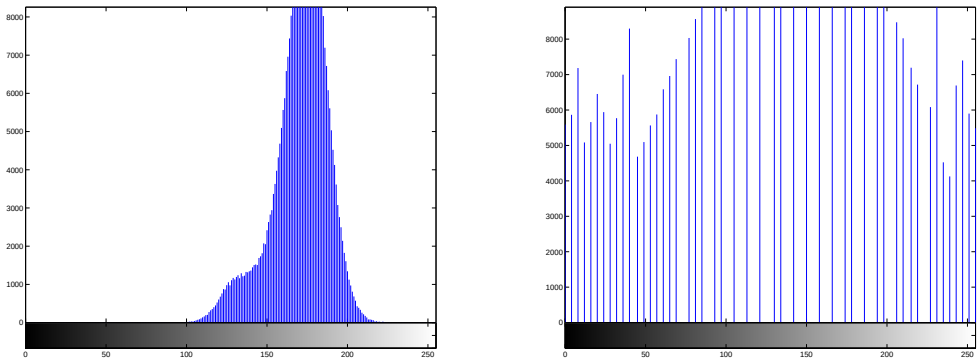
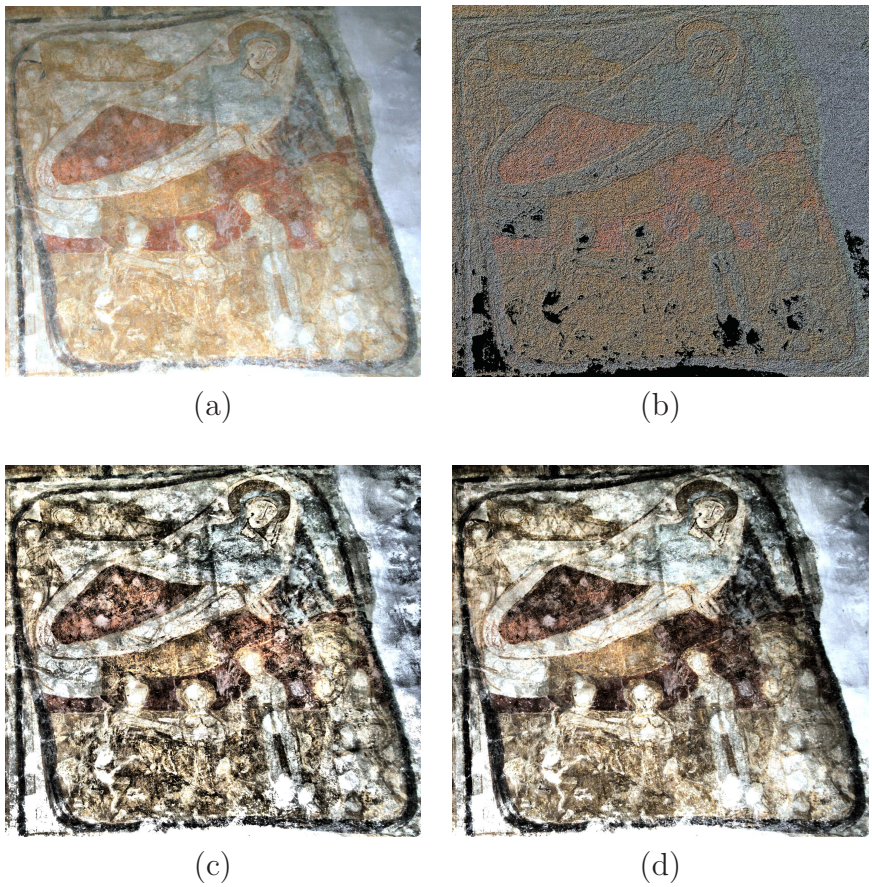


Figure 4.2: Comparsion of original and equalized histogram

Method presented in [5] combines histogram equalization with saturation and desatu-

ration process, which should improve colouring of the image. Process affect only u and v components in uvY colour space, where u and v specify chromaticity and Y stands for brightness. Histogram equalization is then applied on Y component. Principle of saturation and desaturation is enhancing pixel chromaticity, which have some colour value (i.e. those that are not near a white point). These colours are saturated - two of RGB colours are maximized proportionally in ratio of the original pixel and after saturation are desaturated allow the center of gravity by the law of the colour mixing. Details can be found in [6].

Influence of these improving algorithms is shown in Figure 4.3.



(a) (b)

(c) (d)

Figure 4.3: Original and enhanced images. The three enhanced images were made using adaptive histogram equalization algorithm.
Used ranges were 2 (b), 150 (c) and 800(d) respectively.

4.4 Edge Sharpening

Edge detection methods are essentially based on changes in image gradient. In case of colour images we can specify representation of the gradient. It is possible for example to examine

gradient for each spectrum of used colour space and improve each spectrum separately or to examine the "global" gradient measure. These algorithms can lead to different gradient measures, so we have to choose the best one experimentally.

4.4.1 Sobel operator

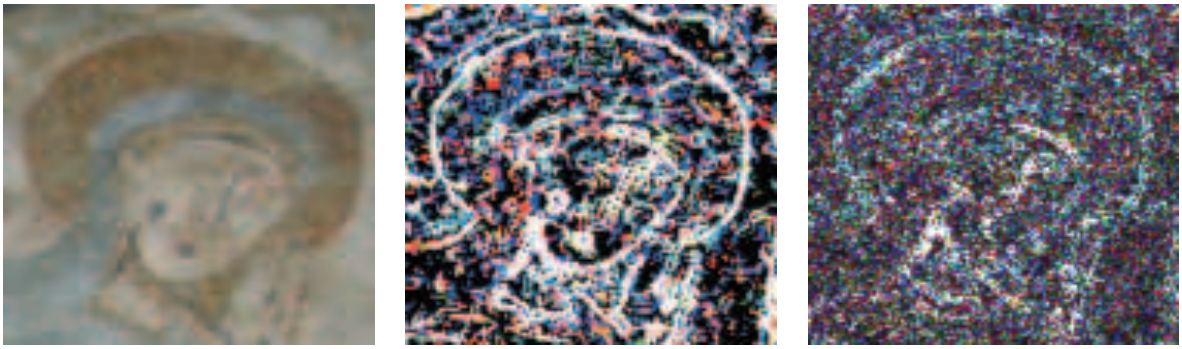
Matrix operators, particularly Sobel's present one of the simplest edge detection methods. Sobel's operator is based on two weighting matrices which increase high differences and decrease low differences between neighboring pixels. Each matrix searches for edges in different direction;

$$S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}.$$

For both matrices we compute the convolution with the original image

$$G = \Omega * S. \tag{4.5}$$

Finally from G_x and G_y we receive the value of $G = \sqrt{G_x^2 + G_y^2}$. The interpretation of this value is then the image containing all the found edges or a matrix for image improvement. The operator G is a discrete representation of the first derivative splitted into two matrices - each matrix represents the partial derivative.



(a) (b) (c)

Figure 4.4: Edge detectors
Original image (a), scaled Sobel operator result (b) and Marr-Hildreth filter (c) with size 5×5

4.4.2 Laplacian

A edge detection method using Laplacian was firstly mentioned in [7] and is often known as Marr–Hildreth algorithm. The second derivative of a function is represented by the Laplacian

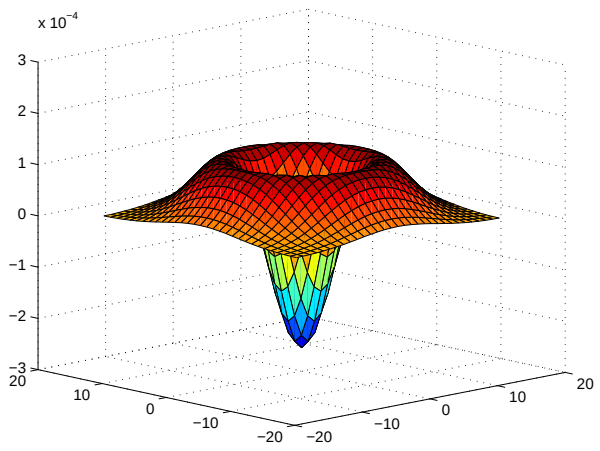


Figure 4.5: Graph of the Laplacian of the Gaussian operator

(or the laplace operator in discrete digital image space). The Laplacian of a function f is defined as:

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f = \sum_{i=1}^{n} \frac{\delta^2 f(x)}{\delta x_i^2}, \tag{4.6}$$

which for two dimensional image function gives:

$$\Delta f(x,y) = \frac{\delta^2 f(x,y)}{\delta x^2} + \frac{\delta^2 f(x,y)}{\delta y^2}. \tag{4.7}$$

If we presume Gaussian smoothing of the image we need the Laplacian of Gaussian operator denoted as LoG, which expresses second derivation of a gaussian 2D function:

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{4.8}$$

LoG can be simply calculated as a sum of the partial derivatives:

$$LoG(x,y) = \frac{\delta^2 G(x,y)}{\delta x^2} + \frac{\delta^2 G(x,y)}{\delta y^2} = G(x,y) \cdot \left(\frac{x^2-\sigma^2}{\sigma^4} \right) \cdot \left(\frac{y^2-\sigma^2}{\sigma^4} \right). \tag{4.9}$$

We can see this in Figure 4.5. Discrete form of this continuous function can be a convolution matrix $n \times n$. Frequently used are 3×3 and 5×5 filters:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.10}$$

Chapter 5: Registration

As mentioned earlier we use different types of data (chemical samples of colour layers, photos of different light spectra) for virtual restoration. Data are acquired in misaligned two-dimensional meshes of coordinates (except for special cases when tripod or another position fixing method is used). These meshes have different scales and can be deformed towards each other. If we want to use universal coordinate system for images obtained from different camera position we have to register them.

There are two approaches to such registration [8]: local and global. Let's assume, for the sake of simplicity, that the paintings are completely flat, making the second type of registration better suited to our purpose, because we can transform each point using the same transformation parameters. Local registration could still be used for corners of buildings, but such data input can be split into two flat images.

5.1 Perspective transformation

With flat images it is a very good idea to use perspective transformation. Our images are flat and obtained from different camera positions. The perspective transformation of two-dimensional data has following universal formula:

$$u_i = \frac{ax_i + by_i + c}{gx_i + hy_i + 1}, \qquad v_i = \frac{dx_i + ey_i + f}{gx_i + hy_i + 1}. \tag{5.1}$$

For this reason we need to have four referential points $[x_1, y_1]$, $[x_2, y_2]$, $[x_3, y_3]$, $[x_4, y_4]$ in referential image and four target points in aligned image $[u_1, v_1]$, $[u_2, v_2]$, $[u_3, v_3]$, $[u_4, v_4]$ from which we can enumerate values a, b, c, d, e, f, g and h from the formula (5.1). The equation system obtained from (5.1) can be expressed also in matrix notation:

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -y_1u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1v_1 & -y_1v_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -y_2u_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2v_2 & -y_2v_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3u_3 & -y_3u_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3v_3 & -y_3v_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4u_4 & -y_4u_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4v_4 & -y_4v_4 \end{pmatrix}^{-1} \times \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{pmatrix}. \tag{5.2}$$

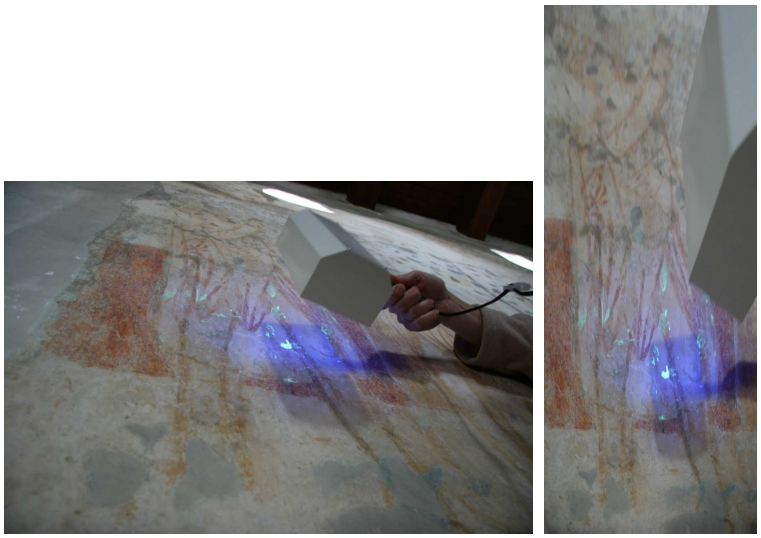


Figure 5.1: Example of image transformed by perspective mapping
First image is the original from the camera and the second one is transformed by perspective mapping. Result of the transformation can be improved when using DPI higher than in this example. In the upper part of transformed image we can recognise smoothing due to the lack of information in input image.

Difficulty of perspective transformation lies in vanish–points. A vanish point gives zero in denominator $gx_i + hy_i + 1 = 0$, making the values in the area incorrect due to discretization and limit in infinity. For manual setting it is highly recommended to place referential points as near as possible to image borders. This set up ensures that the vanish points occur outside of the image.

On the other hand, the transformation is quite powerful. Results shown in Figure 5.1 are very impressive. One disadvantage of presented registration method is the manual setting of referential points. We have not found a method sufficient to register multimodal data automatically especially old images from ‘1960s. There is a possible way mentioned in [9], however it is very hard to use it for eight parametres of perspective transformation.

5.2 Alignment of referential points

As we mentioned above we set the referential points manually. For good registration output we have to select suitable points as precisely as possible. With manual selection it is likely that these points are misplaced. To eliminate this possibility we use alignment algorithm based on covariation measure or on mutual information measure within small vicinity area of misplaced pixel. Such a measure can not be used on the whole image without previous registration. We assume that the transformation has just a small impact on closest vicinity. To achieve good results of aligning method this assumption must be hold.

5.2.1 Covariance measure

Covariance shows the dependency between images. We presume that the differences between mean value and pixel values are corresponding in both images.

For each point at a distance of radius R from the target point we calculate covariance between points lying in the neighborhood of this pixel and points situated in the vicinity of referential pixel. We use the best result (highest value) instead of misplaced target pixel. Covariance is defined as:

$$cov\,(A,B) = E((X - EX)(Y - EY)). \tag{5.3}$$

where X represents the pixel values in vicinity $\{[-R,-R],\ldots,[+R,+R]\}$ around new target pixel A and Y represents pixel values in the same range around referential pixel B center. Both, X and Y, are random values. E represents the expected value and it can be calculated as follows:

$$E(X) = \frac{1}{(2R+1)^2}\sum_{i=-R}^{R}\sum_{j=-R}^{R} colour[A_x + i, A_y + j], \tag{5.4}$$

$$E(Y) = \frac{1}{(2R+1)^2}\sum_{i=-R}^{R}\sum_{j=-R}^{R} colour[B_x + i, B_y + j]. \tag{5.5}$$

These values are calculated separately for each colour band and then added together (with equal weights). Also often used similar measure is correlation which is also usable and where is covariance normalized by standard deviation:

$$cor(A,B) = \frac{cov(A,B)}{\sigma_x \sigma_y} \tag{5.6}$$

5.2.2 Mutual information measure

In case of multimodal data when the covariance assumption (differences between the mean value and the pixel values correspond in both images) is not valid, it is better to use mutual information as a measure, because it is not derived from the real values but from entropy of the region, which should correspond in both images without any restricting presumption. With such kind of data it is likely, that it does not come from the same object (independent). This method was published at [9] and for alignment of misplaced referential points used in [10].

We use the infomation entropy, conditional entropy and joint entropy defined by 5.7, 5.8 and 5.9 respectively. It is easy to see that two random variables are independent if $H(Y,X) = H(Y) + H(X)$.

$$H(Y) = -\sum_{y_i \in \Omega_Y} p(Y = y_i) * \log[p(Y = y_i)], \tag{5.7}$$

$$H(Y|X) = -\sum_{y_i \in \Omega_Y} p(Y = y_i|X) * \log[p(Y = y_i|X)], \tag{5.8}$$

$$H(Y, X) = -\sum_{y_i \in \Omega_Y} p(Y = y_i, X) * \log[p(Y = y_i, X)]. \tag{5.9}$$

We define mutual information as:

$$I(X, Y) = H(X) + H(Y) - H(Y, X). \tag{5.10}$$

According to this formula we want to maximize $I(X, Y)$; that means maximizing entropy of the image Y and minimizing joint entropy of Y and X, with $H(X)$ constant for our purpose. Now we concern a small vicinity of the referring point and we calculate in this vicinity mutual information to target point $I_{i,j}(Y, X)$ for each pixel, where X and Y do not represent whole image, but a small vicinity of our actual pixel $[i, j]$ only. Then we choose the pixel with $I_{i,j}(X, Y)$ maximal.

A discrete version of $I_{i,j}(X, Y)$ would be:

$$p(Y = I) \widehat{=} p(Y = I|Y = colour[x, y], [x, y] \in A), \tag{5.11}$$

$$H(A) = -\sum_{I=0}^{\#colours} p(Y = I) \cdot \log[p(Y = I)], \tag{5.12}$$

$$H(A, B) = -\sum_{I=0}^{\#colours} \sum_{J=0}^{\#colours} p(Y = I, X = J) \cdot \log[p(Y = I, X = J)], \tag{5.13}$$

$$I_{i,j}(X, Y) = H(X) + H(Y) - H(X, Y). \tag{5.14}$$

In order to maximize $I(X, Y)$, we focus only on the term $H(Y) - H(X, Y)$. The presumption for using of the mutual information (and covariance also) as a measure is that the difference after transformation on vicinity of referring pixel would be insignificant. If this assumption is not valid we must split transformation to small steps where the assumption valid is.

Now we finish pre–processing phase of image enhancing. In the next few chapters we propose methods for the image fusion and uncovering connections between different layers of wall painting. For the connection we use segment masks, which can be compared and fused images, which contains the most important information from single images.

Chapter 6: Image fusion

As it was mentioned above as an input we obtain different images. We want to extract from these images all the useful information which is important for conservator analysis. From n input images we want to have only one image with all valuable information. The most important is to decide what is the valuable information. Let us have a closer look on following two methods: the principal component analysis (PCA) and the fusion based on wavelets.

6.1 Principal Component Analysis

This method project the data from the input space into a less dimensional space with by the user defined dimensionality. In this process is the most interesting information (the dimension with the greatest variability) stored priorly. The original algorithm is based on the covariance matrix which can be for the image of 2000x1000 pixels too complex (as a dimension we take a colour band). Therefore we use the Oja's iterative algorithm mentioned in [11]. We want to project the input images to the three major components (we compose a RGB image) which characterize the input data with a minimal error. We have a set of pixels (colours) which have to be normalized. To normalize the given set of pixels we compute in each band separately:

$$\overline{X[b]} = \frac{1}{|X|} \sum_{i=0}^{|X|} x_i[b], \tag{6.1}$$

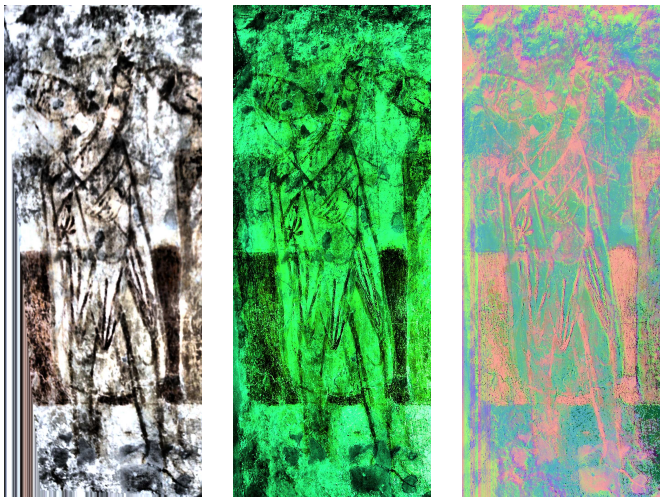
where $x_i[b]$ represents colour of the pixel x_i in colour band b i.e.: $b = 3$ represents the value of the pixel in blue spectra of the first image and the variance of our set:

$$var(X)[b] = \sum_{i=0}^{|X|} \left(x_i[b] - \overline{X[b]} \right)^2. \tag{6.2}$$

Values of each pixel are normalized according with the next formula:

$$\widehat{x_i[b]} = \frac{x_i[b] - \overline{X[b]}}{\sqrt{var(X)[b]}}. \tag{6.3}$$

After normalization we can use the Oja's iterative algorithm which counts in each step the value of the principal component $\overrightarrow{w}$. In the first iteration we initialize this component by a



(a) (b) (c)

Figure 6.1: Join based on PCA

The first two images show registered images in visible light spectrum (a) and ultra–violet spectrum (b). The image (c) shows result of PCA analysis.

normalized random value and in every next step we actualize value as follows:

$$\overrightarrow{w}_{k+1} = \overrightarrow{w}_k + \gamma_k \phi(\widehat{x}_i - \phi \overrightarrow{w}_k), \tag{6.4}$$

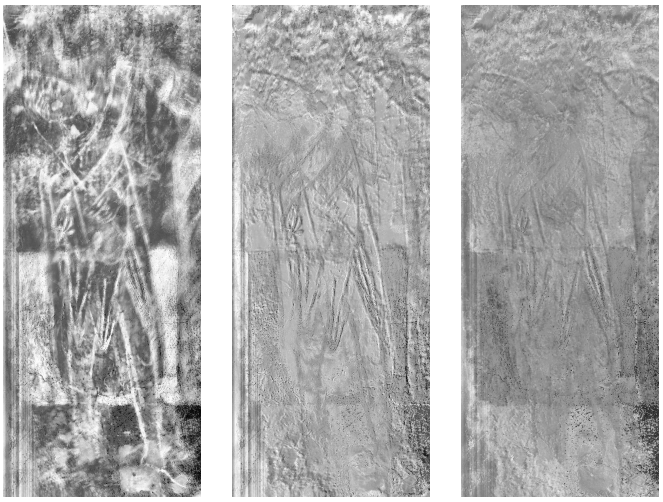
where $\phi = \widehat{x} \cdot \overrightarrow{w_k}$ and γ is a learning factor. This step guarantees normalization of vector $\overrightarrow{w}$ and converges to its local optima if is valid:

$$\sum_{k=0}^{\infty} \gamma_k = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \gamma_k^2 = C.$$

To reach the global optima is necessary to use an extra learning method such as simulated annealing. The details can be seen in [11].

Assumed that Oja's algorithm converges, it is necessary to project the n–dimensional space generated by input images to $n-1$-dimensional which is orthogonal to our vector $\overrightarrow{w}$. Then we can compute the next component for projected $\widehat{x}_i$.

The recursive use of this methods generates up to n orthogonal vectors with descending importance. We can simply use the first three components which we interpret as RGB or HSV colours. These colours are synthetic such as interpretation of the first component in grayscale. The output image give us only the information about the position of interesting objects. One example of the transformation can be seen in Figure 6.1 and each of principal components in a grayscale in Figure 6.2



(a) (b) (c)

Figure 6.2: Three most important principal component of 6.1 (c)

6.2 Fusion Based on Wavelets

Another way of image fusion is the combination of corresponding frequency bands from different images see Figure 6.3. A technique of frequency decomposition can be found in [12] especially the image fusion based on wavelets in [13]. However, our results (see Figure 6.4) was not sufficient for our purpose and therefore we present just one example.

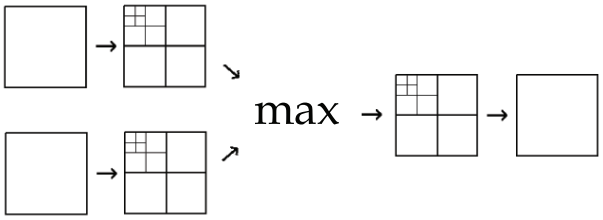
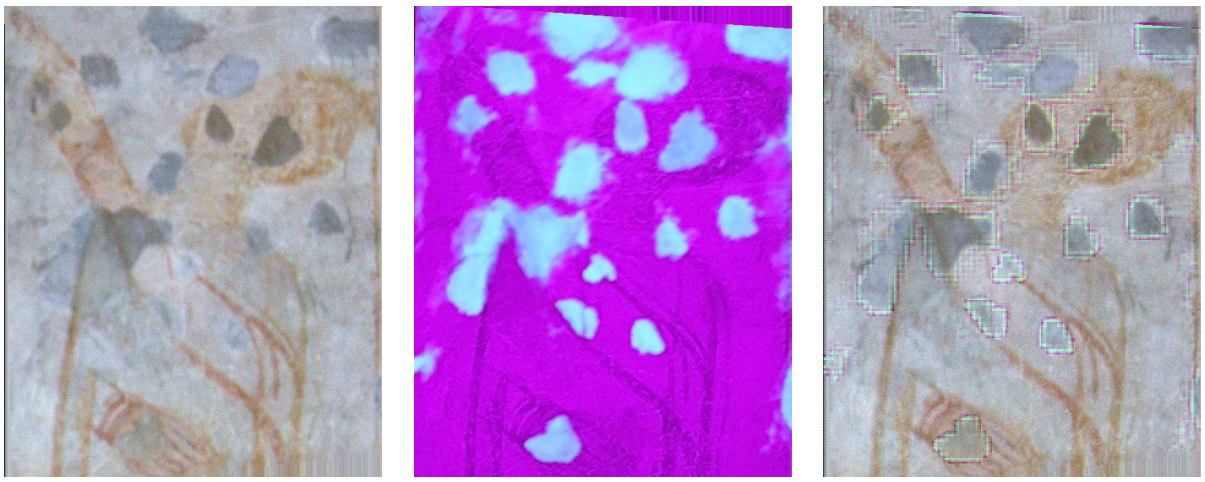


Figure 6.3: The diagram for image fusion algorithm



(a) (b) (c)

Figure 6.4: Fusion based on wavelets.
The first image is in visible light spectrum second in UV light spectrum. On the third image we can see that additive information or combination is insufficient; in addition to that there are residua of wavelet function.

Chapter 7: Segmentation

The last part of our work consists of the segmentation which constitutes splitting of the real image Ω in to regions R_i containing pixels of similar characteristics. In those regions hold two conditions:

$$\bigcup_{i=0}^{n} R_i = \Omega \qquad \text{and} \qquad R_i \bigcap R_j = \varnothing \text{ for } i \neq j,$$

however, the splitting criteria can be very variable. For our purposes it is very advisable to obtain segments which had originally the same colour or texture. We also want to mark regions where the information has been lost (scratches, lacunas). As well as the comparison of segment masks in different layers of image is desirable to a great extent. It means exactly to compare the segment masks in the images in visible spectrum and in UV spectrum.

There are three main ways how to segment images. A global look represents the K–mean algorithm which compare the information at every point in the whole image. And the top–down and the bottom–up approach, where the segments are splitted in accord with their inconsistency or where the segments with local similarity among neighours are joined together, respectively.

7.1 K-mean Colour Quantization

The presented simplest segmentation method is the K-mean algorithm. At first is necessary to define requested number of colours K in output image. These colours are then selected by minimizing of the squared error (7.1). We have $\overline{C}_i$ colours where $i = 1, \ldots, K$ and we want to minimize

$$\arg\min_{C} \sum_{i=0}^{K} \sum_{x_j \in C_i} \|(x_j - \overline{C}_i\|^2, \tag{7.1}$$

where x_j is a pixel value and $x_j \in C_i$ means that the nearest colour mean for x_j is C_i. This minimization is even more complicated when we use three colour bands i.e. RGB. The first possibility is $\overline{C}_i$ three dimensional colour and term (7.1) minimized for all the three colour bands at once. In this case we obtain $\overline{C}_i$ without chromaticity close to any level of gray colour especially for images with more than one colour. The other way, to split quantisation for each band, is more sensitive to chromaticity, but we take a risk of splitting the image into too many segments. More precisely there are $K \times K \times K$ different segments (we obtain for each band K independent segments which). For the minimisation of term (7.1) we use

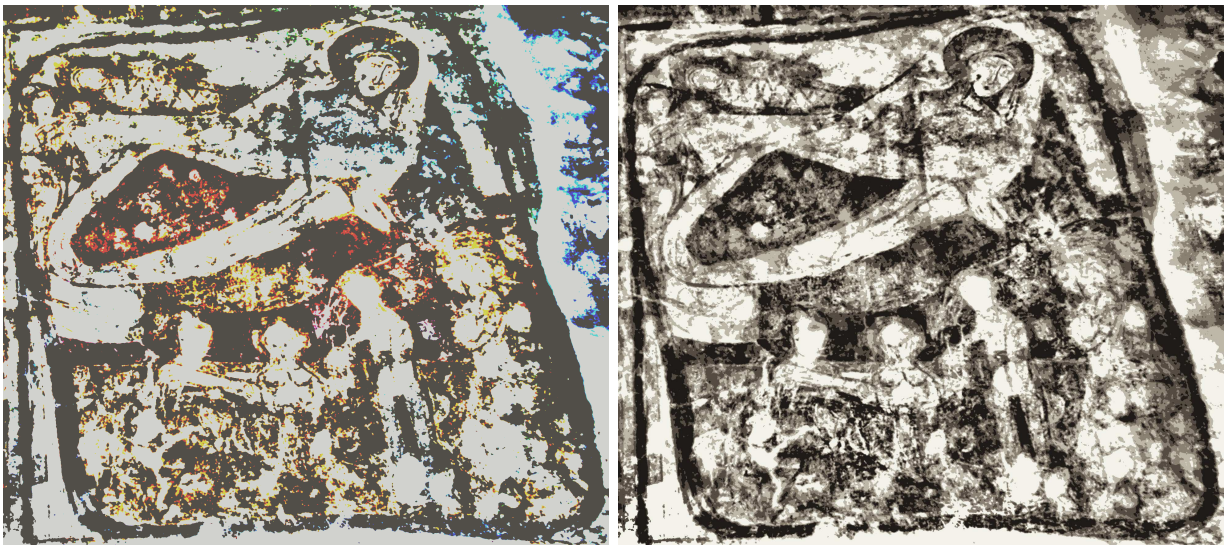


Figure 7.1: Effect of colour quantization

The first image has 3 different means in each colour band. There are three red, three green and three blue means (120000 segments), so we can see more colours and less details. The second image has 5 means over the whole colour space, we can see more details and no colour (155000 segments). The original image has equalized histogram and for a notion of size contains about 3.5 megapixels.

iterative algorithm. At first we randomly set the value of $\overline{C}_i = color[x, y]$ where x, y are set randomly. In each loop of this iterative algorithm we act using the next scheme:

1. Select randomly the pixel P from the image.
2. Find N nearest colour means $\overline{C}_i$ to P and denote their order with o_i
3. Update the values of the colour means by the formula:

$$\overline{C}_{i,k+1} = \overline{C}_{i,k} + \frac{1}{o_i \cdot w_i}(P - \overline{C}_{i,k}) \tag{7.2}$$

As was said above, we can use different means for each colour or universal for all colour bands together. The effect of both possibilities shows Figure 7.1

Another variant of the K-mean algorithm is add to the colour means also coordinates of pixels. Then the algorithm keeps more continuous segments with this information.

7.2 Ohlander Price Reddy Segmentator

An original method was published in [14] by Ohlander, Price and Reddy (OPR). It is based on segment splitting; at the beginning there is only one segment – the whole image, which

we split into segments by using any peak in its histogram. The splitting algorithm works in a loop:

1. Choose any non–processed segment.
2. Create colour histogram of this segment.
3. Find all peaks in histogram.
4. Count left and right threshold of each peak and choose the best peak in the histogram.
5. Split segment to smaller segments by counted thresholds.

As we show in Figure 3.3 it is very important here in which colour space we use the histogram. Each operation as smoothing 4.1 or the histogram equalization 4.3 have changed our histogram. In general, the contrast equalization adds a lot of peaks into the histogram therefore we obtain oversegmented image. On the contrary, the smoothing reduces number of peaks. So the small peaks vanish in spite of their crucial sense for deeper hierarchical splitting. Therefore the utilization of these enhancing methods before OPR Segmentator is not easy and needs some experience.

The peak selection is based on the multispectral searching. We prepare several histograms – in red, green and blue spectra and also in hue, saturation and value spectra. We can add also another colour histogram as uvY and thus improve the effect of the peak selection. At first we smooth the histograms in order to reduce noise effect. A peak candidate we define as a point in the histogram where the previous and the next value are smaller than the peak candidate value in the histogram and it is valid:

1. The difference of the histogram values between a peak candidate and its nearest local minima is sufficient
2. The ratio of a peak candidate value and its nearest local minima value is sufficient
3. The distance between the nearest peak candidates is sufficient

A particular setting of the parametres is discused in [14].

We have changed the part of the best peak selection from peak candidate set. The original method uses priority table mentioned in [14]. We substitute this table with entropy reduction measure. From the set of the peak candidates we choose that one which reduce entropy of processed the segment more than other. The entropy is counted over the splitted histogram of the segment. This histogram has two parts – first one around the peak value and second

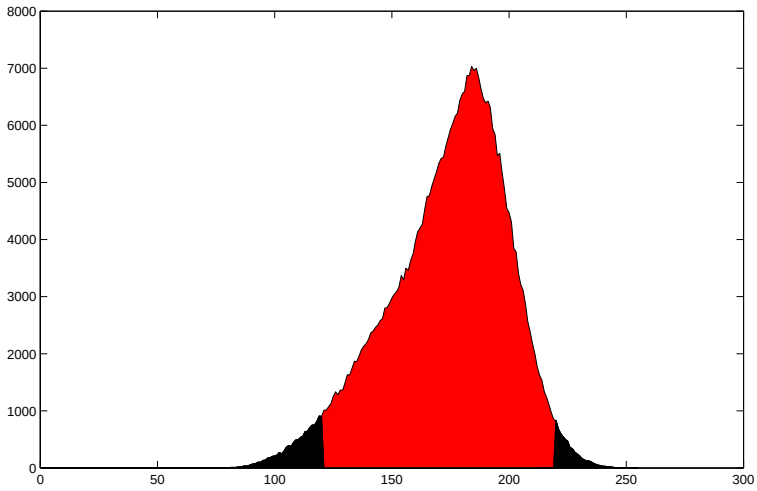


Figure 7.2: Splitted histogram by peak value and borders on left and right

one as a background. The entropy we count as in equation 5.12, a splitted histogram we can see in Figure 7.2

Finally we have to show how the borders of the best peak are selected. The first condition is the local minima, the other conditions are similar to the peak selection:

1. The difference between the histogram value of local minima and the next local maxima is sufficient
2. The ratio between the local minima value and the nearest local maxima value is sufficient
3. The distance between the nearest peak candidate and our cadidate is sufficient

We show the effect of our method in Figure 7.3

7.3 Region growing methods

The last group of segmentation methods contains algorithms based on joining segments. At the beginning we create some primitive segments like simple pixels or nodes in a quadtree and according to their local similarity we join them together. Different variants are based on the difference of the primitive segments and various local measures for segment joining. We propose a simple method which begins with primitive segments i.e. created by the K-mean colour quantization 7.1. For these primitive segments we count similarity of neighbours. The

Figure 7.3: The original and the segmented image with using Ohlander Price Reddy algorithm
On the segmented image can be seen a problem of segmentation with the parts with different brightness. The segmented image consists of 107000 segments

proposed measure for the similarity is based on the colour mean difference, the variance of colours in segments, the size of segments and on the size ratio of joined segments.

The mean colour in each segment is counted as an average in each band by equation 6.1. The mean difference we count simply as:

$$A = |\overline{X} - \overline{Y}|. \tag{7.3}$$

The variance is counted by equation 6.2. The aim is to minimize variance in the segments. We want to minimize variance after joining. Clear counting of variance for each pair of candidates is very time consuming. Therefore we reduce this expensive enumeration and instead of the variance $var(X)$ we compare standard deviation σ_X and colour mean distance A. The standard deviation is given by:

$$\sigma_X = \sqrt{var(X)}. \tag{7.4}$$

Now we can define a "range" of the colours in the segment X as follows:

$$R_X = \langle \downarrow R_X, \uparrow R_X \rangle = \langle \overline{X} - \sigma_X, \overline{X} + \sigma_X \rangle. \tag{7.5}$$

We can prefere joining of such segments where $\overline{Y} \in R_X$ or $\overline{X} \in R_Y$ is true. We particularly use the distance from the nearest border as a measure. To put it more precisely:

$$B = \begin{cases} \overline{A} - \downarrow R_B & if \quad |\overline{A} - \downarrow R_B| < |\uparrow R_B - \overline{A}|, \\ \uparrow R_B - \overline{A} & else. \end{cases} \tag{7.6}$$

(a) (b)

Figure 7.4: The effect of the region growing method
At (a) is presegmented image by the 10-mean colour quantization (36227 segments) at (b) the region growing method is used (20000 segments).

There we take the better result from $\overline{X}$ and $\overline{Y}$.

We also want to join the small segments together as well as join the small segments to the big segments. This requirement we interpret as a measure of the sum of segment sizes and ratio of segments sizes:

$$C = |X| + |Y| \quad \text{and} \quad D = \max\left(\frac{|X|}{|Y|}, \frac{|Y|}{|X|}\right).$$

All these measures $A, \ldots, D$ we join in one, where the results in $A, \ldots, D$ represent votes for the best candidate. We compare all the neighbouring segments and take the best result according with used combined measure.

Effect of segment reduction is on Figure 7.4.

7.4 Morphology

After a pure segmentation are segments often jagged and aliased. Those segments need not to be in the harmony with real borders in the image. For reduction of small segments on the borders and for border smoothing we can use morphology algorithms. Two basic algorithms is the segment opening and the segment closing. Both of this methods are composed from the segment dilatation and the segment erosion in different order.

The erosion of the segment S by the structuring element E is defined by:

$$S \ominus E = \{z \in S | E_z \subseteq S\}, \tag{7.7}$$

where E_z is the translation of E by the vector $\vec{z}$, i.e. $E_z = \{e + z | e \in E\}, \forall z \in S$.

Figure 7.5: The effect of segment closure

For our purpose we take the structuring element as circle and from the segment cut border of width of the radius of our circle.

The dilation of S by the structuring element E is defined by:

$$S \oplus E = \{e + z | e \in E, z \in S\}. \tag{7.8}$$

As in the erosion we take circle as the structuring element and add border of width of the radius of our circle.

By a combination of these two operators we obtain the segment closure as:

$$A \bullet B = (A \oplus B) \ominus B, \tag{7.9}$$

and the segment opening as:

$$A \circ B = (A \ominus B) \oplus B. \tag{7.10}$$

Effect of segment closing can be seen in Figure 7.5

Chapter 8: Alternative approach

As an alternative approach which has appeared during our work is the study of degradation processes. The wall painting is for a long time under the effect of the atmosphere, the acids created in the plaster or under the effect of a mechanical damage. Based on contemporary images and a good degradation model we can reconstruct the probable look of art work many years ago. A complex model of the degradation process is not easy to create due to the unpredictable effects like changes in humidity, possible previous restoration or the information lost (a hole in plaster etc.). We have to simplify our model but on the other hand, model have to reflect the most of the important influences. There we present a modelation of the water diffusion effect.

8.1 Diffusion model

A good idea for the image enhancing is to find the degradation function which through the ages blur the wall painting. We have image function $I(x, y, \tau, p)$ where x and y are the coordinates of pixel, τ represents for how long the diffusion affected the image and p represents type of pigment; for each pigment can have different characteristics. Then we obtain a set of $M \times N \times p$ equations for every pixel:

$$I(x, y, \tau, p) = \sum_{i=0}^{M \times N} p_{i,p}(x, y, \tau) \cdot I(x_i, y_i, 0, p), \tag{8.1}$$

where $p_{i,p}(x, y, \tau)$ represents a distribution function and $\sum$ represents additive joining of colours. This distribution function we can try to compose from the gaussian diffusion and a shifting vector which represents gravitation or another preferred direction of humidity proliferation.

At first we suppose that the granularity of the plaster is constant for each point of the image and therefore the diffusion is homogenous gaussian. For gaussian distribution we obtain equation:

$$p_{i,p}(x, y, \tau) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i - x)^2 + (y_i - y)^2}{2\sigma^2}} \tag{8.2}$$

where $[x_i, y_i]$ is position of our pixel in the image and variance σ^2 depends on τ and on the type of pigment p particularly $\sigma = f(\tau, p)$. This equation is valid for both, continuous image and discrete digital image difference, only in domain of i.

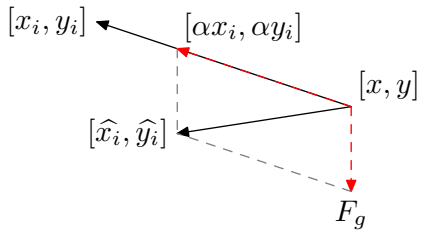


Figure 8.1: Weight distribution between shift function and diffusion function

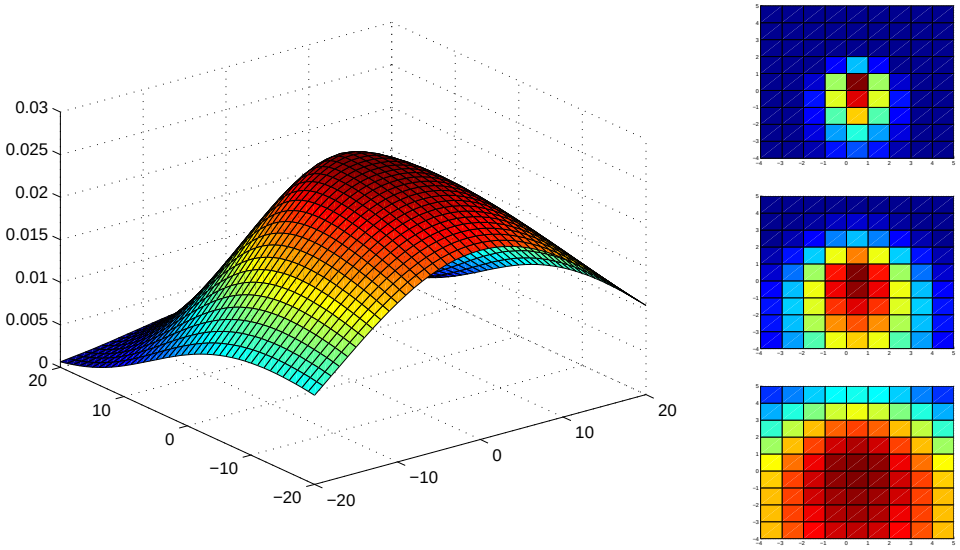


Figure 8.2: Diffusion in discrete model

The original point of pigment has been set at $[0, 0]$ these images shows diffusion effect in pixels with variation 1, 2 and 5. Effect of gravitation is 55%. First graph shows surface of diffusion function but with wider range and higher variation.

The implementation of the shifting vector can be used as follows. We suppose that distance of pigment from the original position is given by the type of pigment p and by time of degradation τ. Then the length of the vector $[x_i - x, y_i - y]$ used in $p_{i,p}(x, y, \tau)$ we divide to the shifting vector and the original direction by the ratio of water viscosity in plaster and gravitation. This ratio will be represented by the coefficient $\alpha \in \langle 0, 1 \rangle$. Finally, we obtain:

$$\widehat{p_{i,p}}(x, y, \tau) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\widehat{x_i}-x)^2+(\widehat{y_i}-y)^2}{2\sigma^2}}, \tag{8.3}$$

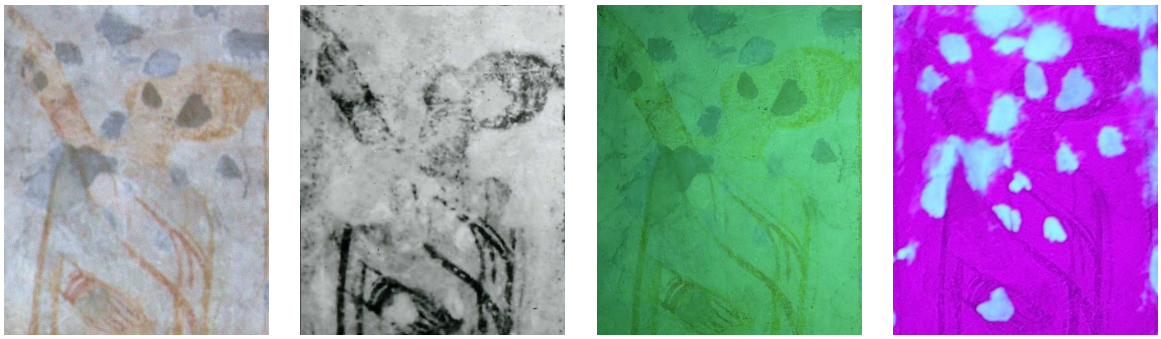
where $[\widehat{x_i}, \widehat{y_i}] = (1 - \alpha)[x, y - F[y]] + \alpha[x_i, y_i]$. The effect of the vector splitting can be seen in Figure 8.1. An example is shown in Figure 8.2.

In spite of the first look our model is not as robust as we desire. The parametres which we include are the preferred direction F, the period of degradation τ (with unspecified transformation function to variance σ^2) and the effect of a preferred direction α. On the other hand,

we suppose that there is no colour change and that the process of diffusion is homogeneous. Both of these presumptions are not realistic but they are necessary for the sake of simplification. The colour change is very common phenomenon that we cannot solve without a special knowledge about dyes. The diffusion also can be homogenous only when special conditions, like constant humidity in the atmosphere hold. If these conditions are not guaranteed, all of our parametres can differ in times and strongly affect the diffusion function.

Chapter 9: Results

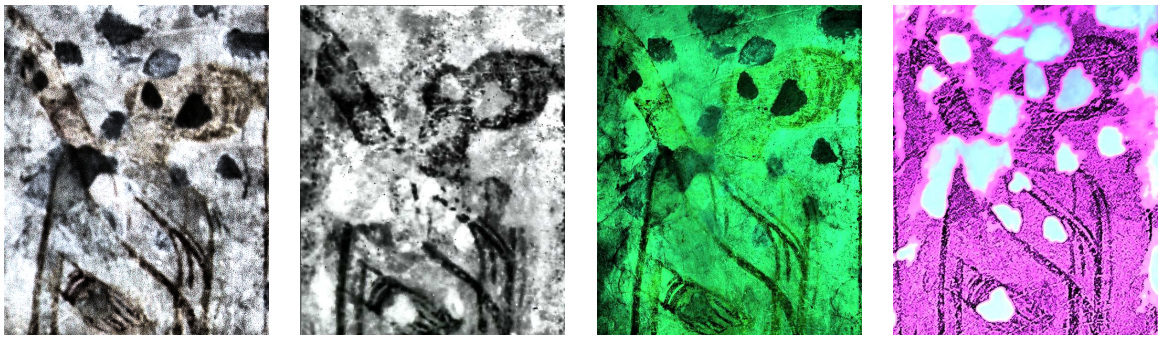
In this chapter we present the results of our approach to the virtual restoration. Each image contains the description of used algorithms and order of images keeps chronology of used algorithms. All presented result was made on original images (see Figure 9.1)



(a) (b) (c) (d)

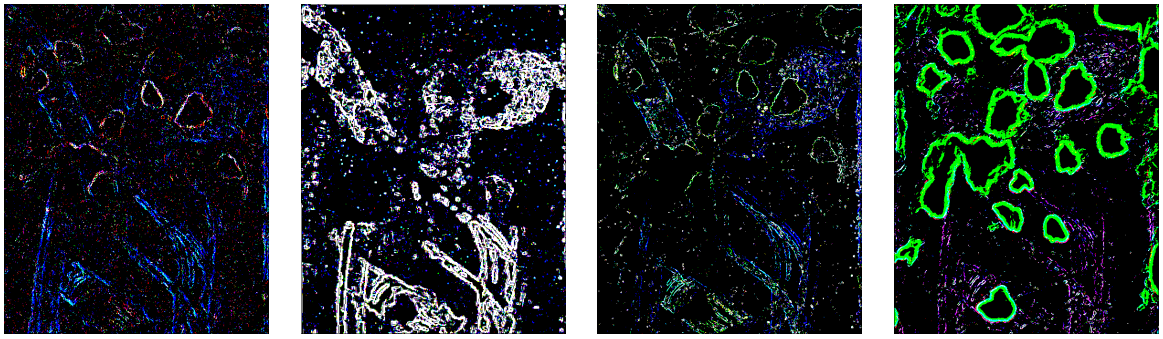
Figure 9.1: Original input images

From the left: (a) photography in visible spectrum, (b) old black and white photography from ‘1960s, (c) broad–band ultra–violet spectrum (UVB), (d) narrow–band ultra–violet spectrum (UVN) with maxima at 368 nm



(a) (b) (c) (d)

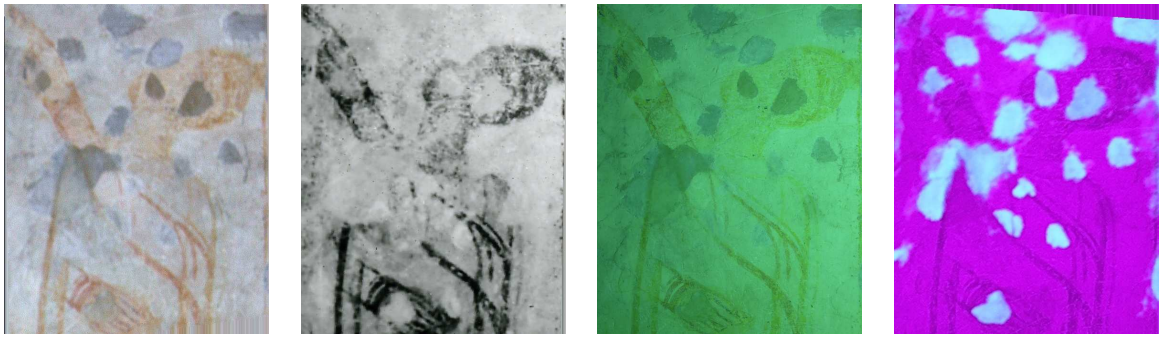
Figure 9.2: Contrast enhancement of input data with region size of 200×200 pixels.



(a) (b) (c) (d)

Figure 9.3: Edge detection in detail (head) of input data

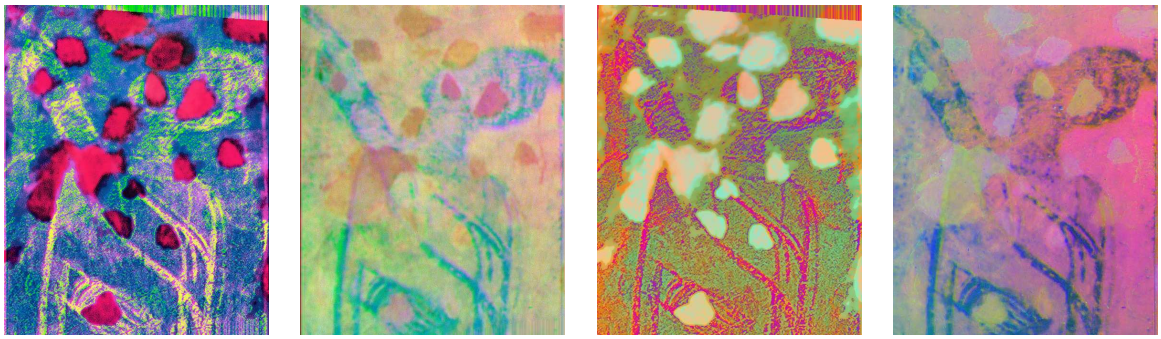
Smooth was necessary only for visible spectra (a) with radius of filter $R = 1$. At (d) are very good visible lacunas in the painting



(a) (b) (c) (d)

Figure 9.4: Registered images

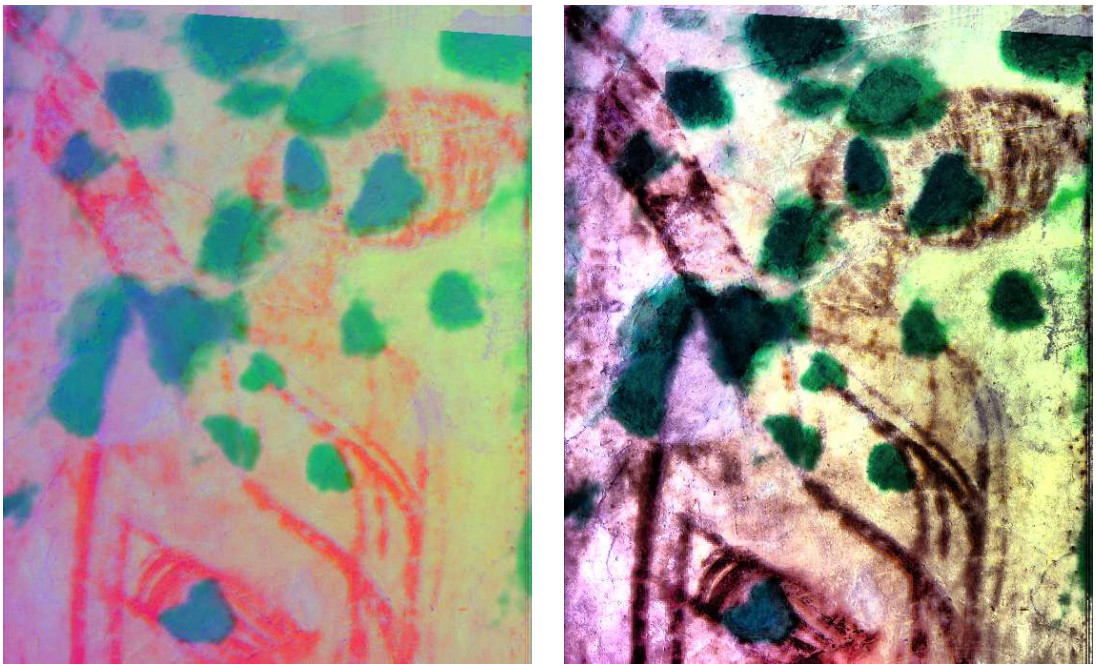
As a reference image for registration was taken image (c). In other images see corners for difference.



(a) (b) (c) (d)

Figure 9.5: Fused images

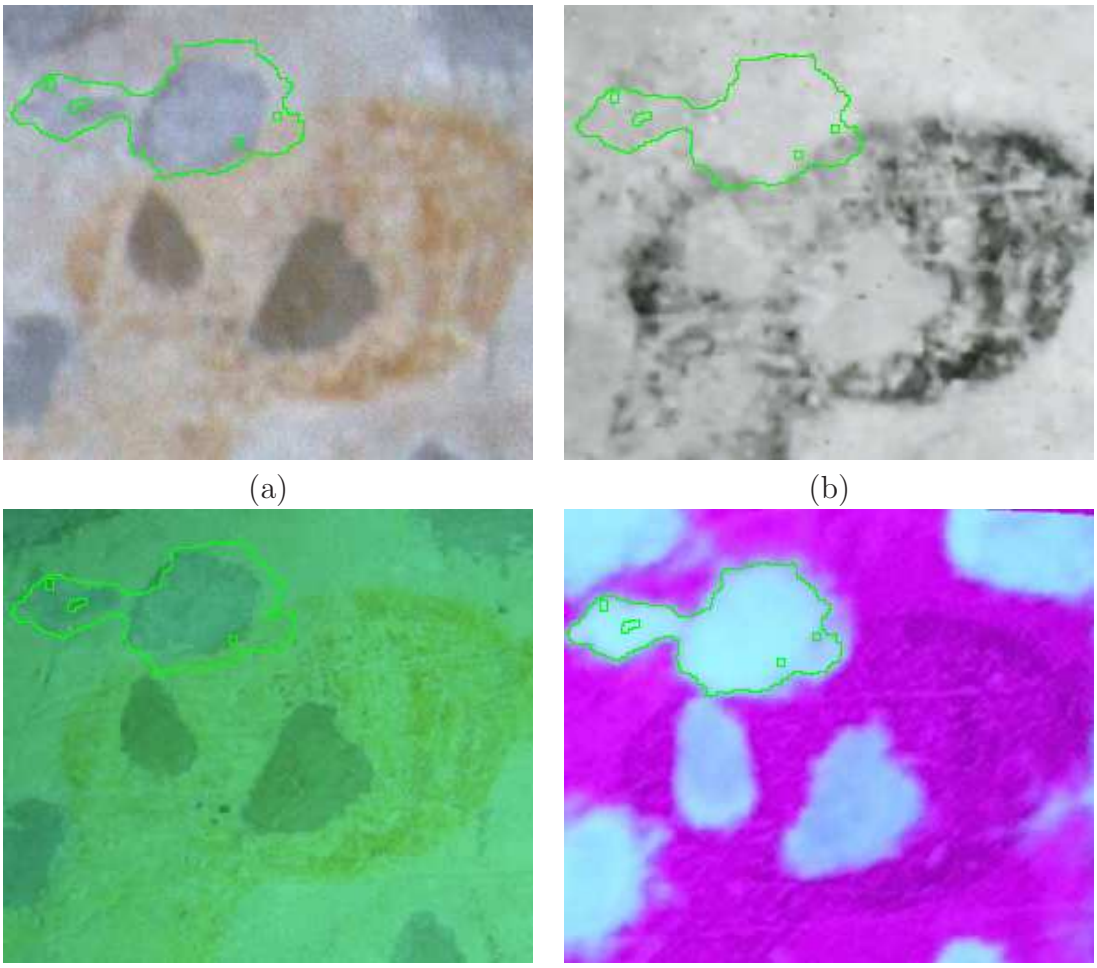
At image (a) are fused images from visible and UVN spectra. This image shows well the edges and the lacunas, but all information hidden in lacunas is lost. (b) contains visible spectra and old photo and shows well the information in lacunas if there is any. (c) is created by fused UVN and visible spectra. The image has high contrast and very good visible contours. And finally (d) contains old photo and UVN image. There we can see details as a collar and a wristlet of person.



(a) (b)

Figure 9.6: The fusion of all input images. (b) shows enhanced verison

We can see lacunas and a collar of the person from UV spectrum and shape from visible spectrum. Some details are visible in lacunas and also on improved look of wrist.



(a) (b)

(c) (d)

Figure 9.7: Mask comparsion

Emphasized region represents segment obtained from image 9.6(a). Segment mask was applied on all input images. There we can recoginse information added from UVN to other modalities.

Chapter 10: Conclusion

Our work presents an approach to the virtual restoration. Our exploration in restoring shows useful algorithms from image processing which can improve the quality and uncover possible hidden parts of the art work. During our work we fight with unspecified parts of our work which was uncovered subsequently and often change the direction of our work. Similarly the conservators uncover the possibilities of the virtual processing. The possibility of a simulation of the degradation process appear a short time before the thesis finalization and also input data were continuously filled up. For that reason there remain a great motivation for the future work.

As our contribution we consider

- Creation of the software for virtual restoration.
- The improvement of Ohlander Price Reddy algorithm.
- Special measure for region growing method.
- New approach to study of degradation processes.

10.1 Future work

Our future work have two possible directions. The first direction continues with a deeper study of degradation processes and their simulation. The second direction improves the virtual restoration and image synthesize.

The degradation processes simulation should improve the work of conservators including the study of the art work and its layers. The possible ways are a study of water diffusion in plaster and diffusion analysis of dyes. Dyes also can decompose (change the chemical structure) and then without a deeper study can be the input image useless due to unpredictable chromaticity changes.

The synthetic images which we want to compose from the input images need corrections or scratches and lacuna filling. Such process requires algorithms for texture synthesis and better segmentation algorithms.

Bibliography

[1] Rafael C. Gonzalez and Richard Eugene Woods. *Digital image processing*. Prentice Hall, 3rd edition, 2007. 954 pages.

[2] D. Donoho, I. Johnstone, and Iain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455, 1993.

[3] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transaction on Image Processing*, 15(12):3736–3745, December 2006.

[4] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart Ter Haar Romeny, and John B. Zimmerman. Adaptive histogram equalization and its variations. *Comput. Vision Graph. Image Process.*, pages 355–368, 1987.

[5] Soo-Chang Pei, Yi-Chong Zeng, and Ching-Hua Chang. Virtual restoration of ancient chinese paintings using color contrast enhancement and lacuna texture synthesis. In *Image Processing, IEEE Transactions*, volume 13, pages 416 – 429, March 2004.

[6] L. Lucchese, S. K. Mitra, and J. Mukherjee. A new algorithm based on saturation and desaturation in the xy chromaticity diagram for enhancement and re–rendition of color images. In *Proc. Int. Conf. Image Processing (ICIP 2001)*, volume 2, pages 1077–1080, September 2001.

[7] D. Marr and E. Hildreth. Theory of edge detection. In *Proceedings of the Royal Society of London. Series B*, pages 187–217, February 1980.

[8] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21:977 – 1000, October 2003.

[9] Paul A. Viola. Alignment by maximization of mutual information. Technical Report 1548, Massachusetts Institute of Technology, June 1995.

[10] Barbara Zitová, Filip Šroubek, and Jan Flusser. An application of image processing in the medieval mosaic conservation. *Pattern Analysis and Applications*, 7:18–25, February 2004.

[11] E. Oja. A simplified neuron model as a principal components analyzer. *Journal of Mathematical Biology*, 15:267 – 273, 1982.

[12] Ingrid Daubechies. *Ten lectures on wavelets*. SIAM, 3rd edition, 1992. 357 pages.

[13] Hui Li, B. S. Manjunath, and Sanjit K. Mitra. Multisensor image fusion using the wavelet transform. *CVGIP: Graphical Model and Image Processing*, 57(3):235–245, 1995.

[14] Ron Ohlander, Keith Price, and Raj Reddy. Picture segmentation using a recursive region splitting method. In *Computer Graphics and Image Processing*, pages 313 – 333, 1977.

Chapter A: Appendix

Content of enclosed CD:

- Text of this thesis in *pdf* and *ps* format
- Figures used in this thesis in encapusulated post–script format (in the directory *figures*)
- Fresco software (in the directory *runnable*)
- Source code of the software (in the directory *src*)
- User documentation (in the directory *doc*)