BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF MATHEMATICS

ÚSTAV MATEMATIKY

TESTING IMAGES CREATION PROGRAM WITH DEFINED SUB-PIXEL SHIFT AND RESIZE PRECISION

PROGRAM PRO GENEROVÁNÍ TESTOVACÍCH OBRAZŮ S DEFINOVANOU SUB-PIXELOVOU PŘESNOSTÍ POSUNU A ZMĚNY MĚŘÍTKA

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR AUTOR PRÁCE	**Eliška Málková**
SUPERVISOR VEDOUCÍ PRÁCE	**Mgr. Jana Hoderová, Ph.D.**

BRNO 2017

Zadání bakalářské práce

Ústav:	Ústav matematiky
Studentka:	**Eliška Málková**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	**Mgr. Jana Hoderová, Ph.D.**
Akademický rok:	2016/17

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Program pro generování testovacích obrazů s definovanou sub-pixelovou přesností posunu a změny měřítka

Stručná charakteristika problematiky úkolu:

Cílem bakalářské práce je vytvořit testovací obrazy umožňující optimalizaci parametrů algoritmu fázové korelace pro sub-pixelovou registraci obrazů.

Cíle bakalářské práce:

Vytvořit program pro generování obrazů se sub-pixelovým posuvem a změnou měřítka za použití konvoluce s proměnným jádrem.

Seznam doporučené literatury:

GOSHTASBY, Ardeshir. Image registration: principles, tools and methods. London: Springer, c2012. Advances in computer vision and pattern recognition. ISBN 144712457X.

DRUCKMÜLLEROVÁ, Hana. Phase correlation: the mathematical background and application to image registration. Saarbrücken, Germany: Lambert Academic Publishing, 2011. ISBN 3846550914.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt
Tato práce se zabývá generování testovacích sub-pixelově posunutých obrazů se známým posuvem. Ty umožní najít vhodné parametry váhové funkce, díky nimž se zpřesní hledání posuvu dvou obrazů. Dále je využito konvoluce s proměnným jádrem ke generování obrazů se změněným měřítkem. K definici konvoluce pro obraz využíváme Fourierovu transformaci. Práce zahrnuje všechnu potřebnou teorii a je přiložen program, který slouží ke generování posunutých obrazů a obrazů se změněným měřítkem.

Abstract
This thesis is devoted to the creation of testing images with known sub-pixel shifts. Their purpose is to find suitable parameters of the weight function in order to refine the finding of the two image shift. Furthermore the variable convolution kernel is used to generate scale-changed images. We use the Fourier transform to define convolution on image. All the necessary theory is summarized in this thesis and the program, which was used to create shifted and scale-changed images, is also attached.

klíčová slova
Fourierova transformace, konvoluce, sub-pixelová přesnost, posun, změna měřítka, proměnné konvoluční jádro

keywords
Fourier transform, convolution, sub-pixel precision, shift, scale-change, variable convolution kernel

Málková, E.: *Program pro generování testovacích obrazů s definovanou sub-pixelovou přesností posunu a změny měřítka.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 26 s. Vedoucí bakalářské práce Mgr. Jana Hoderová, Ph.D.

I declare that I have written the bachelor's thesis *Testing images creation program with defined sub-pixel shift and resize precision* on my own according to advice of my bachelor's thesis supervisor Mgr. Jana Hoderová, Ph.D.,and using the sources listed in references.

April 18,2017 Eliška Málková

I would like to express gratitude to my bachelor's thesis supervisor Mgr. Jana Hoderová Ph.D. for provided guidance and advice during my work on the thesis. I would also like to thank to Prof. RNDr. Miloslav Druckmüller, CSc. for his unlimited patience with my programming skills.

Eliška Málková

Contents

Chapter 1

Introduction

The image registration is widely used process for example in the observation of cell structures and other medical specimens. It is also used for observing space objects such as the Sun, for example to predict the movement of sunspots. This movement is predictable however very hard to observe in details because we are not able to register images with a satisfactory precision.

Image registration is generally considered to be a process of finding mutual geometrical transformation between two images. The methods available for image registration depend on whether the images are only shifted or distorted in more complicated way. A rotated or scaled image is also considered as shifted because both of them can be transformed into a shift. For the registration of shifted images, we can reach even sub-pixel precision. However the precision of the sub-pixel shifts computing still needs to be improved as it is not sufficient enough nowadays.

For this improvement we need to create adequately large set of images with defined sub-pixel shifts and defined scale-changes, which is the task of this thesis.

In order to generate shifted images we use weighted average in the program, because its speed is equal to the usage of convolution kernel method but the code is more simple.

The variable convolution kernel is used to create scale-changed images. This method can make the program to be slower when generating images larger than 256×256 pixels because it needs to be computed for each of the pixels separately. However that is not our case as images of this size are sufficient enough for our purpose.

A primary source for definitions of basic terms in Chapters 2-4 is [1].

Chapter 2

Digital image

First of all we need to define the term "digital image" in order to work with them. Digital image is a two dimensional discrete representation of real-world scene. It pictures a momentary event from the three-dimensional spatial world which is created by a digital camera. The digital image like this contains additive noise which changes an image we wanted to capture. Aim of the digital image processing is to get rid of it among others. The noise is usually caused by increased temperature of the camera sensor or some dust on the lens. For the purpose of further image processing done in this thesis, we assume that the image does not contain additive noise.

Definition 2.1. (Digital gray-scale image) Let $R = \{0, 1, ..., N-1\}^2$, $N \in \mathbb{N}$ and let $W = \{0, 1, ..., w-1\}$, $w \in \mathbb{N}$. Function

$$f(x, y) : R \longrightarrow W$$

is called a *digital gray-scale image* where N is called the *image width* and the *image height*. Elements of R are called *pixels* and value of f in pixel (x, y) is called the *pixel value*. The value of w determines the image *dynamic range*. The dynamic range is *n bits per pixel* (it is an *n-bit image*) if $w = 2^n$.

An image is usually defined to be rectangular, but for the purpose of this thesis a square image is sufficient enough. We use only cropped part of the image which needs to be square so the phase correlation can work properly.

We usually use an object called *image matrix* to represent the image. However there is no need to use operation defined to matrices for image matrix, because it is just a table of pixel values in coordinates (x, y). Every operation applied to image matrix is meant to be applied on each pixel separately.

Definition 2.2. (Digital color image [1]) A *digital color image* is a triple of digital gray-scale images (r, g, b) which are called the *red*, *green* and *blue color channels*.

A digital color image is converted into the gray scale for purposes of image registration. To do that we need to compute a convex combination of the red, green and blue color channels

$$f(x, y) = \text{Round}(c_r r(x, y) + c_g g(x, y) + c_b b(x, y)),$$

where $c_r, c_g, c_b \in \langle 0, 1 \rangle$ and $c_r + c_g + c_b = 1$. The constants c_r, c_g, c_b should be chosen appropriately so they can minimize the standard deviation of additive noise in image f.

There is no prescription for choosing constants which could be applied on all images universally. For general images (taken without any color filters) we use assessment around

$$c_r = \frac{1}{9}, c_g = \frac{6}{9}, c_b = \frac{2}{9}.$$

Definition 2.3. (Additive noise) Let f be a digital gray-scale image representing an ideal image (containing no additive noise), let n be a digital gray-scale image of the same size as f, whose pixel values are rounded independent realization of random variable X, which usually has normal distribution. Let

$$h(x,y) = \begin{cases} f(x,y) + n(x,y) & \text{if } 0 \leq f(x,y) + n(x,y) < w \\ w - 1 & \text{if } f(x,y) + n(x,y) \geq w, \end{cases}$$

then we say that image h *contains additive noise*. Image n is called *noise image*.

Chapter 3

The Fourier transform

The aim of this thesis is to create a program for the generation of testing images with defined sub-pixel shift and resize precision. To do that we use convolution kernel (invariable and variable) which is based on discrete convolution. Convolution on digital images is grounded the Fourier transform, so if the digital image is discrete and convolution on it too, the discrete Fourier transform is also needed. However let us first define all these terms as continuous, because the discrete variants were derived from the continuous ones.

3.1 Basic notions

Definition 3.1. ($\mathcal{L}(\mathbb{R})$) Let us denote $\mathcal{L}(\mathbb{R})$ as the space of all functions $\mathbb{R} \to \mathbb{C}$ such that

$$\int_{-\infty}^{\infty} |f(x)| \mathrm{d}x$$

exists and is finite.

Definition 3.2. ($\mathcal{L}(\mathbb{R}^2)$) Let us denote $\mathcal{L}(\mathbb{R}^2)$ as the space of all functions $\mathbb{R}^2 \to \mathbb{C}$ such that

$$\iint_{\mathbb{R}^2} |f(x, y)| \mathrm{d}x \mathrm{d}y$$

exists and is finite.

More about these spaces can be found in [2].

Definition 3.3. (Finite function [1]) A function $f(x, y) : \mathbb{R}^2 \to \mathbb{R}$ is called *finite* if it is equal to zero outside of the Cartesian rectangle $\langle a, b\rangle \times \langle c, d\rangle$, where $a, b, c, d \in \mathbb{R}, a < b$ and $c < d$.

3.2 Continuous Fourier transform

Definition 3.4. (Fourier transform of functions in $\mathcal{L}(\mathbb{R})$ [1]) Let $f(x) \in \mathcal{L}(\mathbb{R})$. The *Fourier transform* of function f is function $\mathcal{F}\{f\}(\xi) = F(\xi) : \mathbb{R} \longrightarrow \mathbb{C}$ defined as

$$F(\xi) = \int_{-\infty}^{\infty} f(x)\mathrm{e}^{-ix\xi}\mathrm{d}x.$$

Function F is also called the *Fourier spectrum* of function f.

Definition 3.5. (Inverse Fourier transform of functions in $\mathcal{L}(\mathbb{R})$ [1]) Let $G(\xi) \in \mathcal{L}(\mathbb{R})$. The *inverse Fourier transform* of function G is function $\mathcal{F}^{-1}\{G\}(x) = g(x) : \mathbb{R} \longrightarrow \mathbb{C}$ defined

$$g(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} G(\xi)\mathrm{e}^{ix\xi}\mathrm{d}\xi.$$

Theorem 3.6. (Fourier inversion theorem for functions in $\mathcal{L}(\mathbb{R})$ [1]) If $f(x) \in \mathcal{L}(\mathbb{R})$ and f is piecewise C^1, then

$$\frac{1}{2\pi}(P.V.)\int_{-\infty}^{\infty} f(\xi)\mathrm{e}^{ix\xi}\mathrm{d}\xi = \lim_{r\to\infty}\frac{1}{2\pi}\int_{-r}^{r} f(\xi)\mathrm{e}^{ix\xi}\mathrm{d}\xi = \frac{\lim_{t\to x+} f(x) + \lim_{t\to x-} f(x)}{2},$$

in particular, if f is continuous, then

$$\frac{1}{2\pi}(P.V.)\int_{-\infty}^{\infty} f(\xi)\mathrm{e}^{ix\xi}\mathrm{d}\xi = f(x).$$

Moreover, if also $F(\xi, \eta) \in \mathcal{L}(\mathbb{R})$,

$$\mathcal{F}^{-1}\{\mathcal{F}\{f(x)\}\} = \frac{\lim_{t\to x+} f(x) + \lim_{t\to x-} f(x)}{2}.$$

and for continuous function f with $F \in \mathcal{L}(\mathbb{R})$ we have

$$\mathcal{F}^{-1}\{\mathcal{F}\{f(x)\}\} = f(x).$$

Proof. Proof can be found in [1]. □

Definition 3.7. (Fourier transform of functions in $\mathcal{L}(\mathbb{R}^2)$ [1]) Let $f(x, y) \in \mathcal{L}(\mathbb{R}^2)$. The *Fourier transform* of function f is function $\mathcal{F}\{f\}(\xi, \eta) = F(\xi, \eta) : \mathbb{R}^2 \to \mathbb{C}$ defined as

$$F(\xi, \eta) = \iint_{\mathbb{R}^2} f(x, y)\mathrm{e}^{-i(x\xi + y\eta)}\mathrm{d}x\mathrm{d}y.$$

Function F is also called the *Fourier spectrum* of function f.

Definition 3.8. (Inverse Fourier transform of functions in $\mathcal{L}(\mathbb{R}^2)$ [1]) Let function $G(\xi, \eta) \in \mathcal{L}(\mathbb{R}^2)$. The *inverse Fourier transform* of the function G is function $\mathcal{F}^{-1}\{G\}(x, y) = g(x, y) : \mathbb{R}^2 \to \mathbb{C}$ defined as

$$g(x, y) = \frac{1}{4\pi^2}\iint_{\mathbb{R}^2} G(\xi, \eta)\mathrm{e}^{i(x\xi + y\eta)}\mathrm{d}\xi\mathrm{d}\eta.$$

Theorem 3.9. (Fourier inversion theorem for functions in $\mathcal{L}(\mathbb{R}^2)$) If $f(x,y) \in \mathcal{L}(\mathbb{R}^2)$ and is continuous on $\mathbb{R}^2$, then for every $(x,y) \in \mathbb{R}^2$

$$f(x,y) = \lim_{\varepsilon \to 0} \frac{1}{4\pi^2} \iint\limits_{\mathbb{R}^2} F(\xi,\eta) \mathrm{e}^{i(x\xi + y\eta)} \mathrm{e}^{-\varepsilon^2 \frac{\xi^2+\eta^2}{2}} \mathrm{d}\xi \mathrm{d}\eta.$$

If also $F(\xi,\eta) \in \mathcal{L}(\mathbb{R}^2)$, then

$$\mathcal{F}^{-1}\{\mathcal{F}\{f(x,y)\}\} = \frac{1}{4\pi^2} \iint\limits_{\mathbb{R}^2} F(\xi,\eta) \mathrm{e}^{i(x\xi+y\eta)} \mathrm{d}\xi \mathrm{d}\eta = f(x,y).$$

Proof. Proof can be made by generalization of Fourier inverse theorem for functions in $\mathcal{L}(\mathbb{R})$. Even more general proof can be found in [3]. □

3.3 Discrete Fourier transform

Definition 3.10. (Discrete Fourier transform [1]) Let $f(x,y) : \{0,1,...,N-1\} \times \{0,1,...,N-1\} = \{0,1,...,N-1\}^2 \longrightarrow \mathbb{C}, N \in \mathbb{N}$. The *discrete Fourier transform* of function $f(x,y)$ is function $\mathcal{D}\{f\}(\xi,\eta) = F(\xi,\eta) : \{0,1,...,N-1\}^2 \to \mathbb{C}$ defined as

$$\mathcal{D}\{f\}(\xi,\eta) = F(\xi,\eta) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y) \mathrm{e}^{-\frac{2\pi i}{N}(x\xi+y\eta)}.$$

Function F is also called the *Fourier spectrum* of function f.

Definition 3.11. (Inverse discrete Fourier transform [1]) Let function $f(x,y)$ be a function $\{0,1,...,N-1\}^2 \to \mathbb{C}, N \in \mathbb{N}$ and let $F(\xi,\eta)$ be its discrete Fourier transform. The *inverse discrete Fourier transform* of function $F(\xi,\eta)$ is function $\mathcal{D}^{-1}\{F\}(x,y) = g(x,y) : \{0,1,...,N-1\}^2 \to \mathbb{C}$ defined as

$$\mathcal{D}^{-1}\{F\}(x,y) = \frac{1}{N^2} \sum_{\xi=0}^{N-1}\sum_{\eta=0}^{N-1} F(\xi,\eta) \mathrm{e}^{\frac{2\pi i}{N}(x\xi+y\eta)}.$$

Theorem 3.12. (Fourier inversion theorem [1]) Let $f(x,y)$ be a function $\{0,1,...,N-1\}^2 \to \mathbb{C}, N \in \mathbb{N}$ and let $F(\xi,\eta)$ be its discrete Fourier transform. Then the inverse discrete Fourier transform of function $F(\xi,\eta)$ is function $f(x,y)$, i.e.

$$\mathcal{D}^{-1}\{\mathcal{D}\{f(x,y)\}\} = f(x,y).$$

Proof. Proof can be found in [1]. □

Chapter 4

Convolution

Discrete convolution is the most used tool in image processing and it is also used in image registration. Most of the graphic programs use discrete convolution in form of the convolution kernel, e.g. for the edge detection, noise removal, blur or sharpening, etc. The convolution kernel is invariable and can also be used to create sub-pixel shifted images. In order to create resized image using the convolution kernel, it needs to be variable at each step (this method is not used in commercial graphic programs). The kernel can be compared to window moving on the digital image (see Figures 5.2 and 6.1 for illustration). The values of convolution kernel then determine the way of computing the values of a new image pixels which are computed separately.

To define discrete convolution and convolution kernel, we at first define continuous convolution which it is derived from.

4.1 Continuous convolution

Definition 4.1. (Convolution [1]) Let functions $f_1(x,y),\ f_2(x,y) \in \mathcal{L}(\mathbb{R}^2)$. The *convolution* $f_1 * f_2$ of functions $f_1,\ f_2$ is a function

$$f(x,y) = \iint\limits_{\mathbb{R}^2} f_1(s,t) f_2(x-s, y-t) \mathrm{d}s\mathrm{d}t.$$

Theorem 4.2. [1] Let functions $f_1(x,y),\ f_2(x,y) \in \mathcal{L}(\mathbb{R}^2)$ with Fourier spectra $F_1(\xi,\eta)$, $F_2(\xi,\eta)$. Then

$$\mathcal{F}\{f_1(x,y) * f_2(x,y)\} = F_1(\xi,\eta) \cdot F_2(\xi,\eta).$$

Proof. Proof can be found in [1]. □

Theorem 4.3. [1] Let functions $f_1(x,y),\ f_2(x,y) \in \mathcal{L}(\mathbb{R}^2)$ with Fourier spectra $F_1(\xi,\eta)$, $F_2(\xi,\eta) \in \mathcal{L}(\mathbb{R})^2$. Let $f_1,\ f_2$ be continuous. Then

$$\mathcal{F}\{f_1(x,y) \cdot f_2(x,y)\} = \frac{1}{4\pi^2} F_1(\xi,\eta) * F_2(\xi,\eta).$$

Proof. Proof can be found in [1]. □

4.2 Discrete convolution

Definition 4.4. (Periodization of function and its Fourier spectrum [1]) Let $f(x, y)$ be a function $\{0, 1, ..., N-1\}^2 \longrightarrow \mathbb{C},\ N \in \mathbb{N}$ and let $F(\xi, \eta)$ be its Fourier spectrum. The *periodization of the Fourier spectrum* F is function $\tilde{F}(\xi, \eta) : \mathbb{Z}^2 \longrightarrow \mathbb{C}$ defined as

$$\tilde{F}(\xi, \eta) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(\xi, \eta) \mathrm{e}^{-\frac{2\pi i}{N}(x\xi + y\eta)}.$$

The *periodization of function* f is function $\tilde{f}(x, y) : \mathbb{Z}^2 \longrightarrow \mathbb{C}$ defined as

$$\tilde{f}(x, y) = \frac{1}{N^2}\sum_{\xi=0}^{N-1}\sum_{\eta=0}^{N-1} F(\xi, \eta) \mathrm{e}^{\frac{2\pi i}{N}(x\xi + y\eta)}.$$

Definition 4.5. (Discrete periodic convolution [1]) Let $f_1(x, y),\ f_2(x, y)$ be functions $\{0, 1, ..., N-1\}^2 \longrightarrow \mathbb{C},\ N \in \mathbb{N}$. Function $f(x, y) : \{0, 1, ..., N-1\}^2 \longrightarrow \mathbb{C}$ is called the *discrete periodic convolution* of functions $f_1,\ f_2$, denoted by $f(x, y) = f_1(x, y) * f_2(x, y)$ if

$$f(x, y) = \sum_{s=0}^{N-1}\sum_{t=0}^{N-1} f_1(s, t)\tilde{f}_2(x - s, y - t).$$

Theorem 4.6. [1] Let functions $f_1(x, y),\ f_2(x, y) : \{0, 1, ..., N-1\}^2 \longrightarrow \mathbb{C},\ N \in \mathbb{N}$ have Fourier spectra $F_1(\xi, \eta),\ F_2(\xi, \eta)$. Then

$$\mathcal{D}\{f_1(x, y) * f_2(x, y)\} = F_1(\xi, \eta) \cdot F_2(\xi, \eta).$$

Proof. Proof can be found in [1]. □

4.3 Application of two-dimensional discrete convolution to digital image

Definition 4.7. (Convolution kernel) Let $f(x, y)$ be a gray-scale image defined in Definition 2.1 and $k(s, t) : \{0, 1, ..., 2n+1\}^2 \longrightarrow \mathbb{R},\ n \in \mathbb{N}$ being called *convolution kernel* or *convolution mask* if

$$f'(x, y) = f(x, y) * k(s, t) = \sum_{s=0}^{2n+1}\sum_{t=0}^{2n+1} f(x - s, y - t)k(s, t)$$

where $f'(x, y)$ is a new image created by transforming image $f(x, y)$ using convolution kernel. The number $2n + 1$ is called the *kernel size.*

A size of the convolution kernel is always an odd number. Due to this a certain middle element always exists and is then applied to the pixel which value we want to change. Usually, the kernel of the size of 3 is used in graphic programs and sometimes even a kernel which size is 5. The kernel size indicates which pixel values we want to incorporate in the creation of a new pixel, i.e. whether we want only the closest neighbours of the changed pixel to be significant (a kernel size of 3) or whether we want also pixels from the neighborhood of the previous mentioned to be significant (a kernel size of 5).

The usage of the convolution kernel in graphic programs is well described in [5].

Chapter 5

Shifted images

To create shifted and scale-changed images we used the source file `hmi.ic_45s.2014.10.-23_15_00_45_TAI.continuum.fits` which was obtained from the NASA webpage, see [4]. In the program, we cropped out image segment of the size of 256×256 pixels, which was shifted by the known shift or scale-changed by the known scale-change factor. The program is based on procedures programmed by prof. Miloslav Druckmüller who supervised further development of the program. From pre-made procedures we use only the procedure for loading images. The innovations of this program can be found in `Unit2`.

First of all we need to define shifted image. According to the assignment of this thesis, we consider only sub-pixel shifts together with the assumption of sub-pixel shift range in one pixel (i.e 0 - 100% shift of single pixel). Shifts larger than one pixel or shifts in opposite direction (in definition we define shift to the right, see Definition 5.1) can be created by the choice of another reference image, cropped with whole-number shift.

Definition 5.1. (Shifted digital image) Let $f_1(x, y)$ be a digital gray-scale image defined in Definition 2.1. Let $px,\ py \in \langle 0, 1)$ and $M \in \mathbb{N}$ be given numbers such that

$$M < N,$$

$$M + px \leq N - 1,$$
$$M + py \leq N - 1,$$

and let $f_2(x, y)$ be also a digital gray-scale image such that

$$f_2(x,y) = \begin{cases} f_1(x + px, y + py) & \text{if } px \leq x \leq M - 1 + px, \\ & \quad py \leq y \leq M - 1 + py, \\ 0 & \text{else.} \end{cases}$$

Then we take in consideration that part of image, where there is some information stored, i.e. the non-zero area. Image $f_2(x, y)$ is then called *shifted image* and vector (px, py) is called *shift vector*.

5.1 Generation of shifted image using weighted average

For the creation of sub-pixel shifted images, we now have to consider the digital gray-scale image to be a net consisted of the squares of the same size representing the pixels. Each square is valued by a gray-scale digital image function $f(x, y)$ applied to each of the squares. Resultant values are written into the image matrix. We consider that squares have an area equal to 1.

Shifted image is then represented as another smaller net (with squares of the same size) placed over the original net, but shifted by the shift vector (px, py) (see Figure 5.1).

Now we need to compute the values of second image matrix. Each value in matrix is computed as weighted average of up to four values of the first matrix's neighbouring values. The weights are given as areas of the first net square parts which are involved in the new square.

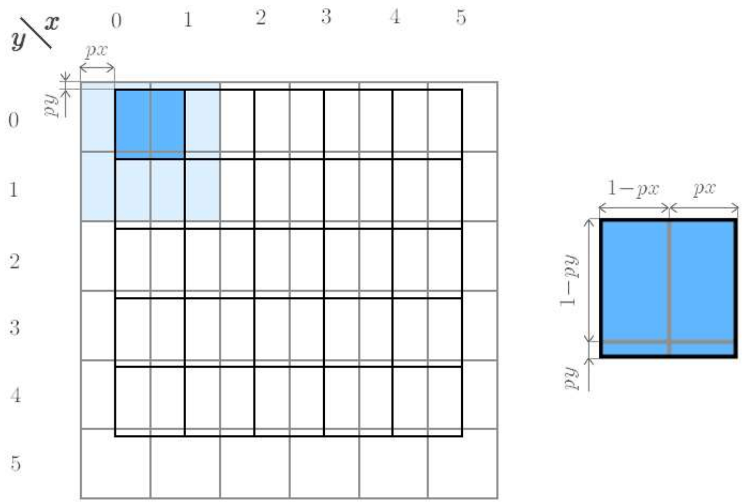

Figure 5.1: Shifted image creation using weighted average

New image matrix can be computed as

$$f'(x,y) = (1-px)(1-py)f(x,y) + px(1-py)f(x+1,y) + \\ +(1-px)py \cdot f(x,y+1) + px \cdot py \cdot f(x+1,y+1).$$

Where $f(x, y)$ is the original image matrix.

5.2 Generation of shifted image using convolution mask

To apply the convolution mask to the creation of sub-pixel shifted image, we need to establish the expected size of the mask. As defined, the size of the mask has to be an odd number and there are no restrictions on its value. In our case, the mask of the size 3 is sufficient enough. Size of convolution mask depends on the amount of pixels which we consider the new pixel is going to be consisted of. For shifted images we need only 4 pixels, so the least needed size of the mask is 3 (see Figure 5.2).

For values of convolution kernel itself, we use previously computed values of weighted average and rest of the kernel elements will be zeros, because they represent pixels which are not included in the new pixel.

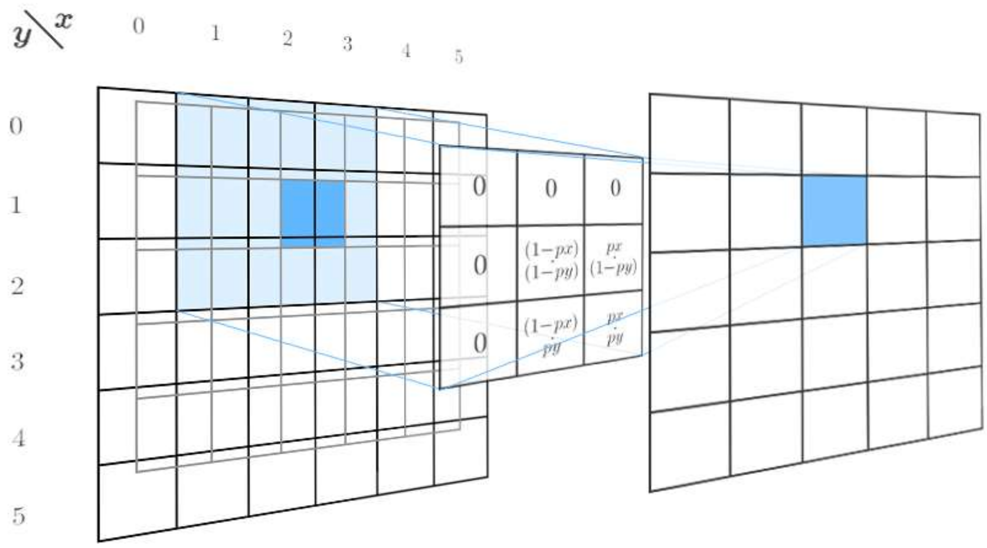

Figure 5.2: Shifted image creation using convolution kernel

Then the new pixel value will be computed as

$$f'(x,y) = f(x,y) * k(s,t) = \sum_{s=0}^{2n+1} \sum_{t=0}^{2n+1} f(x-s, y-t) k(s,t),$$

where convolution kernel is

$$k(s,t) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & (1-px)(1-py) & px(1-py) \\ 0 & px(1-py) & px \cdot py \end{pmatrix}.$$

Chapter 6

Scale-changed images

For the purpose of phase-correlation testing program, only images with decreased sizes will be created. If the precision testing of images with increased sizes is needed, a referential and tested image can be switched. Also images smaller than 50 % of the original image size won't be generated, because a sufficient precision of the phase-correlation testing can not be reached for smaller images.

Definition 6.1. (Scaled image) Let $f_1(x, y)$ be a digital gray-scale image. Let $\alpha \in \mathbb{R}^+$ and $M \in \mathbb{N}$ be given numbers such that

$$\alpha M < N,$$

and let $f_2(x, y)$ be also a digital gray-scale image such that

$$f_2(x,y) = \begin{cases} f_1(\alpha x, \alpha y) & \text{if } 0 \le x \le \alpha M, \\ & \quad 0 \le y \le \alpha M, \\ 0 & \text{else.} \end{cases}$$

Again we take into consideration only the part of the image, where some information is stored. Image $f_2(x, y)$ is then called *scale-changed image* and α is called *scale-change factor*.

6.1 Generation of scale-changed images using variable convolution mask

A variable convolution kernel will be used to generate scale-changed images. Unlike in the case of generating shifted images, the kernel has to be variable, i.e. it has to be computed separately for every pixel. However there is a certain rule for the process of getting the desired values.

To generate scale-changed images in the range of 50 - 100 %, we will also need a kernel of the size of 3 because the new pixel will never be composed of more than 9 original pixels (see Figure 6.1).

The new pixel value can be computed as

$$f'(x,y) = f(x,y) * k(s,t) = \sum_{s=0}^{2n+1} \sum_{t=0}^{2n+1} f(x + \lfloor (x+1)p \rfloor - s, y + \lfloor (y+1)p \rfloor - t) k(s,t).$$

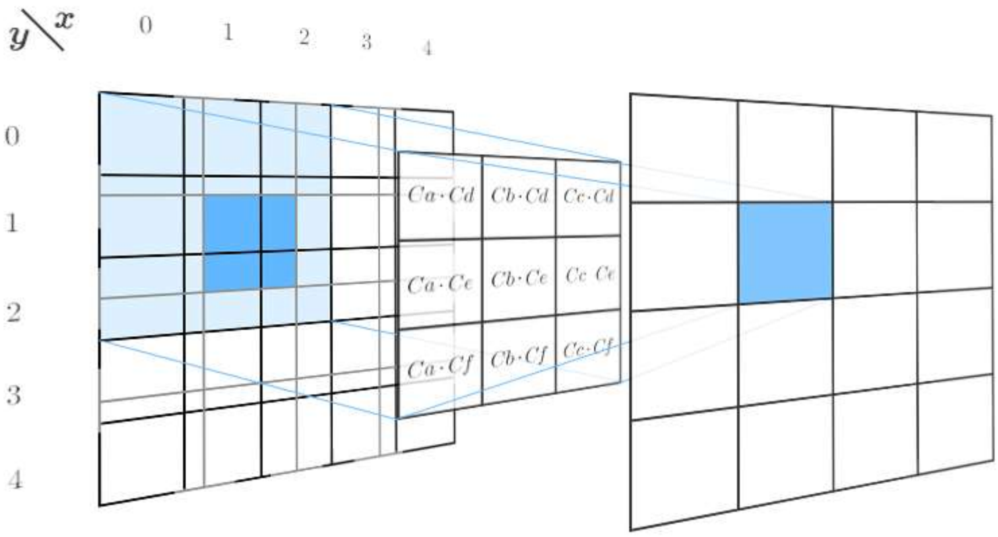

Figure 6.1: Scale-changed image creation using variable convolution kernel

With variable convolution kernel, the pixel, which we read from, also has to be variable in case that the $(x+1)p$ or $(y+1)p$ value exceeds the subsequent integer.
Convolution kernel

$$k(s,t) = \begin{pmatrix} Ca \cdot Cd & Cb \cdot Cd & Cc \cdot Cd \\ Ca \cdot Ce & Cb \cdot Ce & Cc \cdot Ce \\ Ca \cdot Cf & Cb \cdot Cf & Cc \cdot Cf \end{pmatrix}$$

and the values of Ca, Cb, Cc, Cd, Ce and Cf will be

$$Ca = \begin{cases} \lfloor (k+1)p \rfloor - k \cdot p & \text{if} \lfloor (k+1)p \rfloor > \lfloor k \cdot p \rfloor, \\ 0 & \text{else,} \end{cases}$$

$$Cb = \begin{cases} 1 & \text{if} \lfloor (k+1)p \rfloor > \lfloor k \cdot p \rfloor, \\ \lfloor k \cdot p \rfloor + 1 - k \cdot p & \text{else,} \end{cases}$$

$$Cc = \begin{cases} (k+1)p - \lfloor (k+1)p \rfloor & \text{if} \lfloor (k+1)p \rfloor > \lfloor k \cdot p \rfloor, \\ (k+1)p - \lfloor k \cdot p \rfloor & \text{else,} \end{cases}$$

$$Cd = \begin{cases} \lfloor (l+1)p \rfloor - l \cdot p & \text{if} \lfloor (l+1)p \rfloor > \lfloor l \cdot p \rfloor, \\ 0 & \text{else,} \end{cases}$$

$$Ce = \begin{cases} 1 & \text{if} \lfloor (l+1)p \rfloor > \lfloor l \cdot p \rfloor, \\ \lfloor l \cdot p \rfloor + 1 - l \cdot p & \text{else,} \end{cases}$$

$$Cf = \begin{cases} (l+1)p - \lfloor (l+1)p \rfloor & \text{if} \lfloor (l+1)p \rfloor > \lfloor l \cdot p \rfloor, \\ (l+1)p - \lfloor l \cdot p \rfloor & \text{else,} \end{cases}$$

where $k, l \in \mathbb{N}$ and $p \in \langle 0, 1 \rangle$ can be computed as

$$p = ((1-\alpha)^{-1} - 1)^{-1},$$

$$k = x \bmod a,$$

$$l = y \bmod a,$$

where $a \in \mathbb{N}$ is the smallest number given as

$$\lfloor a \cdot p \rfloor = a \cdot p.$$

Chapter 7

Conclusion

The aim of this thesis was to create a program for generating defined sub-pixel shifted images and resizing images using variable convolution kernel (see Chapters 5 and 6). To achieve that we had to draw up the pattern of the kernel.

In Chapter 2, we defined a digital image. The most important part is the definition of gray-scale image (see Definition 2.1) which is used in description of the image generation process.

Chapters 3 and 4 summarise the theory of Fourier transform and convolution which are essential to define the convolution kernel. The definitions are first introduced in general form on continuous spaces and then we proceed to their discrete version. These are more suitable for the image processing, especially for the gray-scale image defined in Chapter 2.

Finally Chapters 5 and 6 discuss the image generation itself and analyse two different approaches of the process. These two chapters are the major part of this thesis. The scale-changed images generation uses a rather uncommon way of the resizing, which had to be drawn up completely.

Bibliography

[1] DRUCKMÜLLEROVÁ, H., *Phase-correlation based image registration.* Brno, 2010. Master's thesis, Fakulta strojního inženýrství Vysoké učení technické v Brně.

[2] FOLLAND, Gerald B., *Fourier analysis and its applications.* Providence, R.I.: American Mathematical Society, 2009. Sally series (Providence, R.I.), 4. ISBN 0821847902.

[3] STEIN, Elias M. and WEISS, Guido L., *Introduction to Fourier analysis on Euclidean spaces.* Princeton, N.J.: Princeton University Press, 1975. ISBN 0-691-08078-x.

[4] The Virtual Solar Observatory, *The Virtual Solar Observatory* [online]. NASA Goddard Space Flight Center: Solar Data Analysis Center, c2005-2017 [cit. 2017-03-27]. Available at: http://sdac.virtualsolar.org/cgi/search.

[5] ŽÁRA J.; et al., *Moderní počítačová grafika.* Brno: Computer Press, second edition, 2004, ISBN 80-251-0454-0.

Used symbols

$\mathbb{N}$	the set of natural numbers
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^+$	the set of positive real numbers, i.e $(0, \infty)$
$\mathbb{C}$	the set of complex numbers
$\mathcal{L}(\mathbb{R})$	space of all functions $\mathbb{R} \longrightarrow \mathbb{C}$ with finite integral of $\lvert f \rvert$, see Definition 3.1
$\mathcal{L}(\mathbb{R}^2)$	space of all functions $\mathbb{R}^2 \longrightarrow \mathbb{C}$ with finite integral of $\lvert f \rvert$, see Definition 3.2
C^1	class of continuous functions with continuous derivatives
$\lfloor a \rfloor$	the integral part of real number a
$\mathcal{F}$	the Fourier transform, see Definitions 3.4, 3.7
$\mathcal{F}^{-1}$	the inverse Fourier transform, see Definitions 3.5, 3.8
$\mathcal{D}$	the discrete Fourier transform, see Definition 3.10
$\mathcal{D}^{-1}$	the inverse discrete Fourier transform, see Definition 3.11
$f(x, y), f_1(x, y), f_2(x, y)$	functions from $\mathcal{L}(\mathbb{R}^2)$ or functions $\{0, 1, ..., N-1\}^2 \longrightarrow R, N \in \mathbb{N}$
$F(\xi, \eta), F_1(\xi, \eta), F_2(\xi, \eta)$	the Fourier spectra of functions $f(x, y), f_1(x, y), f_2(x, y)$, see Definition 3.7
$f_1(x, y) * f_2(x, y)$	the convolution of function f_1 and f_2, see Definitions 4.1, 4.5
(px, py)	the shift vectors between images f_1, f_2, see Definition 5.1
α	the scale-change factor between images f_1, f_2, see Definition 6.1
$\tilde{f}, \tilde{f}_1, \tilde{f}_2$	the periodization of functions f, f_1, f_2, see Definition 4.4
$k(s, t)$	convolution kernel/mask, see Definition 4.7
$f'(x, y)$	new image made by transforming original image $f(x, y)$ using convolution kernel Definition 4.7

Appendix

CD with text files and program in RAD Studio 10.1 Berlin.