CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Electrical Engineering

Diploma thesis

Unconventional methods of image acquisition

May 2014

made by: Helena Procházková
supervisor: Ing. Jan Bednář

Statutory Declaration

Statutory Declaration I declare that I have developed and written the enclosed Master Thesis completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked.

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Date: 12. 5. 2014

. . . . . .

signature

!!! Original zadani !!!

...

Annotation

This project presents a theoretical foundation for Compressive sensing framework (CS) which is new mathematical approach for compressing images with sub-Nyquist image acquisition. The project introduces new efficient constructions of digital cameras based on digital micro-mirror device (DMD) and compressive sensing approach which can operate efficiently across a much broader spectral range than conventional CCD-based cameras and two constructions of lensless imaging systems. One approach uses SLM modulator (transparent LCD display) and a photo-diode, the second one consists of an image detector and a special aperture. By only modifying the transmittance pattern of the aperture it is possible to change its viewing direction instantaneously to arbitrary directions. Image acquisition process is simulated under various conditions and SLM modulator based approach is constructed and tested in real life.

Anotace

Tato práce předkládá teoretický podklad pro Compressive sensing algoritmus (CS), což je nový matematický přístup pro kompresi se sub-Nyquistovským rozlišením. Tato práce představuje nové efektivní způsoby konstrukce digitálních kamer využívající mikrozrcátka (DMD) a compressive sensing algoritmus. Tento přístup umožňuje snímaní v širším spektrálním pásmu než klasická kamera se CCD snímačem. Dále práce pojednává o dvou typech konstrukcí bezobjektivového snímaní. První přístup používá místo objektivu SLM modulátor (průsvitný LCD displej) a jako detektor fotodiodu. Druhá konstrukce bezobjektivového snímání se skládá z detektoru aspeciální apertury. Pouze změnou propustnosti apertury je možné okamžitě změnit úhel pohledu do libovolného směru. Proces snímání obrazu je simulován v různých podmínkách a konstrukce se SLM modulátorem je zkonstruována a otestována při skutečném použití.

Acknowledgements

I want to thank all the graduate students at the department of radioelectronics creating an enjoyable atmosphere and for answering all the (often irrational) questions I came up with. Things could have been a lot more difficult without you.

Most importantly I wish to thank my supervisor Ing. Jan Bednář. This topic is very challenging and because of that your suggestions, advices and support have been essential during this journey. I do not forget about the fruitful conversations we had during this project which were vital help to me. You were eager to help me in this process and thus managed to keep my motivation up also when things just did not go the way we wished.

I would also like to thank Jana Štěpánová for providing me breathtaking pictures, Vojtěch Kolenčík for important advices and emotional support.

I would like to express my gratitude to my family, as this could not have been accomplished without their love and support. Finally, I want to thank my friends for their patience with me, while I was whining about my thesis.

Contents

List of abbreviations

CCD Coupled Charged Device

CoSaMP COmpressive SAmpling Matching Pursuit

CMOS Metal Oxid Silicon

CS Compressive Sensing

DCT Discrete Cosine Transform

DMD Digital Micro-mirror Device

JPEG Joint Picture Experts Group (picture format)

LCD Liquid Crystal Display

Laser Light Amplification by Stimulated Emission of Radiation

LED Light Emitting diode

MP Matching Pursuit

MSE Mean Square Error

OMP Orthogonal Matching

PSF Point Spread Function

PSNR Peak Signal to Noise Ratio

RGB Red Green Blue

RIP Restricted Isometry Property

SLM Spatial Light Modulator

SSIM Structural SIMilarity index

USB Universal Serial Bus

VGA Video Graphic Array

List of Figures

Key words

Compressive sensing, DCT transformation, Lensless imaging, Spatial Light Modulator, Digital Micro-mirror Device, Point Spread Function, Image acquisition, Single-pixel camera, Controllable aperture

Chapter 1

Introduction

Traditional digital camera construction consists of a camera body and a lens. Both should be of good quality for taking satisfying pictures. Lenses are useful for focusing light, admitting more light into the system so that we do not need longer exposure times in order to get bright and sharp pictures. This approach is well established, but recently new image acquisition methods emerged which in some particular applications provide advantages over common digital camera with lens and CCD sensor.

Various non-conventional image acquisition approaches are examined in this work: lensless camera based on SLM modulator, single-pixel camera, lensless camera with controllable aperture. These novel imaging methods strive to be cheap, because of the lack of expensive lenses which are expensive to manufacture, fast - no mechanism and thanks to compressive sensing (CS) we can reconstruct an image of the scene from fewer measurements than the number of reconstructed pixels.

Many natural and man made signals are compressible in a way that just few large coefficients of some Fourier-related transform are needed to reconstruct the original signal faithfully. Compressive sensing is a mathematical framework which specifies how to compress the image and also defines how to reconstruct the compressed data back to the image. Many alternative algorithms were found which enable faster and more precise reconstruction. In this paper we use the CoSaMP (COmpressive SAmpling Matching Pursuit) algorithm published in Needell and Tropp 2008. As the verification of the theoretical background of COSaMP algorithm, a simulation in Matlab computing environment is used.

The single-pixel based camera tries to deal with the sensitivity limitation of silicon material converting photons to electrical signal, which works only at only visual wavelengths. Also there is a huge pressure to come up with even faster capture, sampling and processing rates and requires lower power consumption. A new principle called Single-Pixel CS camera has been invented, based on digital micro-mirror device (DMD) and compressive sensing approach which can operate efficiently across a much broader spectral range than conventional silicon-based cameras within reasonable cost.

Other kind of construction based on compressive sensing approach is a camera which uses no lens. It is cheaper, lighter, smaller, and no physical mechanism is needed for forming the image before it is digitally captured. Also, the lack of the lens causes no blurring for parts of scene that are out of focus which occurs with conventional lens equipped camera – lensless camera is not focused to any particular distance. Its construction is very simple. Instead of lens it uses SLM modulator (transparent LCD display) and a photo-diode. This is the method which was chosen for implementation as a part of this work.

Another similar approach is lensless imaging with controllable aperture. Architecture of this design is very simple. It consist of an image detector and a special aperture. By only modifying the transmittance pattern of the aperture, we change its viewing direction instantaneously to arbitrary directions. We can also use this architecture of lensless camera for capturing just some parts of interest of the scene at the time.

The work is organized as follows. The second chapter provides mathematical and theoretical background for compressive sampling process. The third chapter discusses various unconventional image acquisition methods. The fourth chapter describes general image reconstruction algorithm which forms the basis of many image acquisition approaches. Finally, the fifth chapter describes the process of constructing one of the image acquisition systems and performing measurements on it.

Chapter 2

Compressive sensing

2.1 Introduction

For a proper digitalization of an image, it is necessary to satisfy the Shannon-Nyquist theorem in order to not lose any information. We therefore have to sample a signal with a rate two times higher then the highest frequency contained in the signal. This leads us to too many samples which must be compressed in order to store them. A/D converters are getting really expensive because of the requirements for sampling rate.

The basic principle of lossy compression is to sample the whole image and use some Fourier-related transform (for example *discrete cosine transform* – DCT) for frequency analysis of the input data. Coefficients are computed by DCT and then analyzed for which of them are necessary to transmit/store. This method wastes computational resources by computing some coefficients which are ultimately not used. *Compressive sensing* (CS), on the other hand, does the sampling and compression in one step, which results in obtaining only the coefficients necessary for restoring the image. The advantage of using compressive sensing is using single detector (as opposed to one detector for each pixel in conventional approach) which reduces size and costs of the whole device. Because only single detector is necessary, it's possible to use more expensive detectors with broader spectral range or other desirable properties. A camera based on CS can efficiently handle high-dimensional data sets, for example video or hyperspectral imaging, thanks to its capability of almost immediate compression while taking picture.

2.2 Traditional approach

The traditional approach is to represent digital image by a 2D array of N pixel sensors on a CCD or CMOS chip. Let's set an discrete-time signal x as an $N \times 1$ column vector in $\mathbb{R}^N$ with the samples $x[n] = 1, 2, \ldots, N$. As already mentioned, in order to satisfy the Shannon/Nyquist theorem

and not lose any information, N is potentially a very large number, and the signal x is therefore compressed by a multi-step transform coding process.

Any signal x can be expressed as

$$\mathbf{x} = \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{s_i} \cdot \psi_\mathbf{i} = \mathbf{\Psi} \cdot \mathbf{s} \tag{2.1}$$

where $\psi_\mathbf{i}$ is i-th vector of a basis Ψ (for simplicity let's assume that the basis is orthonormal), s_i is the $N \times 1$ column vector of weighting coefficients, $\mathbf{\Psi}$ is the basis matrix of the basis Ψ. We can compute the weighting coefficients as $s_i = \langle \mathbf{x} \cdot \psi_\mathbf{i} \rangle$. Basically the vector $\mathbf{x}$ represents the signal in time/spatial domain and the coefficients s are coordinates with respect to the basis Ψ.

Many natural and man made signals are compressible in a way that just few large coefficients are needed to reconstruct the original signal. For these signals, DCT (or other transform) yields many zero (or near zero) coefficients, with only a few coefficients having some substantial value. We call these vectors of coefficients representing the original signal *sparse representations*, meaning we only need few coefficients to reconstruct the original signal accurately. We are looking for a basis Ψ in which natural signals we are working with will have this kind of sparse representation, meaning that vector x will be a linear combination of just K basis vectors where $K << N$. This principle is utilized in discrete cosine transform (DCT) and wavelet bases on which JPEG and JPEG-2000 compression standards are based.

This sample-then-compress framework suffers from two inherent inefficiencies. The first one is that a large number N of samples is acquired even though only a small number K of coefficients is ultimately used. Secondly, all N transform coefficients are computed but $N - K$ of them are discarded after the computation and never used.

2.3 Sparse approximation

For better understanding of the compressive sensing process, let's define an ℓ_0 norm which is used to measure the sparsity of a vector. The ℓ_0 norm of an n-dimensional vector x is ([7]):

$$\|x\|_0 = \text{Card}(\{k : x \neq 0, k = 1, \ldots, n\}) \tag{2.2}$$

where Card denotes cardinality of the set, i.e. number of its elements. In other words, the ℓ_0 norm is equal to number of non-zero elements of vector x. Vectors with small ℓ_0 norm (i. e. having many zero elements) are called *sparse vectors*.

Working with sparse vectors numerically saves computation time needed for operations like multiplying a vector by a matrix. Also, it requires less memory in a computer since only values of non-zero elements and their positions need to be stored.

Let's solve a problem: find a vector $\mathbf{x} \in \mathbb{C}^n$ which satisfies a set of equations:

$$\mathbf{Ax} = \mathbf{b} \tag{2.3}$$

where $\mathbf{A} \in \mathbb{C}^{(m+n)}$ is an m-times-n matrix and $\mathbf{b} \in \mathbb{C}^m$ is a vector of length m, $m < n$. This problem has less equations then unknowns so it has infinitely many solutions. For reasons discussed later, we will want to find a solution that is sparse. This choosing of sparse vector is called *sparse representation* of $\mathbf{b}$ because we can exactly reproduce $\mathbf{b}$ from vector $\mathbf{x}$. This leads us to *Sparse approximation* definition. Let's have more general version of 2.4 and find $\mathbf{x}$:

$$\|Ax - b\| < \epsilon \tag{2.4}$$

where $\epsilon \in \mathbb{R}$, *epsilon* > 0, is a tolerance with which we want to approximate the vector $\mathbf{x}$. Since it is no longer required to reconstruct $\mathbf{b}$ exactly, there is way more choices of vector $\mathbf{x}$.

Example of sparse ($\ell_0 = 2$) representation using a randomly generated vector and matrix:

$$\begin{pmatrix} 0.5377 & 0.8622 & -0.4336 & 2.7694 \\ 1.8339 & 0.3188 & 0.3426 & -1.3499 \\ -2.2588 & -1.3077 & 3.5784 & 3.0349 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1.8750 \\ 0.1547 \end{pmatrix} = \begin{pmatrix} -0.3846 \\ 0.4336 \\ 7.1790 \end{pmatrix} = \mathbf{a}$$

Example of sparse approximation ($\ell_0 = 1$) using a randomly generated vector and matrix:

$$\begin{pmatrix} 0.5377 & 0.8622 & -0.4336 & 2.7694 \\ 1.8339 & 0.3188 & 0.3426 & -1.3499 \\ -2.2588 & -1.3077 & 3.5784 & 3.0349 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1.8750 \\ 0 \end{pmatrix} = \begin{pmatrix} -0.8130 \\ 0.6424 \\ 6.7095 \end{pmatrix} = \mathbf{a}' \approx \begin{pmatrix} -0.3846 \\ 0.4336 \\ 7.1790 \end{pmatrix}$$

In the sparse approximation an error (the size of the difference of the two outcomes) is noticeable which is $\|\mathbf{a} - \mathbf{a}'\|_2 = 0.669$. Compared to vector's length 6.7890, the error is small. Finding a sparse approximation is not simple, mainly because there is an infinite amount of solutions.

2.4 Restricted isometry property (RIP)

Let's have a K-sparse signal x of length $N \times 1$ and a measurement matrix Φ of size $M \times N$. The measurement process has vector $\mathbf{x}$ as input and produces measurement $\mathbf{\Phi x}$. From this measurement, the original vector $\mathbf{x}$ is reconstructed. For the reconstruction to be possible, two different input vectors ($\mathbf{x_1}$ and $\mathbf{x_2}$) must be mapped onto two different measurements ($\mathbf{\Phi x_1}$ and $\mathbf{\Phi x_2}$), provided they are K-sparse. Also, the reconstruction process must be stable, meaning that small change in $\mathbf{\Phi x}$ must yield small change in the reconstructed vector $\mathbf{x}$. Candés and Tao ([8]) proposed a

condition for the measurement matrix Φ.

$$(1 - \delta_K)\|\mathbf{x}\|_\mathbf{2}^\mathbf{2} \leq \|\mathbf{\Phi x}\|_\mathbf{2}^\mathbf{2} \leq (\mathbf{1} + \delta_\mathbf{K})\|\mathbf{x}\|_\mathbf{2}^\mathbf{2} \tag{2.5}$$

where $\|\|_2$ denotes the ℓ_2 norm. This condition is called *K-restricted isometry property.* For stability of reconstruction, measurement matrix must map two different vectors to two different vectors. If the input differs $(\mathbf{x_1} - \mathbf{x_2})$ then the output must differ $(\mathbf{\Phi x_1} - \mathbf{\Phi x_2})$ accordingly, therefore

$$(1 - \delta_K)\|\mathbf{x_1} - \mathbf{x_2}\|_\mathbf{2}^\mathbf{2} \leq \|\mathbf{\Phi x_1} - \mathbf{\Phi x_2}\|_\mathbf{2}^\mathbf{2} \leq (\mathbf{1} + \delta_\mathbf{K})\|\mathbf{x_1} - \mathbf{x_2}\|_\mathbf{2}^\mathbf{2} \tag{2.6}$$

which follows directly from 2.5, must hold for all K-sparse vectors. $\|\mathbf{x_1} - \mathbf{x_2}\|_\mathbf{2}$ is larger than zero, because x_1 and x_2 are different vectors, and therefore $\|\mathbf{\Phi x_1} - \mathbf{\Phi x_2}\|_\mathbf{2}$ must also be larger than zero.

It is computationally difficult to get the result if the measurement matrix Φ satisfies the inequality (and therefore has K-restricted isometry property). There are fortunately many types of random matrices with a good restricted isometry behavior, and they satisfy the restricted isometry condition with high probability. If 2.7 holds and matrix's elements are *i.i.d.* (independent and identically distributed, Gaussian etc.), then the RIP can be considered satisfied with high probability:

$$M \geq O(Klog(N/K)) \ll N \tag{2.7}$$

2.5 Compressive sensing process

Compressive sensing skips the intermediate stage of taking N samples and goes straight into a compressed representation by computing $M < N$ linear measurements between input x and a collection of vectors $\{\Phi_j\}_{j=1}^M$ yielding measured values $y_j = \langle x, \Phi_j \rangle$. To get a $M \times N$ matrix Φ we stack measurement weighing vectors Φ_j^T as rows and measurements y_j into the $M \times 1$ vector $\mathbf{y}$ so that we can write

$$y = \Phi x = \Phi \Psi s = \Theta s \tag{2.8}$$

where Ψ represents DCT matrix (therefore $x = \Psi s$ computes inverse DCT) and $\Theta = \Phi\Psi$ is an $M \times N$ matrix. It is worth noting that this process, as described, works with vectors only, but when working with 2D images (which are represented as matrices), we can serialize these element-by-element into vector and then perform the process as described here. The resulting reconstructed vector is then reshaped back into matrix of the original size.

The base matrix Ψ is constructed as follows: each column of Ψ contains DCT of one basis vector. For each vector in orthonormal base (having one element with value one and other elements having zero values) we perform DCT transform and serialize the result into a column vector. Computing these vectors for each member of the orthonormal base and stacking them together as columns yields

the matrix Ψ. When working with images, we are computing 2D DCT of matrices with exactly one element having value of one and serializing the resulting matrix into a column vector.

This process in is non-adaptive in that Φ and Ψ do not depend on in any way on the signal x.

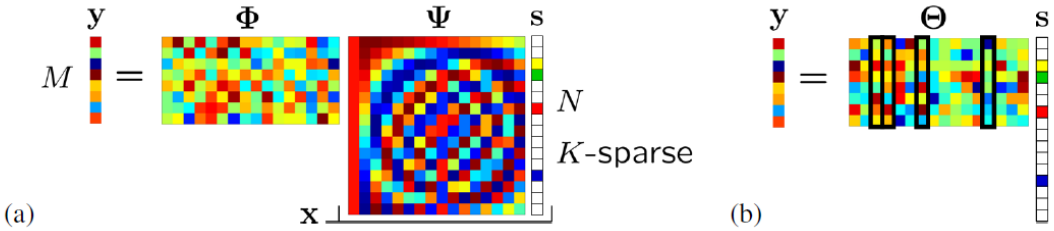


Figure 2.1: Compressive sensing measurement [2]

For the CS to work two conditions must be observed. Firstly, the matrix Φ should be random (in practice this is done by pseudo-random number generator), see [2]. Second requirement is that a reconstruction algorithm must be developed whose task is to recover the original signal x from the measurements y.

Recovering the image x from the random measurements y can be done using ℓ_1 optimization: We define the ℓ_p norm of the vector s as $(\|s\|_p)^p = \sum_{i=1}^{N} |s_i|^p$. If $p = 0$ we get the ℓ_0 norm that counts the number of non-zero entries in s (we say vectors with K zero elements to be *K-sparse*). We want to find a vector α which satisfies the equation for a possible solution and has the minimal ℓ_0 norm or in other words has the highest number of zero elements from all possible solutions:

$$\alpha = \text{argmin}\|\alpha'\|_0 \text{ such that } \Psi\Phi\alpha = y \tag{2.9}$$

By using just

$$M \geq O(Klog(N/K)) \tag{2.10}$$

random measurements we can exactly reconstruct K-sparse vectors and closely approximate compressible vectors. This is a convex optimization problem.

Chapter 3

Unconventional imaging methods

The theoretical ideas introduced in the Chapter 2 can be implemented in practice in various ways. The common feature of these solutions is some light modulator (in the solutions described here it's either transparent LCD display or an array of micromirror devices) and a light detector which provides the measured values from which the original image is reconstructed. The light detector together with the modulator are basically performing the matrix multiplication between input image x and the matrix Θ displayed on the light modulator.

3.1 Single-Pixel CS camera

Silicon material's ability to convert photons at visual wavelengths into electrons led to the evolution of CCD and CMOS digital technologies. These kinds of sensors are cheap and ubiquitous, but their sensitivity is limited to a certain part of the light spectrum. In this chapter we will describe a novel kind of camera architecture based on digital micro-mirror device (DMD) with a new mathematical theory and algorithm of compressive sensing (CS). It contains optical computer which consist of DMD array, two lenses, a single photon detector and an analog-to-digital (A/D) converter. It's main function is to compute random linear measurements of the scene under view. Having only one detector enables us to use more exotic light detectors having broader spectral ranges than traditional CCD/CMOS sensors.[1]

3.1.1 Architecture

The light beam from a light source illuminates photographed object and its reflection is subsequently focused by biconvex Lens 1 (see fig. 3.1.1) onto a DMD array which consists of N electrostatically actuated micro-mirrors arranged in a rectangular matrix. (It is not focused on CCD or CMOS chip as usual.) Each micro-mirror is equipped with a random access memory (SRAM) cell. According to the value stored in this memory, each micro-mirror rotates around a hinge which enables each micro-mirror to be set independently into one of two positions: +10 deg

and -10 deg from horizontal. This causes the light falling onto each DMD to be reflected to either towards Lens 2 (for logical one) or away from it (for logical zero). All micro-mirrors in the array are operated independently and their orientation is set simultaneously. Each pixel is represented by one micro-mirror and each micro-mirror receives different light intensity from the photographed object. We will use $x[n]$ for the signal (light intensity) arriving at micro-mirror n. We will also use $\Phi_m[n]$ to designate orientation of mirror n, $\Phi_m[n] = 1$ if the mirror n is oriented towards Lens 2 and $\Phi_m[n] = 1$ if it is oriented away from it. The Lens 2 collects the reflected light from all micro-mirrors which are oriented towards it (set to logical 1) and focuses it onto a single photon detector which integrates all the values $x[n] \cdot \Phi_m[n]$ to compute the measurement $y[m] = \langle x, \Phi_m \rangle$ (scalar multiplication) as its output voltage, which is then digitalized by A/D converter. Otherwise the light beam is oriented away from Lens 2, as it is shown at picture 3.1.1.

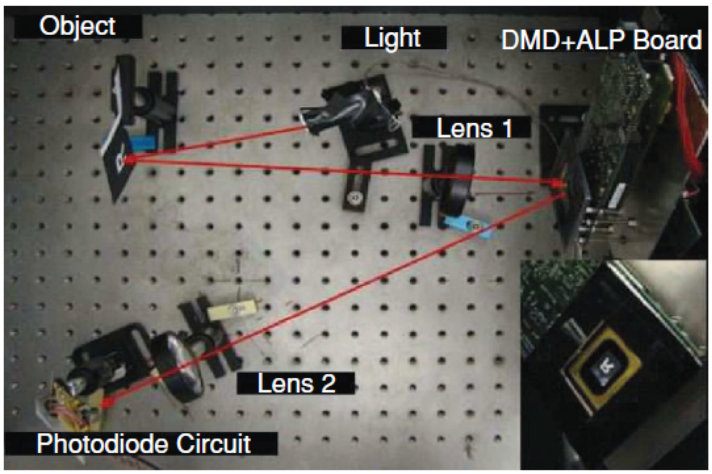


Figure 3.1: Scheme of the single-pixel CS camera in a lab [2]

By using pseudo-random generator we set the micro-mirror orientations Φ_m randomly and compute CS randomized measurements $y = \Phi \cdot x$ and then repeat the process M times to get the vector $\mathbf{y}$.

3.1.2 Examples of image reconstruction

3.2 Lensless imaging

One of the many approaches to compressive sensing is an architecture which enables compressive imaging without any lens. Its properties are as follows. Architecture of the camera uses no lens and camera is therefore cheaper, lighter, smaller, and no physical mechanism is needed for forming the image before it is digitally captured. Also, the lack of the lens causes no blurring for parts of

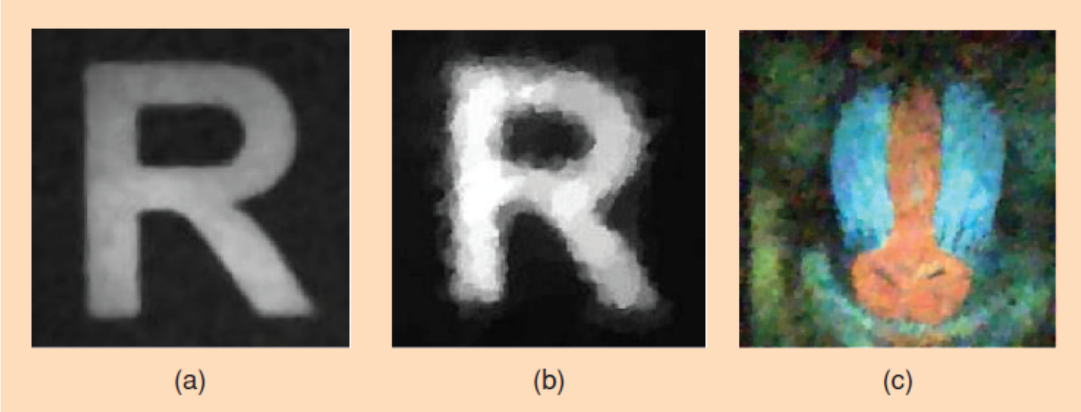


Figure 3.2: Single-pixel photo album. (a) 256×256 conventional image of black-and-white R. (b) Single-pixel camera reconstructed image from $M = 1.300$ random measurements ($50x$ sub-Nyquist). (c) 256×256 pixel color reconstruction of a printout of the Mandrill test image imaged in a low-light setting using a single photomultiplier tube sensor, RGB color filtres, and $M = 6.500$ random measurements. (From [1])

scene that are out of focus which occurs with conventional lens equipped camera – lensless camera is not focused to any particular distance. The sharpness and resolution of images taken are limited only by the resolution of the aperture assembly which is the number of aperture elements ("pixels", described in the next paragraph). This architecture can be used in many other applications not only because of the capability of imaging in other spectra such as infrared and millimeter waves, but also for detecting anomalies or extracting features such as speed of moving objects.

The architecture consists of the aperture assembly, in this case transparent LCD display, which is made of two dimensional array of *aperture elements* and a single sensor made of three-color photo-electric detector. There is no lens used. Each aperture element has own transmittance which can be individually controlled. The sensor, which can be considered as a single detection element, is used for taking compressive measurements. In the picture 3.2 , there is a geometrical context

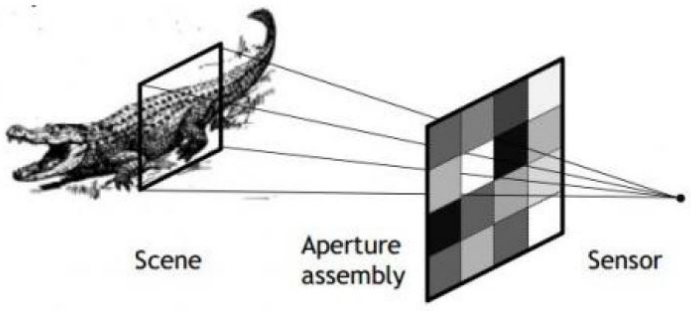


Figure 3.3: principle of lensless imaging using aperture assembly[3]

between an element of an aperture assembly and a sensor which forms cone of a bundle of rays.

Pixel value of the image (the value measured by the detector) is defined by integral value over all rays within a cone from all aperture elements.

3.2.1 Compressive sensing process for lensless camera

Patterns displayed on the aperture assembly are generated in accordance with the theoretical background described in the Chapter 2.5. By using compressive sensing, less measurements than the number of pixels are needed. Let's define a sensing matrix. It is a random matrix with values from 0 to 1. The number of columns in the sensing matrix is equal to the number of total elements in an aperture assembly and each row of the sensing matrix completely defines a pattern of transmittance of aperture assembly elements. The number of rows of the sensing matrix corresponds to the number of measurements which is less than the number of pixels (elements in the aperture assembly). Each measurement from the sensor is the integration of the intensity of all rays going through the aperture assembly multiplied by transmittance of respective aperture element. The intensity is modulated by the transmittances before the integration. Therefore a measurement from a sensor is a projection of an image onto the corresponding row of the sensing matrix. [3]

There is also another variation of this architecture called multi-view imaging. By using more sensors than just one, a set of measurement vectors is taken simultaneously. This fact has a huge advantage. If the scene is far away enough and sensors are close enough to each other, there is a high correlation between the images taken by each sensor. These measurements from each sensor can be considered as independent measurements of the same image and used for reconstruction of the image. This fact causes enhancement of the quality of reconstructed images.

3.3 Lensless imaging with controllable aperture

The basic principle of this new design and comparison to an ideal lens camera is shown in picture.

This idea of this kind of architecture leads us to present its capabilities. By only modifying the transmittance pattern of the aperture, we change its viewing direction instantaneously to arbitrary directions. The conventional cameras are limited by mechanical constraints and produce motion blur. We can also use this architecture of lensless camera for capturing just some parts of interest of the scene at each time distance. By selecting just some area of the scene, we will get images of far apart moving objects with higher resolution. In contrast, conventional cameras are forced to distribute the limited resolution of the detector uniformly over a wide field of view [4].

In a lens based camera, each point of the image detector integrates the light from single point in the scene. The purpose of the lens here is to increase the total brightness of the image and influencing the local blur in defocused areas. On the other hand, in a lensless imaging system with controllable aperture, the light is integrated from entire field of view into each point on a single image detector.

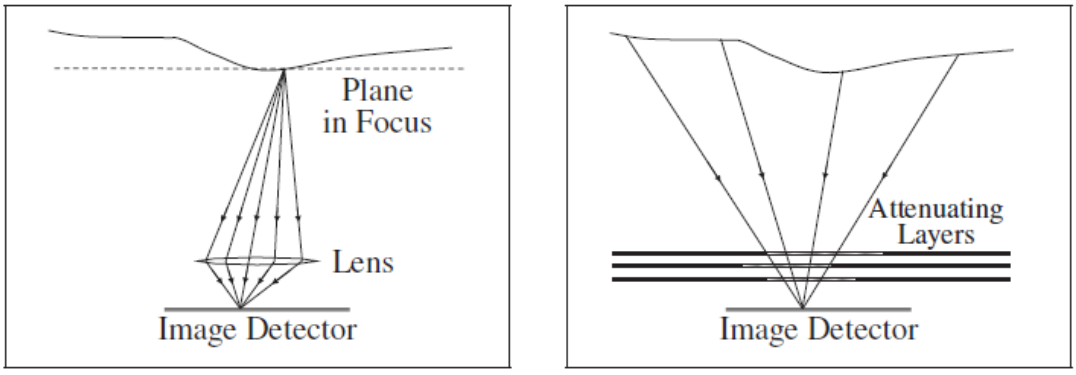


Figure 3.4: Comparison between the lens-based camera and lensless-based camera with controllable aperture

[4]

Each image point is modulated by attenuating aperture which defines the geometry and photometry of the imaging process. Architecture of this design is very simple. It consists of an image detector and a special aperture which is a flat attenuator whose transmittances are controllable in space and in time. (A constant binary transmittance pattern can be used to simulate a conventional aperture.) This architecture requires the attenuating layers (or in the simplest for a single-layer) to be controllable, which is achieved by using liquid crystal sheets, putting into different distances from the detector.

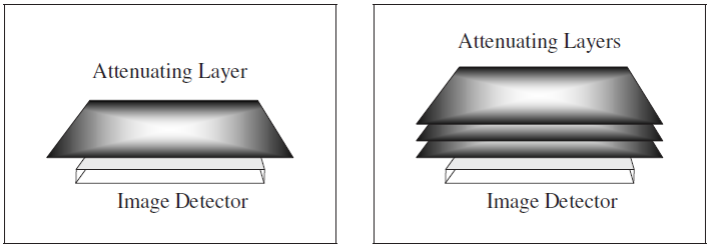


Figure 3.5: lensless-based camera architecture with controllable aperture

[4]

Imaging without lenses

Let the scene be far enough from the camera so we can show the difference between imaging with standard aperture and imaging with multilayered aperture. The picture 3.3 shows a set of K parallel flat layers at distances $f_1..f_K$ from the image plane. The image is taken by the image detector. Let $S_f(u)$ be the image of an ideal (diffraction-free) pinhole camera with a pinhole at

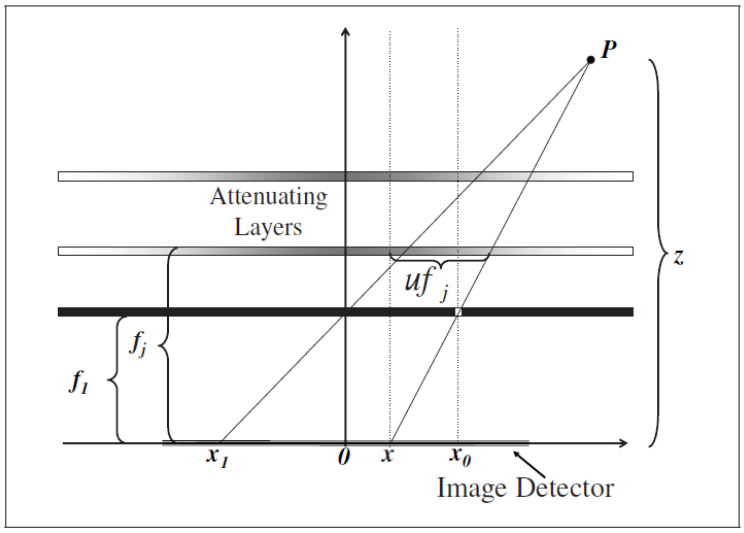


Figure 3.6: the scene-to-image mappings
[4]

distance f_1 from the centre of the image detector, then we can write an convolution [4]

$$I(x) = \int T(x-u)S_f(u)du \tag{3.1}$$

Specially, a shifted ideal pinhole corresponds to a shift of S_f as shown :

$$I(x) = \int \delta(x-u-d)S_f(u)du \tag{3.2}$$

where δ represents Dirac's delta function. For detailed description see [4].

An controllable Single-Layer Aperture is constructed as a controllable pinhole camera which transmittance pattern corresponds to a pinhole disk. It has the ability to control the aperture in space and time. That means the ability to shift the pinhole to any arbitrary location on the aperture instantaneously as discribed 3.2. Another usage of this architecture is an computational sensor witch makes an convolution of the scene with pre-defined patterns. This can be used for object detection tasks.

3.3.1 Multi-layer aperture

An controllable Multi-Layer Aperture can produce spatially varying scene-to-image mappings. This approach can be used for example for monitoring systems. By using more pinholes the scene can be split into sub-images. Since the camera is programmable, we can select which parts of the scene are captured. Once the region of interest is chosen, it can be captured with higher resolution than can be provided by a standard camera, because the latter needs to capture whole field of view.

Since this architecture can dynamically change the transmittance pattern of the aperture according to the motion in the scene we can use the object tracking algorithm for detecting the regions of interest.

Another usage of this architecture is a face detection as shown in 3.3.1. In this application, we can take advantage of the ability to partition the field of view into sub-images. For example, we can capture a part of the scene with a pinhole into top sub-image and put the result of the optical normalized correlation of the top part of the scene with the face template (or a convolution with the flipped template) into the bottom sub-image. A big advantage of performing the correlation in this way is alleviating computational load on the main processing computer, since part of the task is computed optically. When using only a single template, the face detection is not accurate and robust enough, as can be seen from the example image (black empty boxes represent false positives). Furthermore using single template works only for certain distance range from the camera. For detecting faces over lager area, multiple templates are needed.[4]

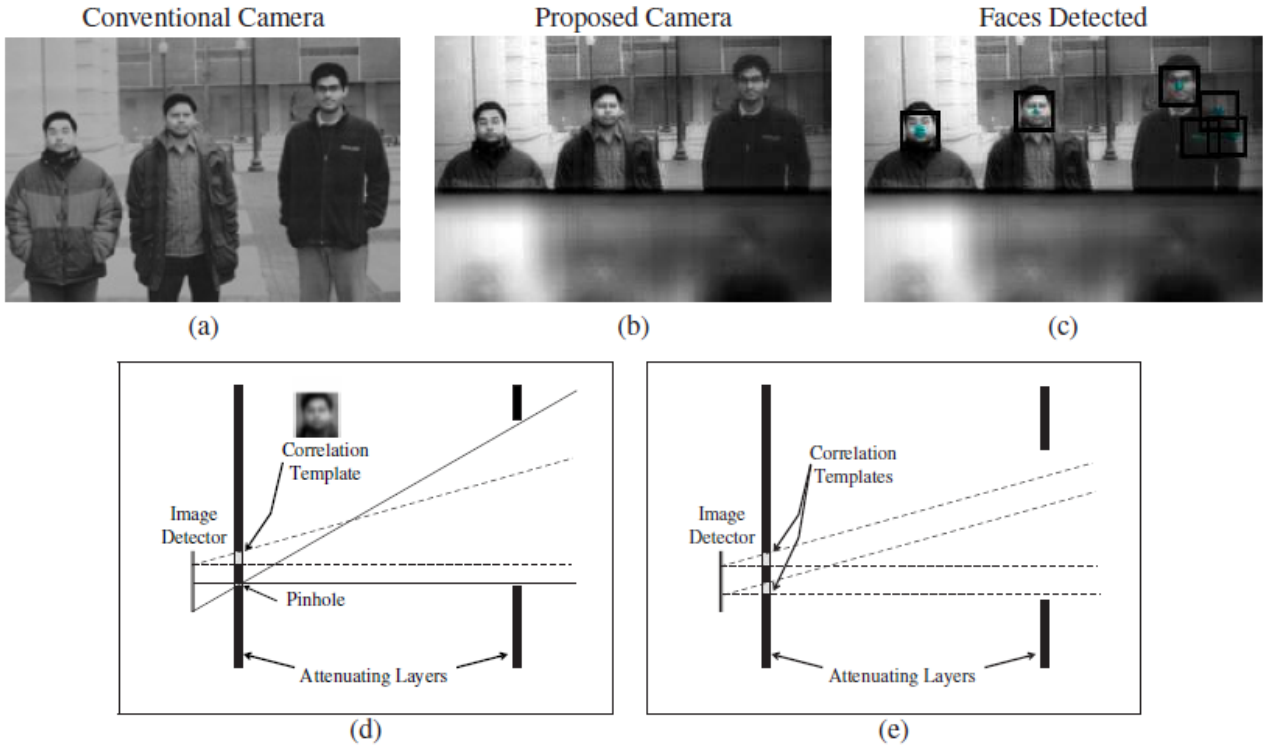


Figure 3.7: Optical face detection. (a) image taken by conventional camera, (b) same scene taken by lensless camera architecture,(c)the detected faces also with some false detections, (d)architecture of the attenuating layers and their transmittance patterns, (e)two convolutions are applied simultaneously to half the field of view. The camera can compute convolutions of multiple templates. [4]

Chapter 4

Compressive Sensing Reconstruction

The result of the compressive sensing process is a vector of measurements $y = \Phi s$ (Φ being the measurement matrix and s being the original measured vector). Reconstructing the original image (or, more precisely, the sparse representation of the original image, e.g. its Fourier transform) from these measurements is an optimization task which has as its input the vector of measured values y, measurement matrix which was used to obtain these measurements Φ, and tries to find the original measured vector s (or its approximation $s' \approx s$), which satisfies $\Phi s' \approx y$ and is sparse.

The approximation of s is now in Fourier domain. For getting a reconstructed original vector (original image), it is neccessary to multiply it with Θ (therefore to perform inverse Fourier transform) to get x.

The first reconstruction algorithm developed was the solution to ℓ_1 minimization, however, many alternative algorithms have been found which enable faster and more precise reconstruction. Any practical reconstruction algorithm should have all of the following properties [5]:

- It should accept samples from a variety of sampling schemes.
- It should succeed using a minimal number of samples.
- It should be robust when samples are contaminated with noise.
- It should provide optimal error guarantees for every target signal.
- It should offer provably efficient resource usage.

In this paper we use the CoSaMP (COmpressive SAmpling Matching Pursuit) algorithm published in [5]. It is an application of the more general *Matching Pursuit Algorithm* to the compressive sensing reconstruction problem.

For understanding the principle of it, we should know how the Matching Pursuit algorithm works. Matching pursuit, first introduced in [9], is a class of iterative algorithms which involve finding the "best matching" projections of multidimensional data onto an over-complete dictionary

D, which is a set of functions. Basic principle can be described as:

$$f(t) = \sum_{n=0}^{+\infty} \alpha_n g_{\gamma n}(t) \tag{4.1}$$

where α_n are coefficients and $g_{\gamma n}(t)$ are so called *atoms* (elements of dictionary of functions).

For example, consider the Fourier series representation of a signal. The dictionary is made from sinusoidal basis functions (the smallest possible complete dictionary). Finding a representation where most of the coefficients in the sum are close to 0 (sparse representation) is desirable for signal coding and compression.

Matching pursuit will first find the one atom from the dictionary that has the biggest inner product with the signal, then subtract the contribution due to that atom, and repeat the process until the signal is decomposed and the process reaches setted tolerance. More precisely, in every iteration of the algorithm, it generates for any signal f and any dictionary D a sorted list of indices and scalars which are the sub-optimal solution to the problem of sparse signal representation. The residual after calculating γ_n and α_n is denoted by $R_n + 1$ [10],[11],[12].

Matching Pursuit algorithm can be summarized as follows [6] [7],:

Input:
signal $f(t)$, matrix A (lines of dictionary D), tolerance T
stopping criterion e.g. until a level of accuracy is reached (tolerance)

Output:
approximation vector: (α_n) (list of coefficients)

Initialization:
$R_1 \leftarrow f(t)$
$n \leftarrow 1$

repeat
find $g_{\gamma n} \in D$ with maximum inner products $|\langle R_n, g_{\gamma n} \rangle|$
$a_n \leftarrow \langle R_n, g_\gamma n \rangle$
$R_n - 1 \leftarrow R_n - a_n g_{\gamma n}$
$n \leftarrow n + 1$
until $\|R_n\| < T$ **or** $\|R_n - R_{n-1}\| < T$

where $\|R_n\|$ is an error which decreases exponentially in each step and " $\leftarrow$" is a shorthand for "changes to" (for instance, "largest $\leftarrow$ item" means that the value of largest changes to the value of item).

The improvement of MP is Orthogonal Matching pursuit (OMP). The main difference from the MP is that rather than simply taking the scalar product of the residual and the new dictionary item to get the coefficient weight, we fit the original function to all the already selected dictionary items via least squares or projecting the function orthogonally onto all selected dictionary atoms, hence the term orthogonal matching pursuit. ([10],[9],[11]). This can lead to better results than standard MP, but requires more computation.

4.1 CoSaMP (COmpressive SAmplingMatching Pursuit) algorithm

CoSaMP (COmpressive SamplingMatching Pursuit) algorithm published in [5, 6] is a variant of Orthogonal Matching Pursuit (OMP) – it is its extension and differs from OMP in tighter bounds and its convergence and performance.

Let $\phi \in \mathbb{R}^{\mathbf{m}} \times \mathbf{n}$ be a measurement matrix that obeys the restricted isometry property of order 2s obeying $\delta_2 s \leq c$. Let $y = \phi \cdot \mathbf{x}$ be a measurement of the signal $x \in \mathbb{R}^N$ with error $e \in \mathbb{R}^N$ of arbitrary noise. Then for a given precision parameter η, the CoSaMP algorithm, produces an s-sparse vector $\hat{x}$ that satisfies

$$\|x - \hat{x}\|_2 \leq C \cdot max\{\eta, \frac{1}{\sqrt{s}}\|x - x^s/2\|_1 + \|e\|_2\} \tag{4.2}$$

, where $x^s/2$ is a best $s/2 - sparse$ approximation to x. Furthermore the algorithm has running time

$$O(L \cdot (\frac{x_2}{\eta})) \tag{4.3}$$

where L is the cost of a matrix multiply with$\mathbf{\Phi}$ or $\mathbf{\Phi^T}$.

As input, the CoSaMP algorithm requires four pieces of information [5]:

- Access to the sampling operator via matrix vector multiplication.
- A vector of (noisy) samples of the unknown signal.
- The sparsity of the approximation to be produced.
- A halting criterion.

These steps are repeated until the halting criterion is triggered.
The CoSaMP algorithm consists of five main steps [6],[9].

- Identification: Finds the largest 2s components of the signal proxy.

- Support Merge: Merges the support of the signal proxy with the support of the solution from the previous iteration.
- Estimation: Estimates a solution via least squares with the constraint that the solution lies on a particular support
- Pruning: Takes the solution estimate and compresses it to the required support.
- Sample Update: Updates the "sample", namely the residual in F-space.

CoSaMP(Φ, y,s) algorithm can be summarized as follows:

Input: Sampling matrix Φ, noisy sample vector y, sparsity level s,tolerance$T$

Output: An s-sparse approximation a of the target signal

Initialization:

$a^0 \leftarrow 0$ { Trivial initial approximation }
$v \leftarrow y$ {Current samples = input samples }
$k \leftarrow 0$

repeat

$k \leftarrow k + 1$

$y \leftarrow \Phi * v$ { Form signal proxy }
$\Omega \leftarrow supp(u2s)$ {Identify large components }
$T \leftarrow \Omega \cup suppa^(k-1)$ {Merge supports}

$bj|T \leftarrow \Phi_T u$ {Signal estimation by least-squares }
$bj|T^c \leftarrow 0$

$a^k \leftarrow b_s$ {Prune to obtain next approximation }
$v \leftarrow y - \Phi a^k$ { Update current samples }

until $\|R_n\| < T$ **or** $\|R_n - R_{n-1}\| < T$

where $\|R_n\|$ is an error which decreases exponentially in each step and " $\leftarrow''$ is a shorthand for "changes to" (for instance, "largest $\leftarrow$ item" means that the value of largest changes to the value of item).

According to paper [5],let's see these assumptions:

- The sparsity level s is fixed.

- The $m \times N$ sampling operator Φ has restricted isometry constant $|delta_(4s) \leq 0 : 1$.
- The signal $x \in \mathbb{C}^N$ is arbitrary, except where noted.
- The noise vector $e \in \mathbb{C}^m$ is arbitrary.
- The vector of samples $u = \Phi x + e$.

4.2 Simulating CoSaMP algorithm

For verification of CoSaMP algorithm above and theory of compressive sensing, I performed simulation in the Matlab computing environment. For recovering the image I used the CoSaMP reconstruction algorithm implemented in 4. Matlab source code of the simulation is available in the Appendix 6.1 .

Whole simulation works with grayscale images only, since the physical process we are simulating uses only one light detector. Input images are therefore transformed into grayscale before continuing with the simulation. Images are represented as matrices of floating point numbers with double precision. Range of values (values representing black and white) is not important, because the whole process is linear and we normalize the output image before displaying it.

4.2.1 Simulation process

The simulation requires three input parameters:

- Input image file
- Number of measurements to be performed on the input image
- Number of coefficients to be recovered during the image reconstruction phase – s. Therefore, the output vector will be s-sparse. Due to the limitations of the CoSaMP implementation used in the simulation, this value has to be at most one third of the number of measurements.

The simulation works as follows. Input image is loaded from the specified file as a matrix and then serialized into a column vector x (length of x being $width \times height$ of the original image).

Then a random matrix Φ is generated – its entries are generated using pseudorandom number generator with uniform distribution on an open interval $(0, 1)$, which satisfies RIP condition as described in section 2.4. The width of Φ is the number of pixels in the input image ($width \times height$) and the height of Φ is the number of measurements specified as one of the input parameters.

The random matrix Φ is then multiplied by column vector of the input image x and the result is stored into a row vector y, therefore $y = \Phi x$. This is how the taking measurements from image is simulated. The height of the vector y is the number of measurements. In the physical measurement process, rows of Φ are displayed sequentially onto the light modulator and elements of y are measured sequentially as well; in the simulation we perform this process all at once as a matrix multiplication.

At this point, the simulation and physical process converge, since now we have a vector of measurements y and we run the CoSaMP algorithm in order to reconstruct the original vector x (therefore the original input image). In order to reconstruct the image, we need to multiply the Φ by the DCT base matrix Ψ, computing the matrix $\Theta = \Phi\Psi$. The DCT base matrix is computed by performing DCT transforms on a set of base elements and stacking the results together as columns of Ψ, as described in section 2.5.

The implementation of CoSaMP we used (from [18]) takes these input parameters:

- reconstruction matrix Θ
- vector of measurements y
- required number of non-zero coefficients of the result s

CoSaMP implementation tries to find the vector S_{est} which tries to satisfy $y = \Theta \cdot S_{est}$ and sparsity requirement (i.e. having at most s non-zero elements). Of course in real system it's not possible to find exact solution, so the equation will not hold exactly and there will be some residual reconstruction error $y = \Theta \cdot S_{est} + R_n$. The vector S_{est} corresponds to the DCT of the original image and $R_n = y - \Theta \cdot S_{est}$ is estimation residuum.

The CoSaMP implementation we used outputs number of iterations and reconstruction error along with the result vector S_{est}. To get the original image, we first calculate the estimation of the original vector x by computing $x_{rec} = \Psi \cdot S_{est}$, which is essentially inverse DCT transform. We then rearrange elements of x_{rec} into matrix with dimensions of the original picture, producing the final output of the simulation.

4.2.2 Performance considerations

Performing the reconstruction of the original image entails two computationally intensive tasks: multiplying by the DCT base matrix Ψ and the CoSaMP algorithm itself. Also, storing some matrices in the process posed a critical problem because they were so demanding on operational memory that we had to find a way of making the calculations without storing the whole matrix in memory.

Intermediary matrix caching

As a part of the reconstruction process, the matrix $\Theta = \Phi \cdot \Psi$ needs to be computed. This is very time demanding task, because the number of elements of Ψ grows with fourth power of the size of

the input image (see 4.2.2) and number of elements of Ψ is $width \cdot height \cdot N$ where N is the number of measurements. Even with various optimizations used by Matlab environment, multiplying these matrices takes most of the time of the whole simulation process (in our case, the time it took was about 90–97 % of the whole computation).

There is no way to speed up this computation, but we can take the advantage of the fact that neither Ψ nor Φ depend in any way on the input image itself (Φ contains random data and Ψ serialized DCT base vectors of particular size). The only thing that influences their computation are the dimensions of the input image (which we can expect to be constant in practice) and the number of measurements. Therefore, once we compute the matrices Φ and Θ, we can cache them for later use in subsequent images of the same size and for the same number of measurements.

In our simulation, we use Matlab functionality to save matrices into file which we name according to the dimensions of the input image and the number of measurements. We first check for the existence of this file and load the matrices from it in case it exists, otherwise (when the simulation is ran for the first time) we compute and store them. As already mentioned, these matrices are very large and even though Matlab uses compression when storing the file, the size can be in order of hundreds of megabytes for small input images used in our simulation. However, storage is relatively cheap and in real system we would expect that only few of these files would need to be stored (input image size would be constant, therefore one file for each number of measurements).

DCT matrix multiplication

Even for small input images, the DCT base matrix Ψ has substantial size, as it is a square matrix with $width \times height$ elements in each dimension, therefore having $(width \times height)^2$ elements, proportional to fourth power of the linear size of the input image. This is very demanding for operational memory, because even for 100×100 image the matrix has 10^8 elements. When using floating point numbers in double precision (having 8 bytes according to the IEEE 754 standard), the matrix Ψ requires about 800 MB of operational memory.

Since we are generating the Ψ matrix for the sole purpose of using it in multiplications (when computing $\Theta = \Phi \cdot \Psi$ and when computing the final image from the reconstruction $x_{rec} = \Psi \cdot S_{est}$). Furthermore, in both multiplications the matrix we are multiplying by is relatively small compared to Ψ, so both one of the operands and the result do not require substantial memory for storage, the only problematic matrix is Ψ itself.

We have solved this issue by implementing methods which are performing the multiplication by the matrix Ψ on the fly, meaning they generate the elements of Ψ only as they are needed in the computation, not storing them in memory. Since we need to be able to multiply by Ψ both from left-hand side and right-hand side, we had to implement two methods which differ in their internal workings (for details see the source code in the Appendix 6.1).

4.2.3 Simulation results

In the followings pictures, there are results of running CoSaMP reconstruction algorithm on simulated data. Variations of images are used as inputs, such as landscape (6.1), portrait (4.2.3) and image of geometric structures (6.1) to see how MP behaves in different situations. Images of landscape and geometric structures can be found in Appendix A (6.1, 6.1).

Each set of pictures shows results for one input picture and various combinations of the input parameters. On top of each set, there is the original picture loaded to Matlab computing environment, and the results of CoSaMP reconstruction algorithm are underneath. This implementation also includes the information of how many iteration were necessary for reconstructing the image with the certain error. Comparison of these can be seen in table TABULKA CHYB

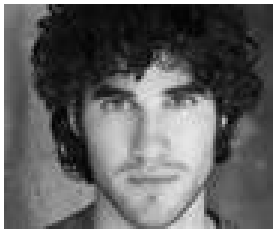
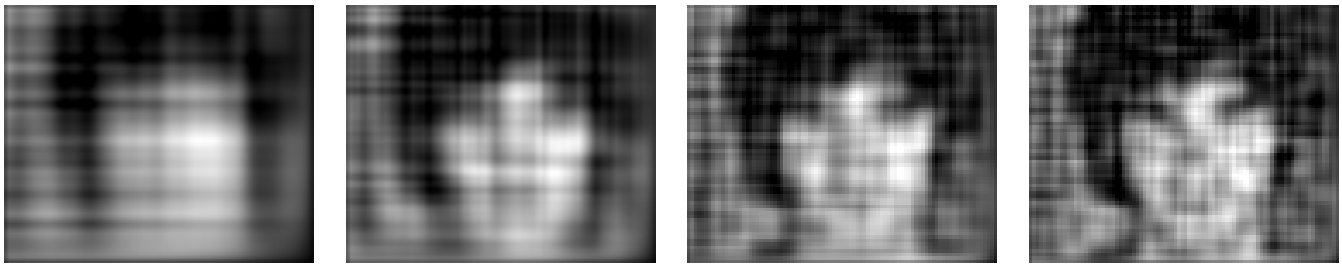
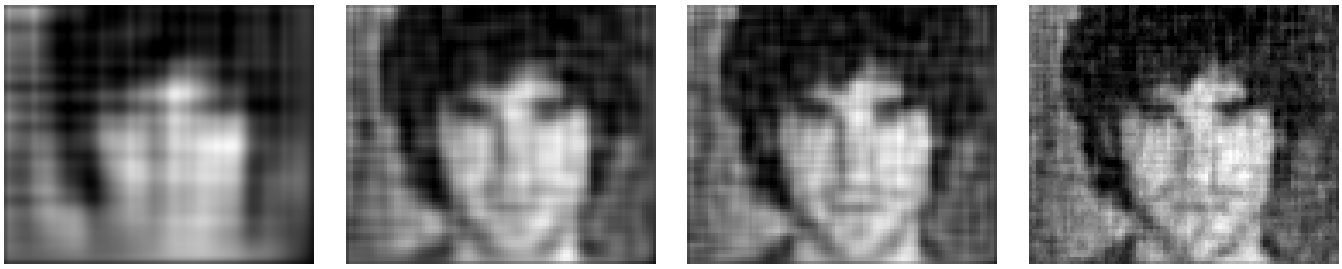


Figure 4.1: original picture of human face



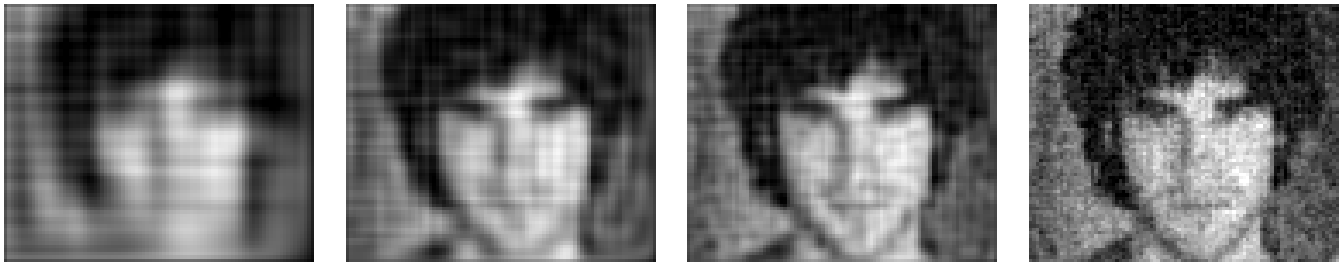
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 4.2: **1000 measurements** without noise



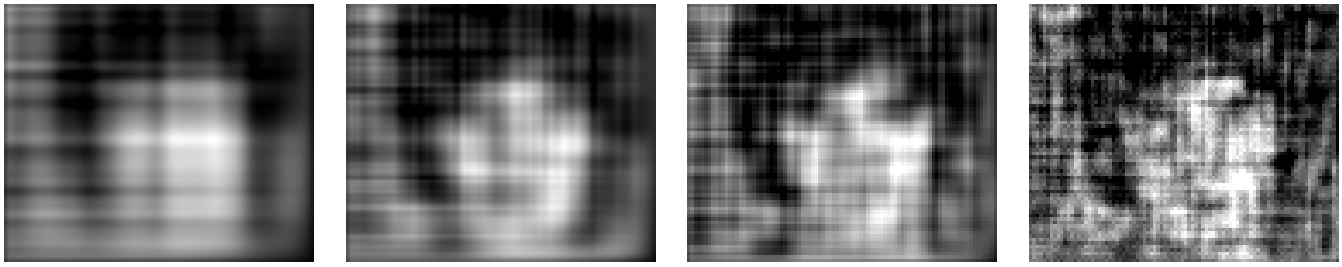
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 4.3: **2000 measurements** without noise



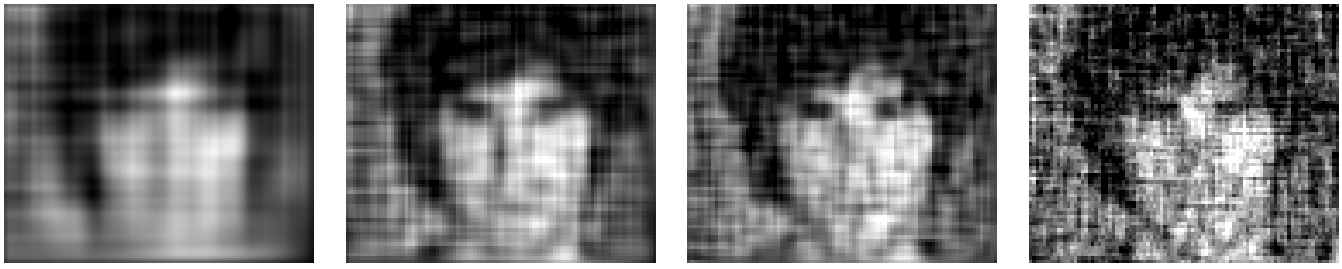
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 4.4: **3000 measurements** without noise



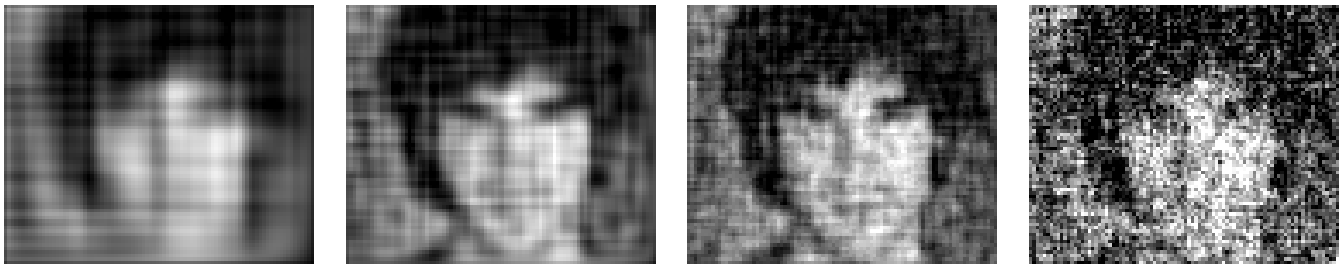
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 4.5: **1000 measurements** with added noise, k=500



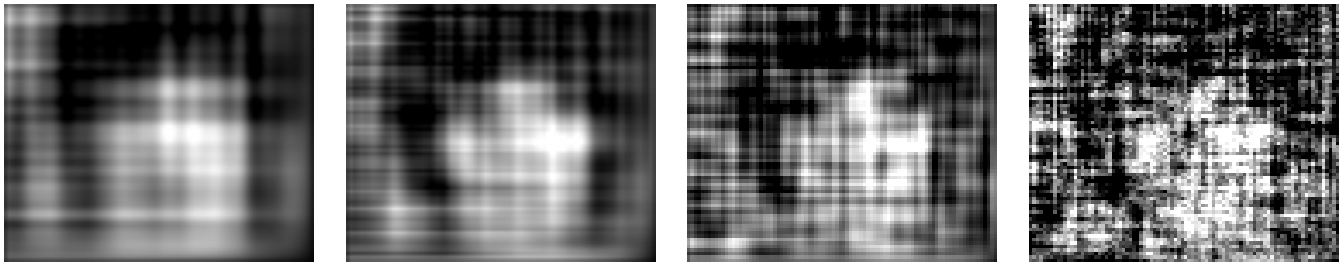
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 4.6: **2000 measurements** with added noise,k=500



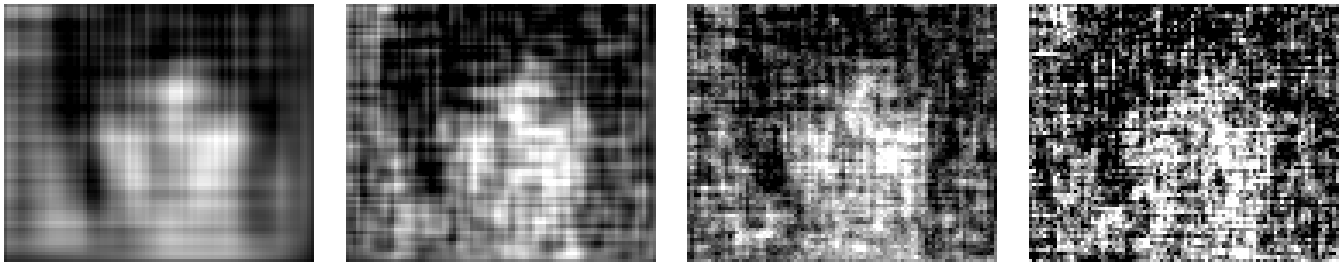
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 4.7: **3000 measurements** with added noise, k=500



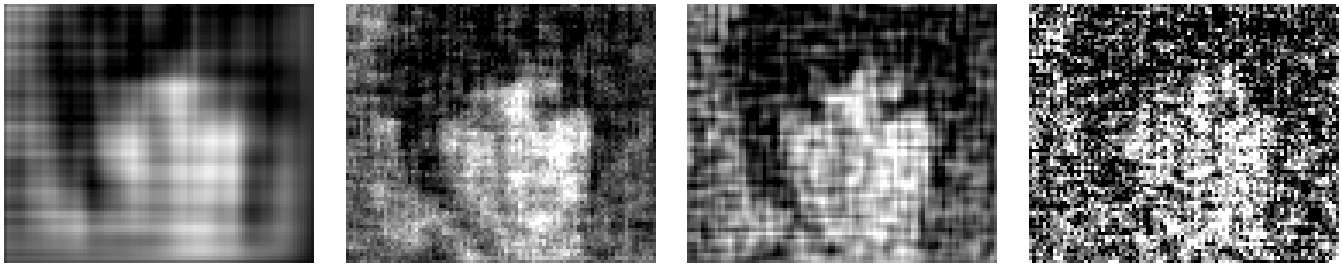
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 4.8: **1000 measurements** with added noise,k=1000



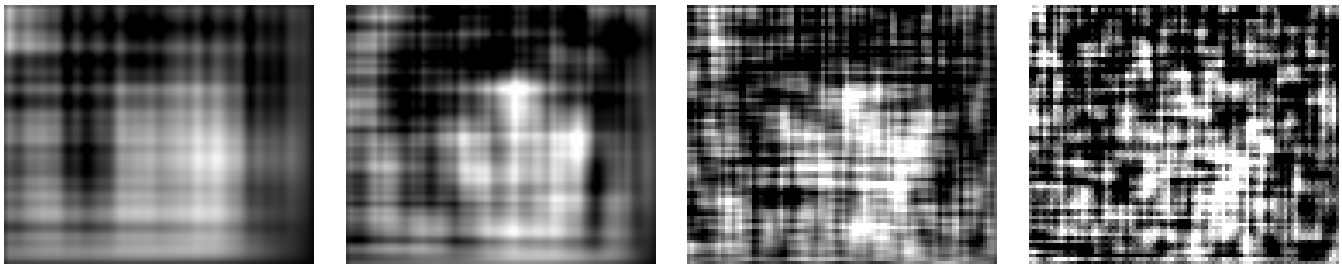
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 4.9: **2000 measurements** with added noise,k=1000



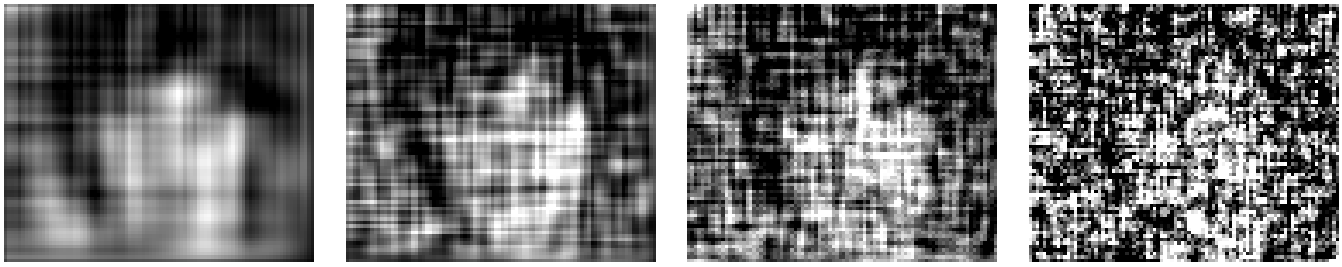
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 4.10: **3000 measurements** with added noise,k=100



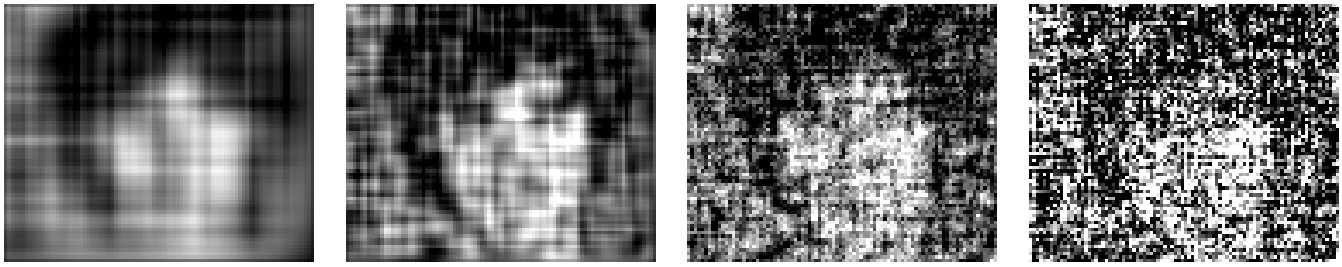
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 4.11: **1000 measurements** with added noise,k=1500



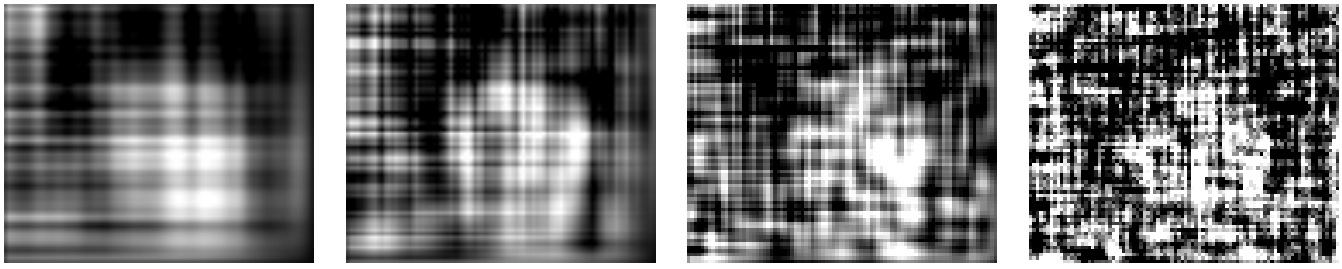
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 4.12: **2000 measurements** with added noise,k=1500



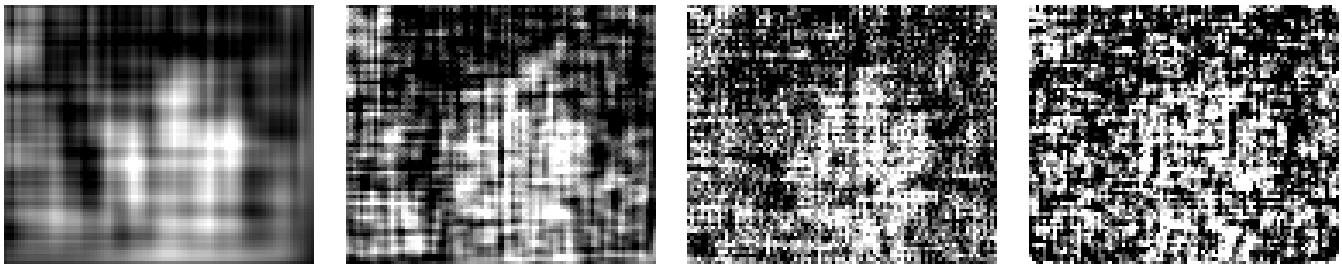
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 4.13: **3000 measurements** with added noise,k=1500



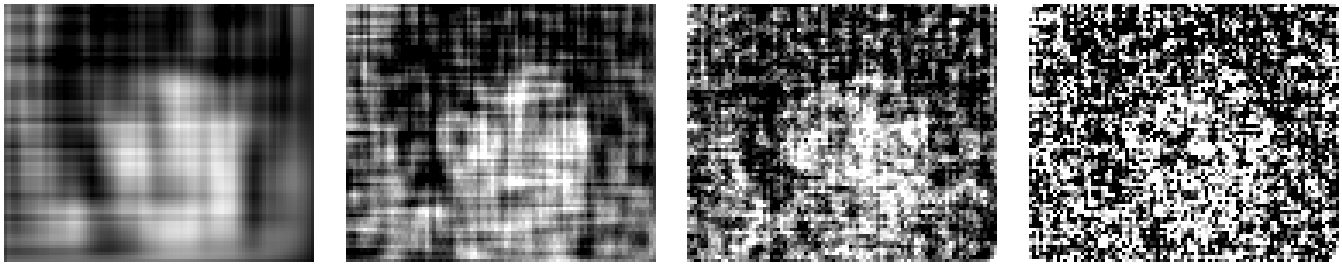
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 4.14: **1000 measurements** with added noise,k=2000



(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 4.15: **2000 measurements** with added noise,k=2000



(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 4.16: **3000 measurements** with added noise,k=2000

It is obvious that the reconstruction process has been successful. A size of input images are $90 \times 60 = 5400$, $90 \times 75 = 6750$ and $100 \times 60 = 6000$ pixels. If standard compression method were used, for example DCT, it would need 5400, 6750 and 6000 measurements for successful recovery. It is obvious that by increasing the number of measurements, the quality of the reconstructed picture is higher, but by using CoSaMP algorithm only a small number of measurements can be used and still the picture is recognizable.

4.2.4 Simulations with adding noise

In the previous chapter, the simulation of the CoSaMP was introduced. In the simulation, a noise-less measurement was assumed. In order to test the system for real-world scenarios, a random noise was added to measured values, without modifying the rest of the simulation process. The main source of noise in this construction is thermal noise of the light detector (photo-diode or CCD), which is normally modelled by using Poisson random distribution. Under standard conditions, Poisson distribution can be approximated by normal distribution [14], which is commonly used for image acquisition simulation purposes.

The adding noise was according to :

$$y = y + k \cdot n \tag{4.4}$$

where n is a vector of random numbers the same size as y and k is a constant (in our cases 500, 1000, 1500 and 2000) and y is an image vector. Random numbers are generated by a normal distribution with mean 0 and standard deviation 1.

After reconstructing the image, it was compared with the original image using three different comparison metrics:

Mean Square Error

The simplest metric is Mean Square Error (MSE), which is simply the second power of difference of values between the images, averaged over all pixels. Given a noise-free $m \times n$ monochrome image X and its noisy approximation Y, MSE is defined as:

$$MSE = \frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} (X(i,j) - Y(i,j))^2 \tag{4.5}$$

MSE of the simulation can be seen 4.2.4.

Peak signal-to-noise ratio

Peak signal-to-noise ratio (PSNR) is a ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. [17]

PSNR is most easily defined via the mean squared error (MSE):

$$PSNR_dB = 10 \cdot log_{10}((2^n - 1)^2) MSE \tag{4.6}$$

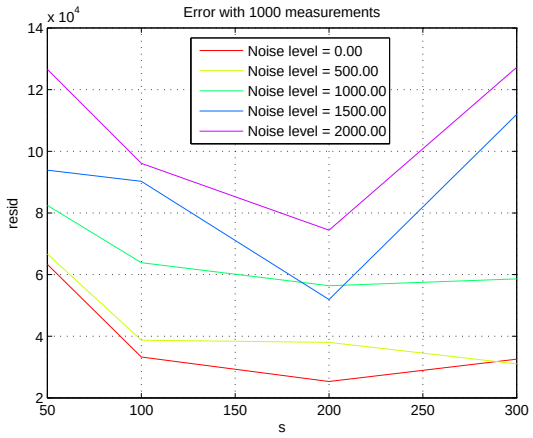
where n is a number of bits representing each value (in our case $n = 8$).

PSNR of the simulation can be seen 4.2.4.

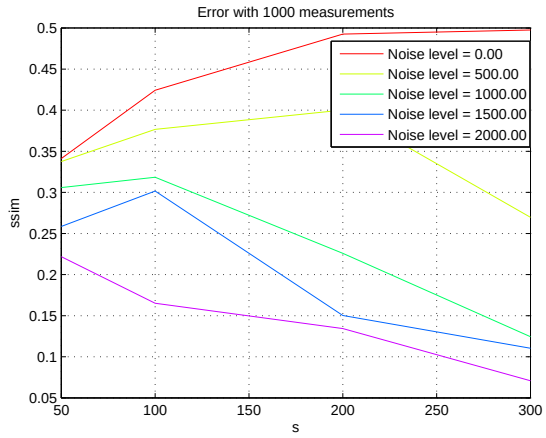
The Structural SIMilarity index

The Structural SIMilarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index can be viewed as a quality measure of one of the images being compared, provided the other image is regarded as of perfect quality. The SSIM aims to quantify the *perceived* quality degradation, with respect to properties of human vision, especially taking into account the fact that humans perceive differences in *structural* properties of the image the most. For example, slightly altering the overall brightness of the image would yield substantial MSE, but would not be perceived as a structural difference (with the exception of areas where saturation would occur), therefore the SSIM would be relatively small. On the other hand, blurring the image would not produce high level of MSE, but SSIM would raise dramatically, as the structure of the image would

be degraded. The full description of the SSIM algorithm is given in [15]. In this work 4.2.4, a Matlab implementation provided by the authors was used.

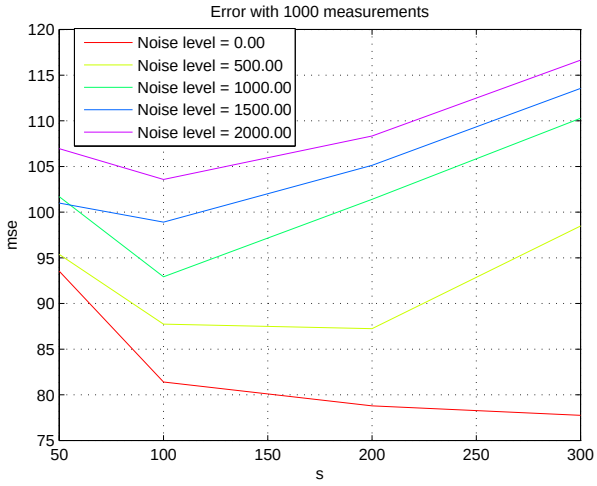


(a) Reconstruction error for 1000 measurements

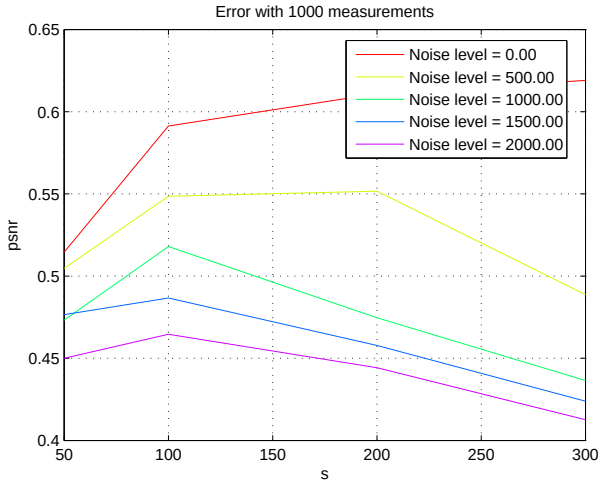


(b) SSIM for 1000 measurements

Figure 4.17: Reconstruction error and SSIM for 1000 measurements



(a) MSE for 1000 measurements



(b) PSNR for 1000 measurements

Figure 4.18: MSE and PSNR for 1000 measurements

For more detailed view, see , where you can find graphs for 2000 and 3000 measurements as well.

Chapter 5

Construction of the lensless-based camera using compressive sensing

Lets briefly describe measuring architecture. In the picture 5 a fundamental scheme of lensless image camera is introduced. We will be taking image of static models illuminated by external light. In the distance from the scene, there is placed a device consist of LC 2002 Transmissive Spatial Light Modulator (SLM), photo-diode used as a detector and an USB Power and Energy Meter Interface, which is able to store these measurements digitally to a PC for later processing. These devices are shown in figure 5.

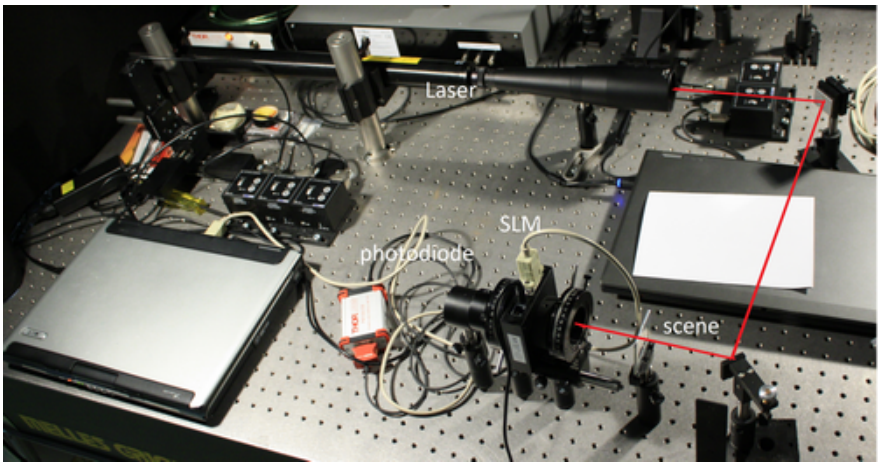


Figure 5.1: A fundamental scheme of lensless image camera using compressive sensing

5.1 Architecture

The key component of the whole system is the SLM modulator which is used as a controllable aperture through which the light from the scene enters the system. Its function is to have various degrees of transparency in specific places and in various times. It therefore behaves as an transparent

Figure 5.2: LC 2002 Transmissive SLM,photo-diode used as a detector and the USB Power and Energy Meter Interface

LCD display, provided the polarizing filters, which are added, because the SLM modulator doesn't contain them by default which you can see in the picture 5. The polarizing layers must be rotated relatively to each other by 90°. Resolution of the display is 800×600 pixels and its active area is $21.0 \times 26.0\,\text{mm}$. The input data are provided as a standard two dimensional picture through a VGA connector. The SLM modulator and a light detecting diode are connected through a tube, which prevents light from entering the system any other way than through the SLM modulator. The length of the tube (the distance from the aperture to the detector) defines the field angle of the captured scene.

5.2 Measuring process

Whole measuring process can be controlled by Matlab software, but there was no license available for Test and Measurement toolbox. When controlling the SLM modulator from the computer, it behaves like a second computer screen. So we controlled the measuring process by an ad-hoc software interface implemented in *Python*. The Python controlling software is also able to remove undesirable features, such as toolbars and borders, from the Matlab figure displayed on the device. This is essential because the aperture should contain assembly pattern which consist only of the computed matrix and no other features, which would distort the measuring process.

The photographed scene must be illuminated by external light. For the photographed scene we used templates with lines, squares, dots, letters, white paper or such which you can see it the picture 5.2.

A light source was placed close to the photographed scene. Afterwards, the whole image acquisition process is controlled through a Matlab script ran on a personal computer. It generates measuring matrices and calls Python software for displaying them on the SLM. Matrices are displayed sequentially and after displaying each of them, the measurement of the total light is acquired from the photo-diode. The result of the measurement process is therefore a set of measured values,

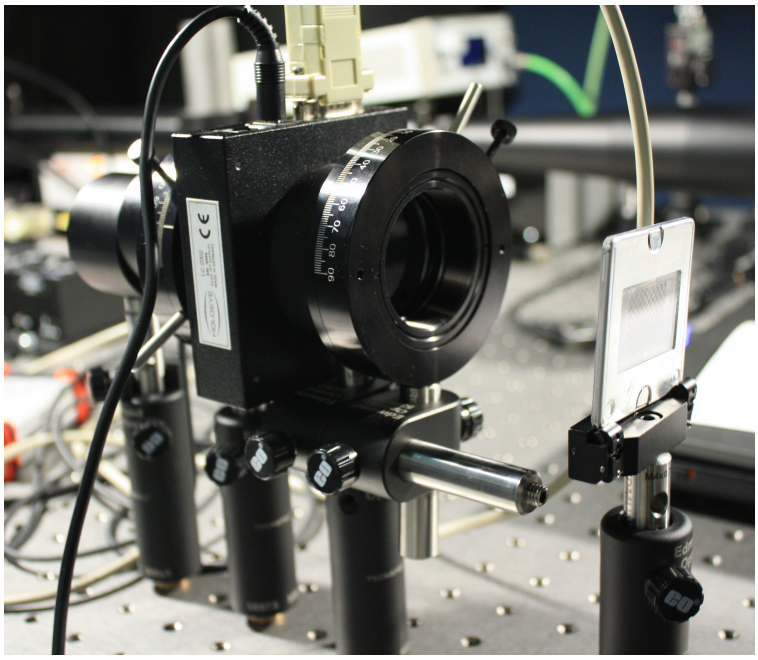


Figure 5.3: the detail of construction

one value for each measuring matrix. The measured values are then imported to Matlab software for subsequent evaluation.

5.2.1 Choosing illumination light

It appeared as a main problem using the proper illuminations by the external light. Initially, a fluorescent tube was used which proved to be bad solution, as only very small fraction of the light power was detected, under the resolution obtainable by the measuring chain (in the order of nanowatts). The fluorescent tube was replaced by a incandescent bulb. Even though the power detected by photo-diode raised substantially (hundreds of μW), the reconstructed image of a white paper showed visible lines caused by frequencies interferences caused probably by the room lightning where the fluorescent tubes are used, as shown in the picture 5.2.1. There is clearly visible horizontal lines represented by cca $10Hz$ caused by subtraction of the SLM modulator frequency ($60Hz$) and frequency of a bulb mixed with fluorescent tubes which was connected to an electric socket ($50Hz$) and verticals lines represented by frequency cca $110Hz$ caused by summation of SLM modulator frequency ($60Hz$) and frequency of a bulb mixed with fluorescent tube which was connected to an electric socket ($50Hz$).

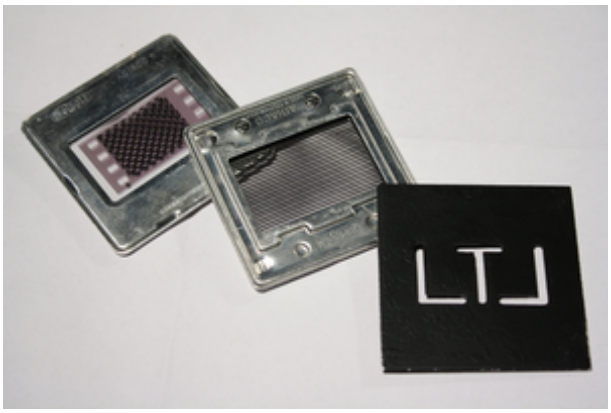


Figure 5.4: examples of photographed scene

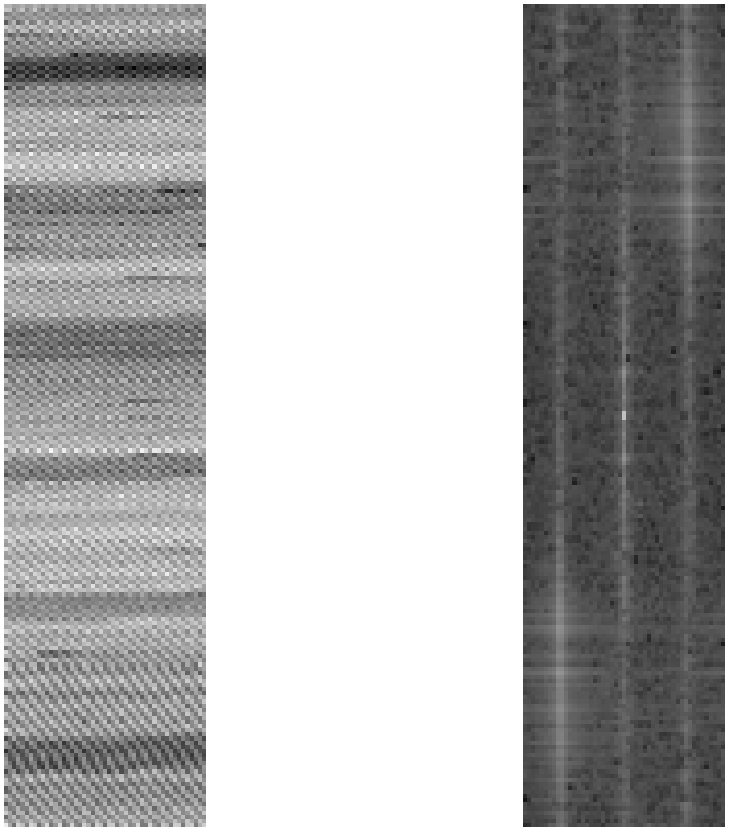


Figure 5.5: Pixel by pixel reconstruction, illuminated by a bulb with frequency interferences and its spectrum

That is why illumination by LED diode was chosen as an alternative. The diode is a non-directional type, but its illuminating power radiated in a direction effective for the measurement was still in order of μ W, which is above the resolution of the measurement chain and nears the maximum power output of the LED diode. In order to measure the maximum achievable power output of the photo-diode in this setting, the photographed scene was without any photographed

object and a white image was displayed on the SLM once in each 50 images, as was described previously. Results are on the picture 5.2.1

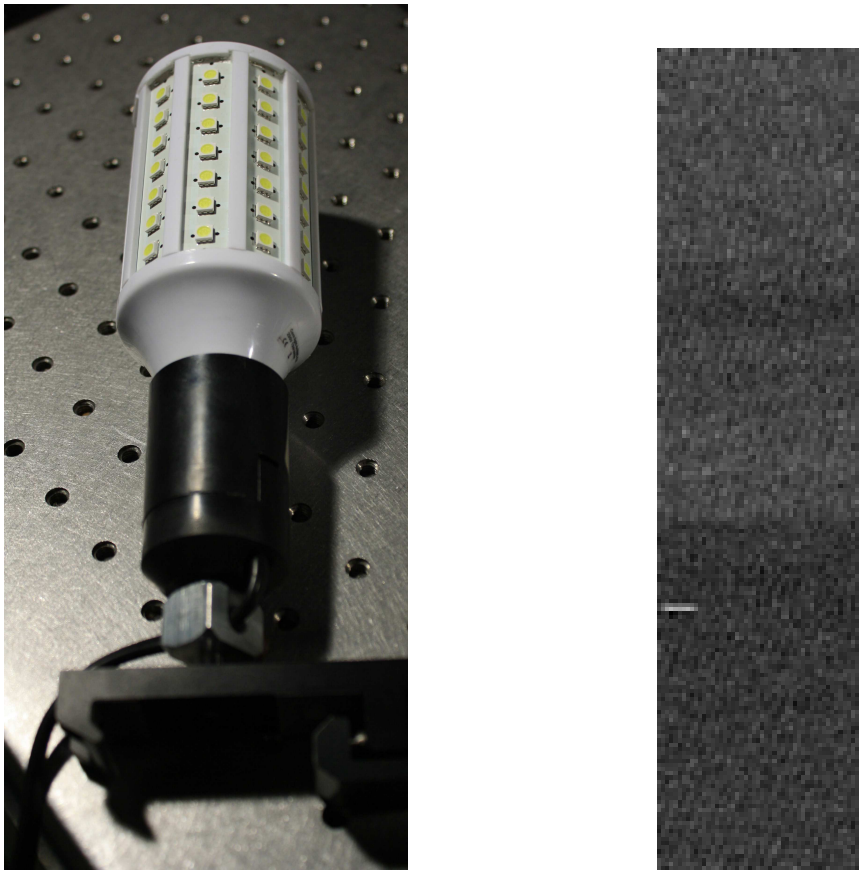


Figure 5.6: LED diodes reconstructed image and powers measured by System verification using pixel by pixel measurement added white image in each 50 images

As a next improvement, we used directional type of LED diode with illuminating power of 2.5 W. Spectrally, it has the highest peak around 500 nm. In the previous case we powered the LED diode from an electric socket (alternating current), so in this case we bulit a construction with a rectifier which provided a direct current. It was supposed to remove eventual blinking of the diode. Still the power detected by a photo-diode was in order of nW. You can see the construction of LED diode with a rectifier at picture 5.2.1

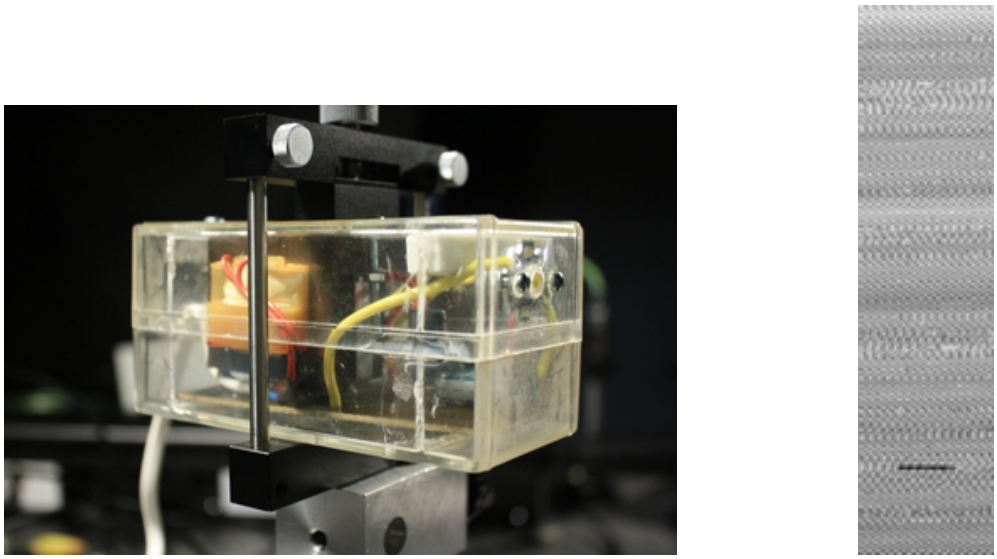


Figure 5.7: construction of LED diode with a rectifier and its result

We assumed, for successful detection and reconstruction it is needed to intensify the power of illuminating light. That is why laser (coherent light) was used. In the picture 5 you can see construction of lensless imaging process where laser was used. Because the laser itself has a narrow beam, the expander is used. It expands twenty times the beam which goes from the laser. Then, the light beam is delivered onto a scene and follows through the SLM onto a photo-diode. The laser takes place in a darken room so during the measuring no other light than the laser did not influenced the measuring process. The result of using laser beam you can see in the picture 5.3

In order to raise the power of illuminating light, a halogen bulb was used. The illustrative picture can be seen in the figure 5.2.1. By this illumination, we achieved the biggest power detected by photo-diode, but using this illumination the algorithm was not able to reconstruct the image. We assumed, that the construction presented in this paper probably needs a collimated light beam.

Figure 5.8: high power halogen bulb

5.3 System verification using pixel by pixel measurement

For initial verification of the functionality of the whole measuring chain, we used the scanning technique called *"Pixel by pixel"*. It is a trivial measuring method for testing the overall functionality of the system and it doesn't perform compressive sensing process. It works as follows: instead of generating random matrices, only black matrices (all zeroes) with exactly one white pixel (transparent) are used. The number of matrices is the same as the number of pixels of the measured picture (each pixel has one corresponding matrix with that particular pixel white and other pixels black), so the result of the measuring process is one measurement for each pixel.

During the measuring process, there must be a pause between displaying the image on the SLM and the actual measurement of the light level. Otherwise there might be a chance that the photo-diode measured value for a wrong image or for some intermediate transition state between subsequent images. However, making this pause too long would make the measurement process very time consuming. To experiment with various values for the length of the pause and to make sure that the diode is measuring the light levels for the proper image (therefore that the pause is long enough), a completely white image was displayed on the SLM once in each 50 generated images. Values from these test measurements should have the highest level of all measured values since in these cases the aperture admits as much light as possible into the system. This is apparent from the reconstructed image – the last column should be white, indicating that the time for loading the image and taking measurement by the photo-diode is suitable enough. The result can be seen in picture 5.3.

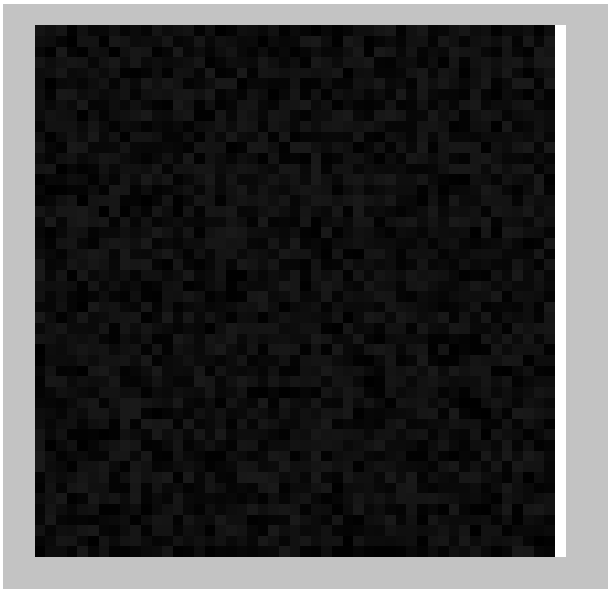


Figure 5.9: verification of measuring process

The result of an image reconstructed by a system using *pixel by pixel* can be seen in the picture 5.3. There you can see a visible letter **T**. The template which was used was consist of black material in which was extirpated a letter **T**. That is why the letter **T** is white and surroundings dark. The template was placed 8 cm from the SLM.

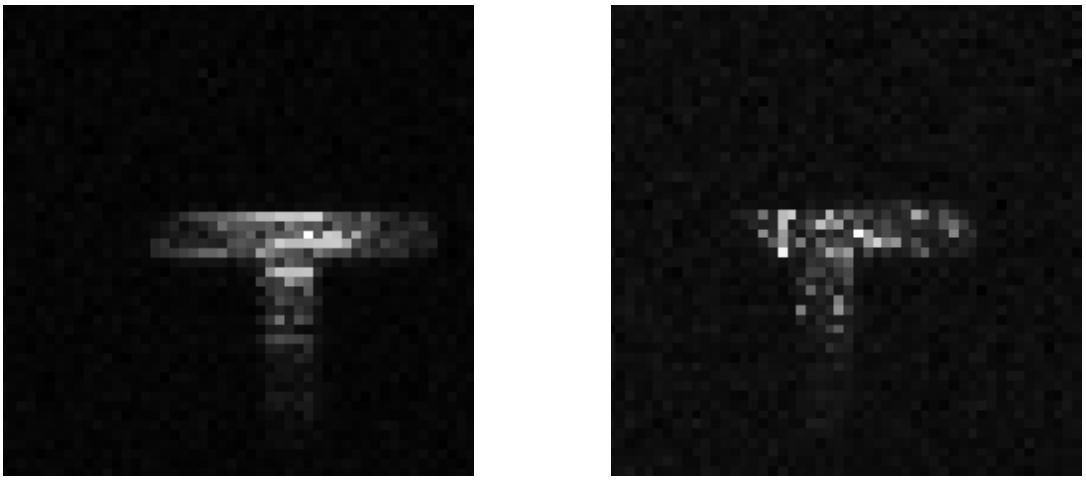


Figure 5.10: image taken by system verification pixel by pixel

However, this construction has its limitations. If the template pattern has fine details (like the first template in the first picture 5.2), the result is not satisfying – see picture 5.3. The fine details are lost and the edges are blurred.

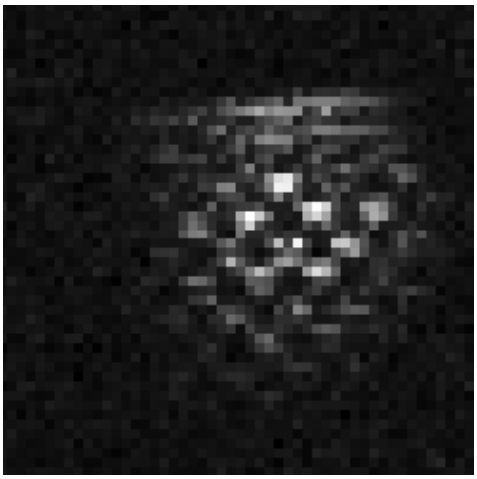


Figure 5.11: image of squares -first template by system verification pixel by pixel

5.4 System verification using combined pixel by pixel and CoSaMP algorithm

In the previous section 5.3, we verified that the construction is working. So we proceeded to image reconstruction via CoSaMP algorithm, but the resulting reconstructed image was noise as you can see in the picture 5.4.

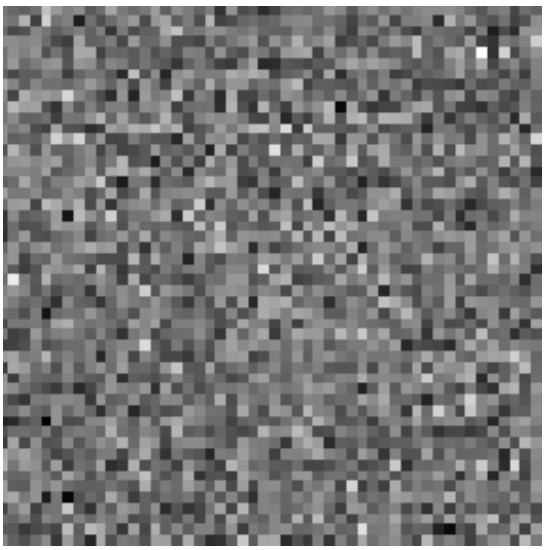


Figure 5.12: image taken by the presented construction and reconstructed by CoSaMP algorithm

In an attempt to troubleshoot the system, an verification input was created which combines a pixel by pixel method with CoSaMP algorithm. The aim of this was to localize the error in the measuring chain. It works as follows: The image is illuminated by laser light. The SLM modulator displays unit matrix and the result vector is used as a input for the CoSaMP algorithm. Therefore, we are performing CoSaMP algorithm but on a "trivial" input, with the same number of measurements as the number of pixels in the original image and the matrix Φ being unit matrix. This method was successful as you can see in the picture 5.4. In the picture you can see a form in the shape of a letter **T**.

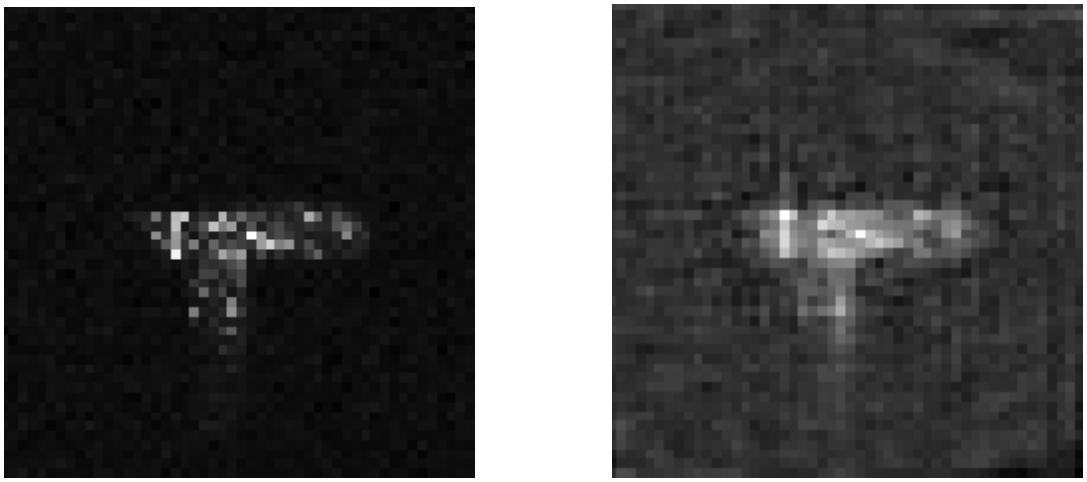


Figure 5.13: Comparison of images pixel by pixel and reconstructed by CoSaMP algorithm

It is obvious that taking image by this construction is working, also CoSaMP algorithm was able to reconstruct the image (which under standard conditions, would not have been necessary, since we have already measured all pixels). But as was mentioned before, using random matrices displayed on the SLM instead of black ones with only one transparent pixel.

Next natural idea was to not give CoSaMP complete set of measured pixels, but select only every second one (i.e. selecting every second measurement from the last example). This produces correct results as well (see figure 5.4), with the exception of every second row in the image which is not reconstructed, because the algorithm had no data for these pixels. The size of the picture was 50 pixels, so the number of measurements was 2500 and the number of sparse coefficients was 300.

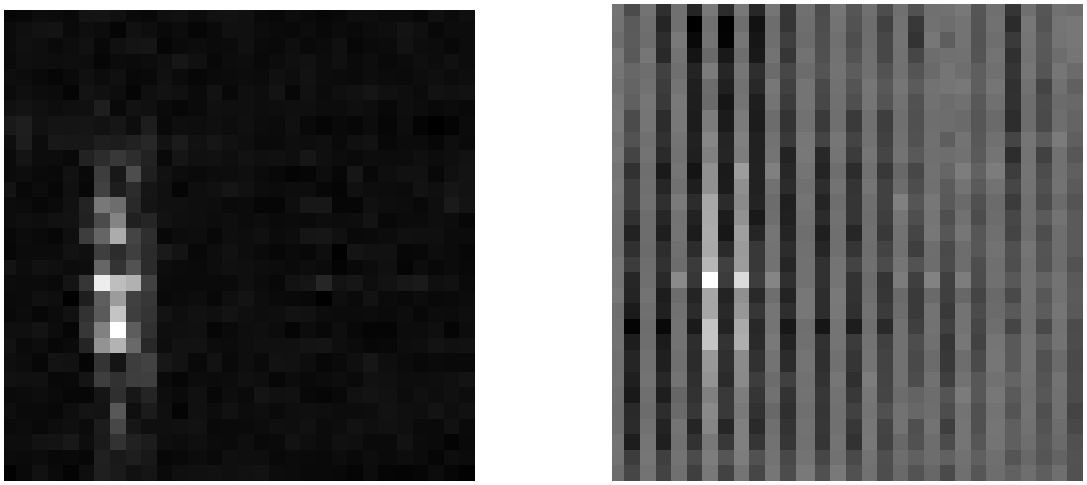


Figure 5.14: Comparison of images reconstructed by CoSaMP algorithm

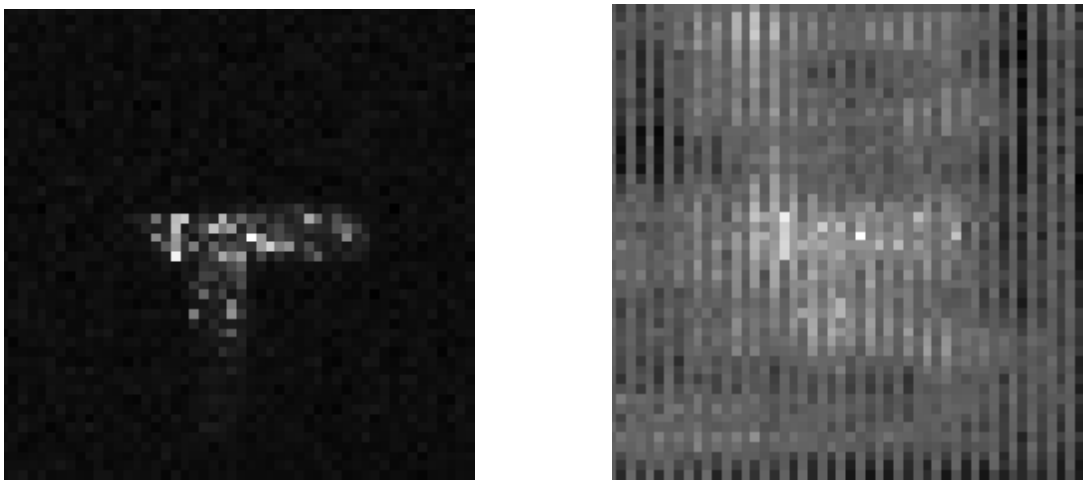


Figure 5.15: Comparison of images reconstructed by CoSaMP algorithm

The last idea was to use data from normal measurement (using random matrices on the modulator), but to normalize the input data which goes to the CoSaMP algorithm. The data taken by the system pixel by pixel was in range $1 \cdot 10^{-7}$, but in simulation of CoSaMP algorithm the input data was around $1 \cdot 10^{5}$, so the CoSaMP algorithm was ran after multiplying the measurement values by $1 \cdot 10^{11}$. The result can be seen in the picture 5.4 – it is apparent that this has not helped, which was a bit expected since CoSaMP is mainly linear process.

Figure 5.16: CoSaMP result after multiplying the imput by $1 \cdot 10^{11}$

5.5 Real system simulation

In order to make it possible to examine the behaviour of the physical system, the simulation process from the chapter 4 was adapted to take properties of the constructed physical system into

account. This was done by measuring Point Spread Function of the constructed system and then use this information in the simulation of the measurement process.

5.5.1 Point Spread Function (PSF) of the construction

Point Spread Function (PSF) is a system's impulse response, or in other words a projection of point light source. The point source is very easily imaginable and from mathematical perspective easy to analyse, however, in practice it is not feasible to simulate it accurately. A light source of given intensity and zero size is regarded as an ideal point light source. The source should emit light into all angles with the same intensity, which means uniform emission into the whole solid angle. Practical realization of point light sources is done either by small light-bulb, concentrated arc or by a bigger light source covered by iron board with small hole inside. As a result, these point sources emit uniform light intensity (W/cm^2) only in limited solid angles.

Ideal display system would depict the point light source on the image surface again as a point, in all other places the image surface would be irradiated with zero intensity. However, in real system, different influences can occur as for example dust, scratches, material imperfections of lenses (e.g. small air bubbles), disuniformities on the cover edge, diffraction of light rays caused by getting through an aperture or optical aberrations causing defocusing. Some of these influences are removable, however diffraction is for each aperture (especially in pin-hole or lens-less camera) essential constrain imaging system. By these effects, the lighting beams spreads a bit to the closest surroundings, where in the case of ideal imaging system only origin point would be displayed. If we put into this point the beginning of origin of coordinate system, we can construct a function $h(x, y)$ of two spatial variables x and y, which would give us for each coordinates an intensity of the illumination of that point. We would call this function Point Spread Function (PSF). It is an impulse response of the imaging system. In the picture 5.5.1, there is a typical shape of PSF known also as the Airy disc.

If we consider an unit light source coming from a point source going through an optical system and if we neglect the loses in the optical system, for PSF the following equation must hold[16]:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x, y)\mathrm{d}x\mathrm{d}y = 1 \tag{5.1}$$

Practically, we can measure PSF of the optical system by taking picture of a source of the light close to the point source by its parameters. The taken image would be equal to PSF response. The example of the image is shown in the picture 5.5.1.

The real measuring process is not measuring perfectly (influenced by limited resolution). Even if we had an ideal imaging system, PSF measured this way would not be displayed as a single point. In this case, we can consider the measuring process itself to have a non-ideal PSF and the final PSF

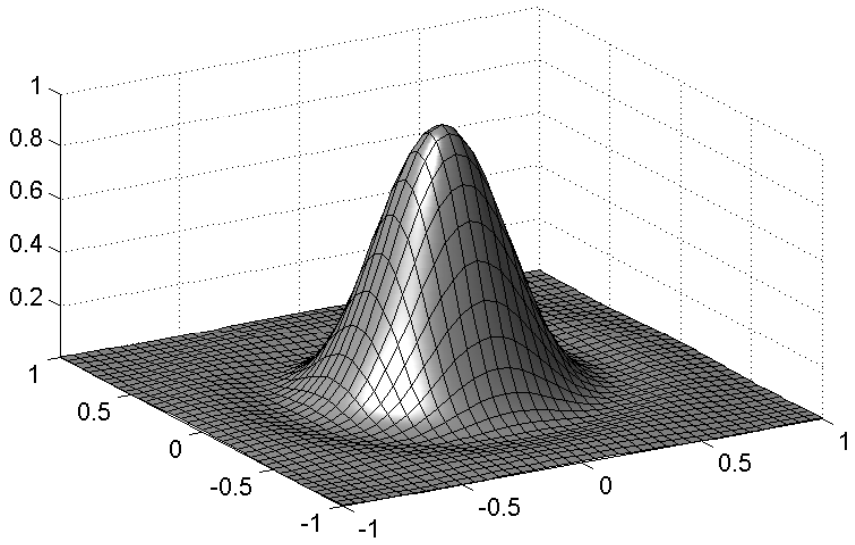


Figure 5.17: Example of PSF

would be a convolution of PSF of the optical system and PSF of measuring process:

$$h(x', y') = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y) h_f(x' - x, y' - y) \mathrm{d}x \mathrm{d}y, \tag{5.2}$$

where $h(x, y)$ is PSF of the optical system and $h_f(x, y)$ is PSF of the measuring process ([16]).

5.5.2 PSF of systems limited only by diffraction

Considering the optical system as an ideal one, without any optical aberrations and with a perfectly round aperture, the only thing causing light dispersion will be light diffraction (light curvature). PSF this optical system is shown in the picture:5.5.1 and is called *Airy disc*. Its mathematical formula is ([16]):

$$W(r) = \frac{W_{c0}[2J_1(u)]^2}{u^2} \tag{5.3}$$

where $W(r)$ represents an intensity of illumination in the distance r from the origin of coordinate system (point of intersection of optical system axis with the imaging plane), W_{c0} is intensity of illumination in the beginning, J_1 is a Bessel function of the first kind of first order and

$$u = \frac{2\pi r sin(\alpha)}{\lambda} = \frac{2\pi r \rho_m}{\lambda \cdot s'} \tag{5.4}$$

The meaning of others symbols are shown in the picture 5.5.2.

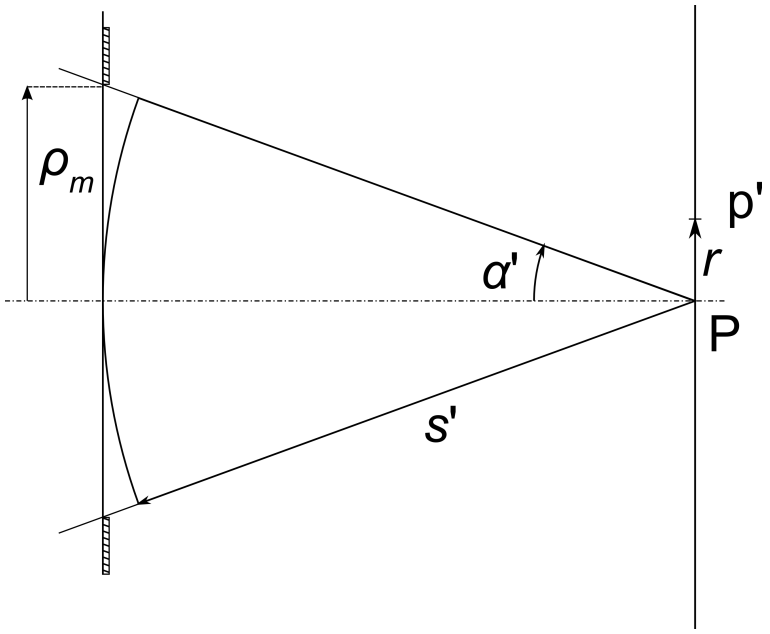


Figure 5.18: Spatial parameter of the imaging system [16]

Airy disc will display on the image plane as a white centred circle with concentric circles around it (see picture 5.5.2). Most of the light energy is directed to the centred circle – cca 84 % of the whole amount of a power stream. The remaining 16 % is spread out to side lobes. If the system is affected by aberrations or if the system is slightly diverged from the ideal, the difference is usually not noticeable in the size of the picture, but by relative distribution of light power between the centre circle and side circles. Usually, part of the light beam which originally illuminated the centre will be diffused to the edges of the picture. This difference is in order of tens of percent and will be hardy visually noticeable from PSF.

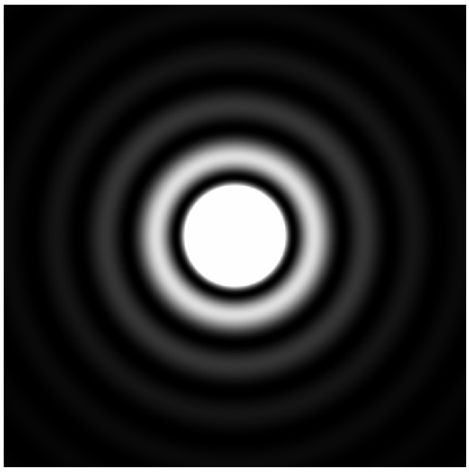


Figure 5.19: Airy disc displayed on the image plane. For clarification purposes, the image is lightened and the centred circle is thus saturated white in most of its area. (Generated in Matlab.)

5.6 Simulating PSF of the presented lensless camera

The knowledge of PSF of the real system is essential. It gives us an objective evaluation of how precisely a system can reproduce the input image. PSF shows how much the system blurs the image.

PSF was measured by using laser as a point light source and measuring the resulting image by pixel-by-pixel method. The picture 5.6 shows an PSF of the construction presented in this paper. The resolution of the SLM is 600×800 pixels, but in our system the effective resolution was reduced by a factor of fifty, therefore one pixel of the image is represented by 12 pixels on a SLM.

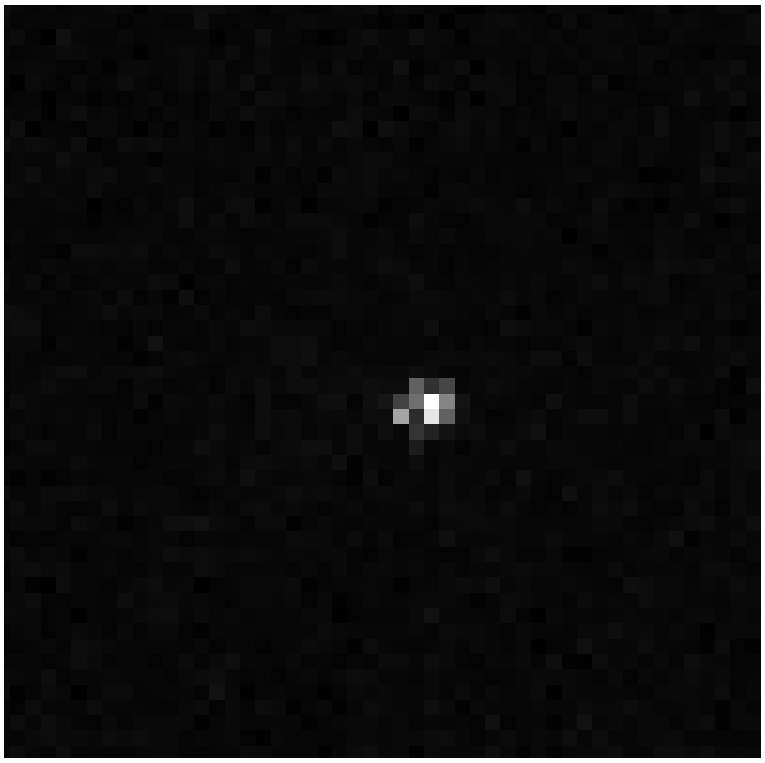
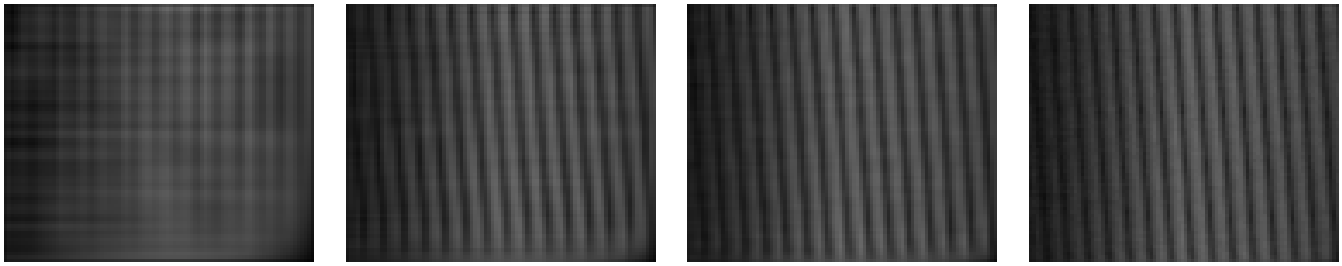


Figure 5.20: PSF of presented construction



(a) s = 100 (b) s = 300 (c) s = 500 (d) s = 1000

Figure 5.21: **3000 measurements** without noise

The measured PSF was integrated into the simulation process by modifying the Ψ matrix which is used for the simulation of the measurement process. In the conventional simulation, we have calculated the vector of measured values as $y = \Psi \cdot x$, whereas now we use modifies matrix Ψ' ($y = \Psi' \cdot x$), where Ψ' is calculated by successively taking each row of the original matrix Ψ, converting it back into 2D pattern displayed on the modulator, then calculating 2D convolution between this pattern and the measured PSF function resampled to the appropriate size, and then serializing the resulting matrix back into the corresponding row of Ψ'. Note that the original matrix Φ is still used when calculating the matrix Θ, therefore CoSaMP works with the original non-convoluted matrix, which is the case in the real system as well.

This modified simulation has been ran for some of the combinations of parameters used in the chapter 4. The results are displayed in the figure 5.6. It's clear that the algorithm is unable to reproduce the image in a satisfactory way. The blurring caused by the non-ideal PSF makes the algorithm unable to reproduce the detail contained in the original picture. It's possible that similar problems encountered in the real system have the same cause.

Chapter 6

Conclusion

New unconventional methods of image acquisition such as lensless imaging are a new promising field which reacts to pressure for more flexible, cheap and complex devices than conventional cameras.

Thanks to compressive sensing approach, it is not needed to sample all pixels. It has been proved that it is more efficient to employ random measurements and acquire only measurements which are needed. This fact is nicely seen in chapter 4.2. However, it has been found out for the reconstruction algorithm that there exists a threshold of number s-parse coefficients which when crossed, the algorithm processing takes way more time.

The goal of this work was to present unconventional methods of image acquisition which appeared thanks to a new mathematical theory – compressive sensing (CS) such as a construction of Single-Pixel camera based on digital micro-mirror device (DMD).

DMD based approach is relatively close to conventional camera in that it uses lens to focus the image and only differs in the data acquisition part. However, it requires special hardware (lens, DMD array) in order to work. Thanks to using only single light detector, it can be adapted to operate in unconventional spectral areas without requiring arrays of expensive sensors.

Lensless camera with SLM uses similar principle, but it is a more minimalist solution, requiring only one sensor (like DMD array approach), simple SLM modulator and no lens. However, when taking images of conventional scenes, there is image blurring caused by the absence of the lens and/or diffraction in case of high resolution SLM (SLM behaves like an array of pinhole cameras). But these disadvantages can turn out to be unimportant for some special applications.

Lensless imaging camera with controllable aperture differs fundamentally from the other two approaches. It uses conventional array of sensors, but the controllable aperture allows to tailor the system for image acquisition under special circumstances (moving objects, pattern detection etc.).

The lensless imaging with SLM and a photo-diode was chosen for physical implementation and closer study. The system was successfully assembled from prepared components and its function was verified by the pixel by pixel measuring technique. During this phase, it came out that the success

of the measurement is sensitive to the type of light source used and in this particular construction, favourable results were only achieved using collimated laser beam.

The CoSaMP measurement and subsequent reconstruction process were simulated in Matlab computing environment. For a supplied input picture, a Matlab script first simulates image acquisition via compressive sensing using random matrices, and then from simulated measurement data the original picture is reconstructed. Capability to add noise to the measured values was later added to this system. This simulation was essential to be able to assess the CoSaMP algorithm performance and results for various combinations of algorithm parameters (number of measurement, sparsity of the output vector, noise level). Several simulations were performed and evaluated using various metrics.

During simulation, it turned out that for some parameters the image acquisition and reconstruction process return favourable results, but also that the reconstruction process is very time-intensive (although some optimizations were devised and implemented which mitigated this issue). Also, there were hints of problems with the stability and robustness of the algorithm.

The next step was to implement the CoSaMP sensing process in practical system. The construction itself was working as tested by the pixel-by-pixel measurement. The sensing photo-diode took the measurements which were successfully interpreted in Matlab environment as an image. However, although the CoSaMP algorithm was able to reconstruct image taken just by scanning each pixel by pixel, it was not able to reconstruct an image by the method of compressive sensing – using random matrices and measure just a fraction of measurements. The results of the physical application of CoSaMP in this case were unsatisfactory. The reason of this is still unknown, one of the possible reasons might be a blurring caused by lens-less system, because when using random matrices, more then one pixel is opened in SLM modulator.

6.1 Further work

The construction can be improved in some areas in terms of practical usability.

As was mentioned before, the template was illuminated by a laser. For practical usage it would be advantageous to tune the system so that it works properly with more conventional light source (e.g. testing the system with a collimated beam of light).

Other area of possible exploration is enhancing the fidelity of reproduced images by employing calibration methods, in particular flat-field correction and dark frame correction (i. e. taking into account the value measured when all pixels on the modulator are closed, because in this case even closed pixels transmitted light)

One possibility to improve the physical system would be to use SLM modulator with built-in polarisation layers, which would improve transparency contrast between open and closed pixels.

In this work, effort was made to simulate the system as thoroughly as possible, but it would be possible to improve simulation so that it reflects all physical properties of the system, or optionally

to combine PSF processing with adding noise etc. Also, the measurement of PSF can probably be modified to be more robust, using proper point light source and averaging over many measurements.

One interesting field would be comparison of the presented system with the more complicated system using lenses and DMD array, which should theoretically provide better results. This is hinted by the fact that the simulation was working until PSF was used to modify the measured values.

Bibliography

[1] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling", IEEE Signal Process. Mag., vol. 25, no. 2, pp. 83-91, 2008.

[2] R.G. Baraniuk, "Compressive sensing," IEEE Signal Processing Mag., vol. 24,no. 4, pp. 118–120, 124, 2007.

[3] Gang Huang,Hong Jiang, Kim Matthews and Paul Wilford "Lensless imaging by compressive sensing," Bell Labs, Alcatel-Lucent, Murray Hill, NJ 07974

[4] A. Zomet and S. K. Nayar, "Lensless Imaging with a Controllable Aperture", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun, 2006.

[5] Deanna Needell and Joel A. Tropp "CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples",Communications of the ACM,Dec, 2010.

[6] Graeme Pope, "Compressive Sensing A Summary of Reconstruction Algorithms",Eidgenössische Technische Hochschule, Zürich,Feb, 2009.

[7] Philip Breen, "Algorithms for Sparse Approximation",University of Edinburgh,2009

[8] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," IEEE Trans. Inform. Theory, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[9] Mallat, S. G. and Z. Zhang (1993, Dec). Matching pursuits with time-frequency dictionaries,Signal Processing, IEEE Transactions on 41(12), 3397–3415.

[10] Pati, Rezaiifar, and Krishnaprasad, Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition ,1993

[11] Davis, Mallat, and Avellaneda ,Orthogonal Matching Pursuit,1997

[12] G. Davis, S. Mallat, Z. Zhang,Adaptive time-frequency decompositions with matching pursuits,Optical Engineering, New York University, Courant Institute 251 Mercer Street, New York, NY 10012,1994

[13] http://www.mathworks.com/matlabcentral/fileexchange/32402-cosamp-and-omp-for-sparse-recovery

[14] Daniel Kekrt, Tomáš Lukeš, Miloš Klíma, Karel Fliegel ,2D Iterative MAP detection: Principles and Applications in Image Restoration,RADIOENGINEERING, VOL. X, NO. Y, JUNE 2014

[15] Image Quality Assessment: From Error Visibility to Structural Similarity,Zhou Wang, Member, IEEE, Alan Conrad Bovik, Fellow, IEEE, Hamid Rahim Sheikh, Student Member, IEEE, and Eero P. Simoncelli, Senior Member, IEEE

[16] BOREMAN, Glenn D. Modulation transfer function in optical and electro-optical systems. SPIE PRESS. 2001, ISBN 0-8194-4336-0.

[17] http://www.ni.com/white-paper/13306/en/pdf

[18] http://www.mathworks.com/matlabcentral/fileexchange/32402-cosamp-and-omp-for-sparse-recovery

Appendix A

Appendix A

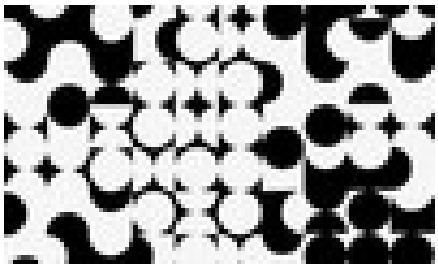
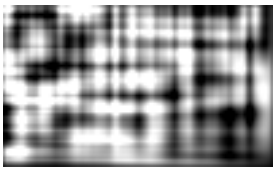
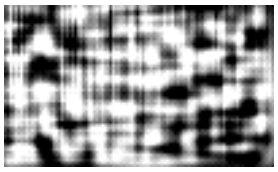


Figure 1: original picture of geometric shape

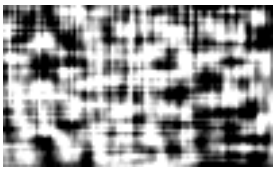
(a) s = 50



(b) s = 100

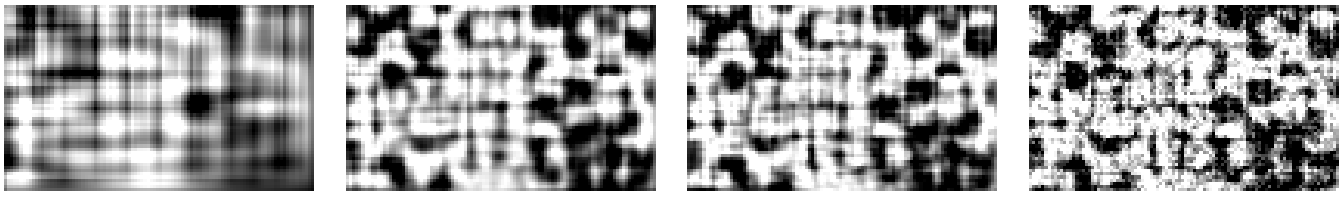


(c) s = 200



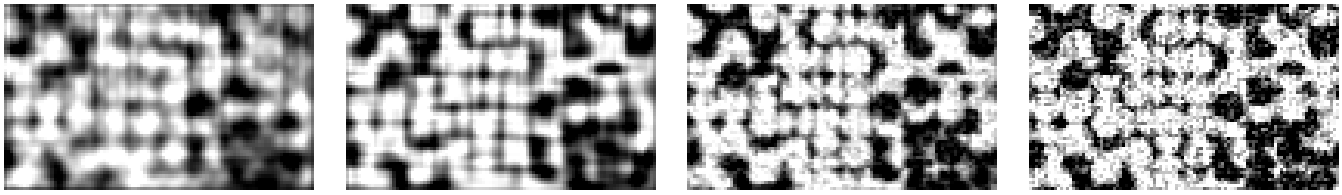
(d) s = 300

Figure 2: **1000 measurements**without noise



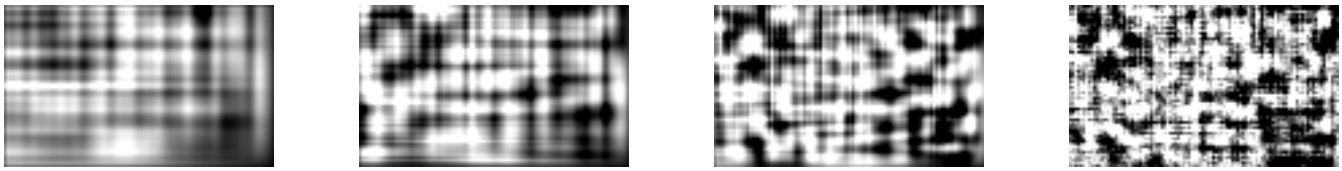
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 3: **2000 measurements**without noise



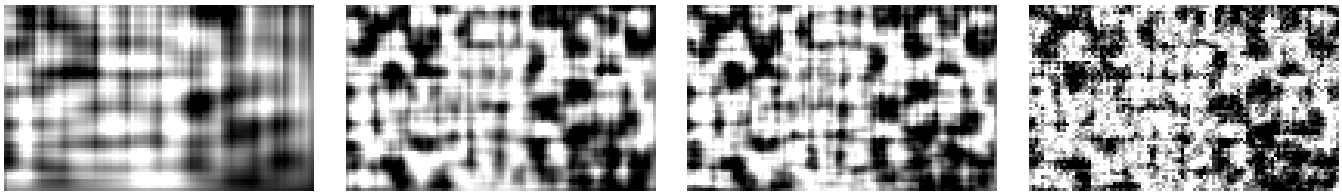
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 4: **3000 measurements**without noise



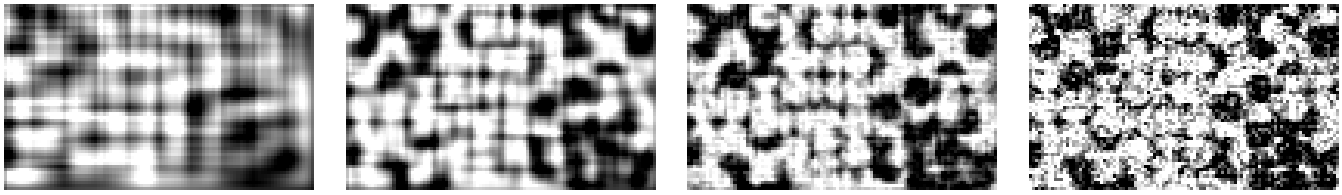
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 5: **1000 measurements**with noise,k=500



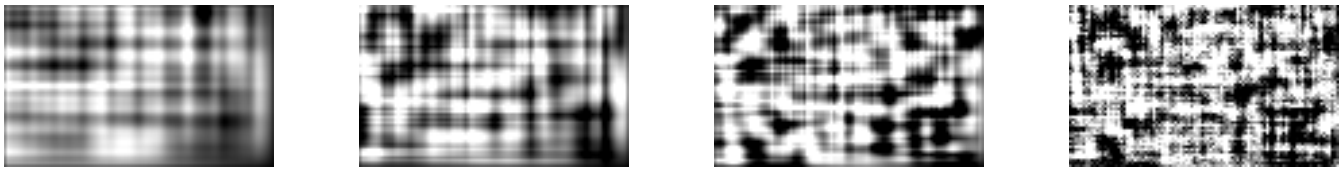
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 6: **2000 measurements**with noise,k=500



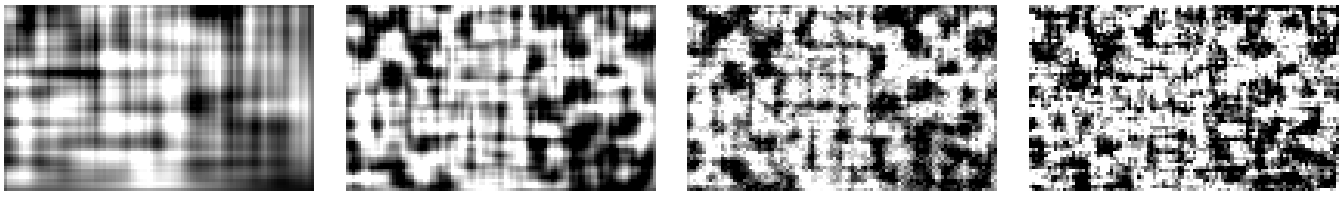
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 7: **3000 measurements**with noise,k=500



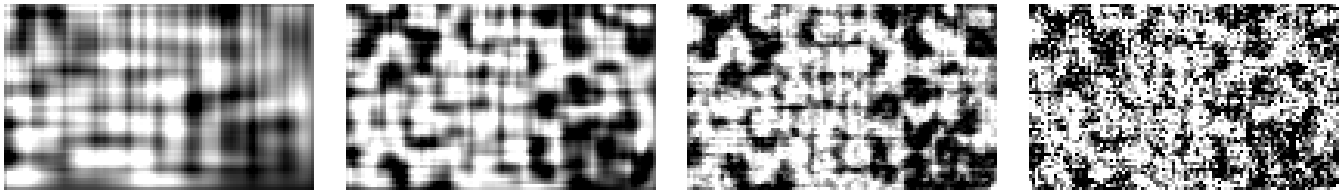
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 8: **1000 measurements**with noise,k=1000



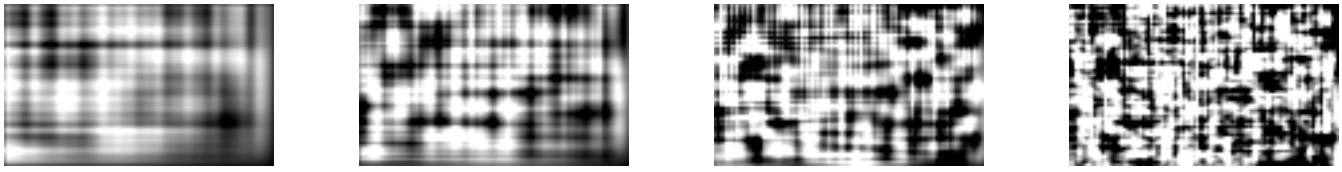
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 9: **2000 measurements**with noise,k=1000



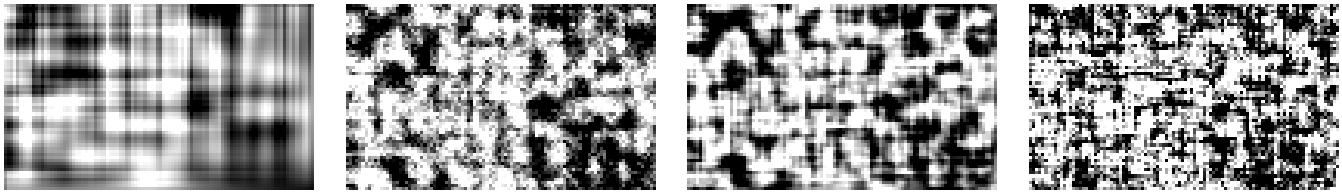
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 10: **3000 measurements**without noise



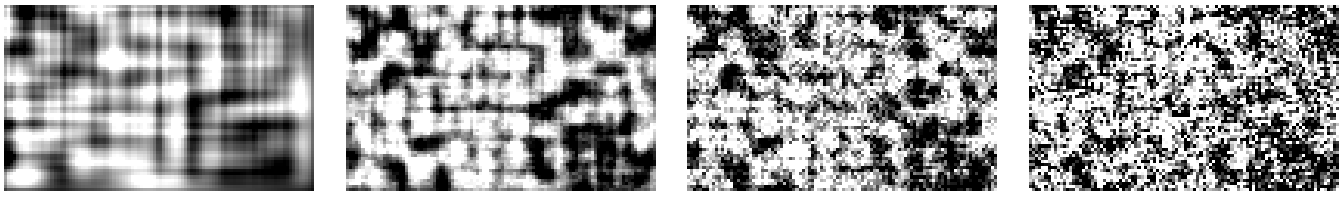
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 11: **1000 measurements**with noise,k=1500



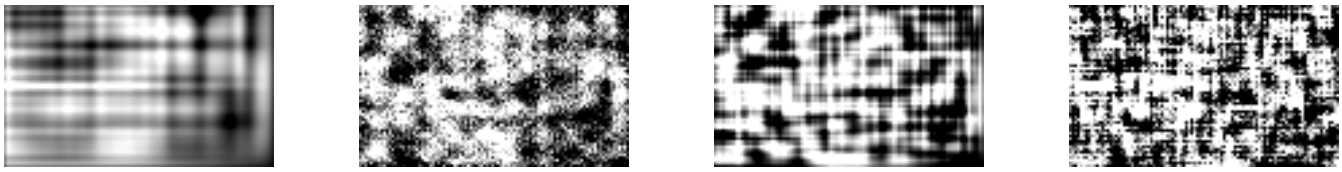
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 12: **2000 measurements**with noise,k=1500



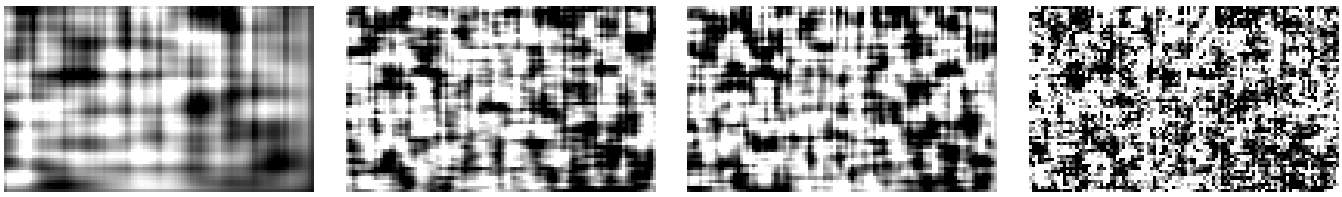
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 13: **3000 measurements**with noise,k=1500



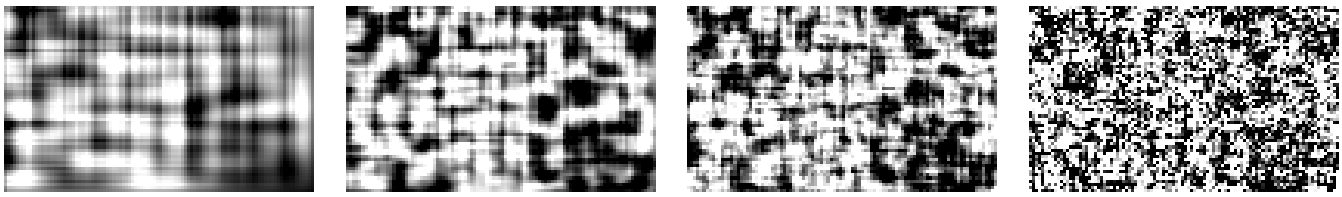
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 14: **1000 measurements**with noise,k=2000



(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 15: **2000 measurements**with noise,k=2000

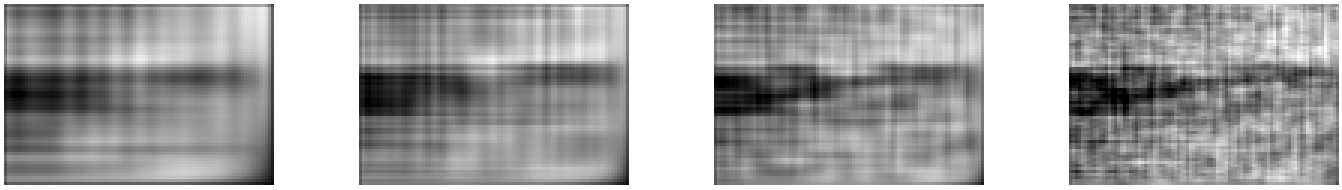


(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 16: **3000 measurements**with noise,k=2000

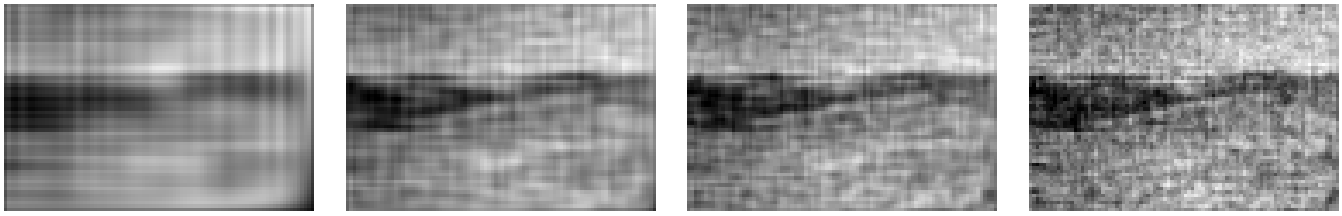
Picture of a landscape :

Figure 17: original picture landscape



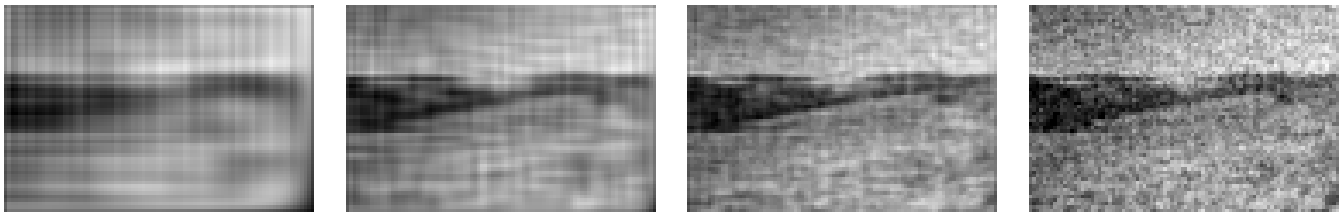
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 18: **1000 measurements**without noise



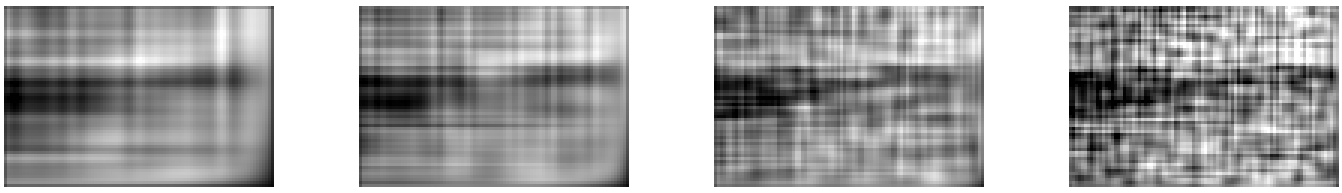
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 19: **2000 measurements**without noise



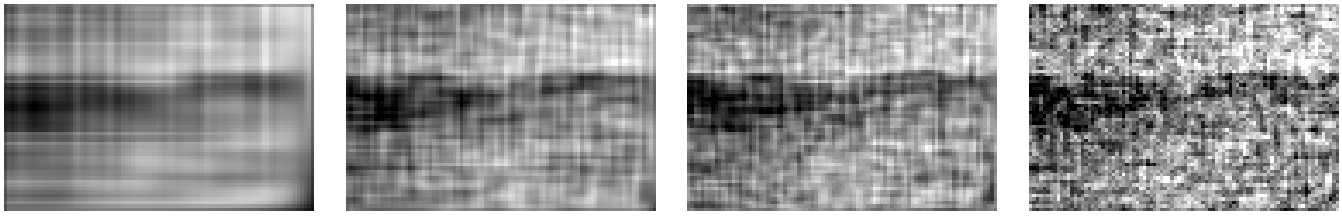
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 20: **3000 measurements**without noise



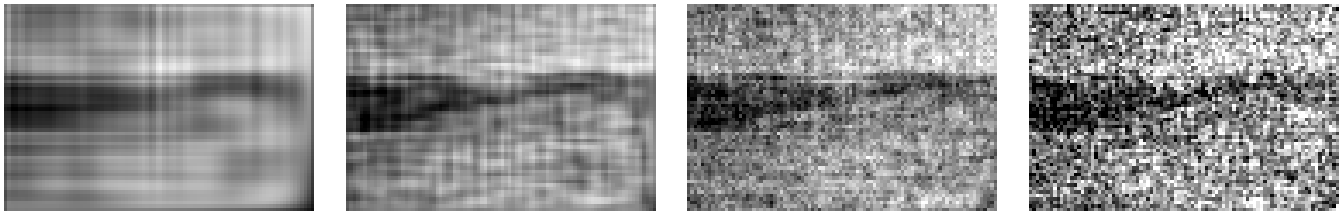
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 21: **1000 measurements**with noise,k=500



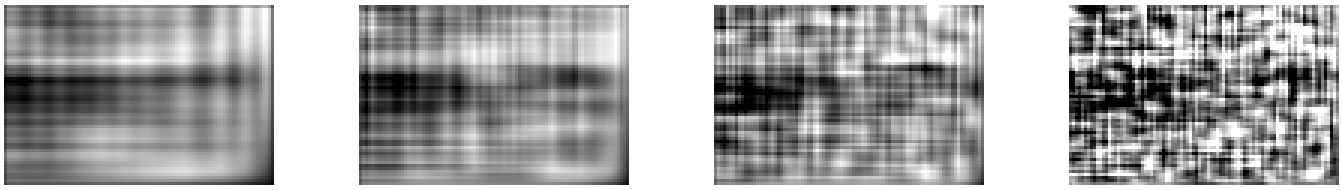
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 22: **2000 measurements**with noise,k=500



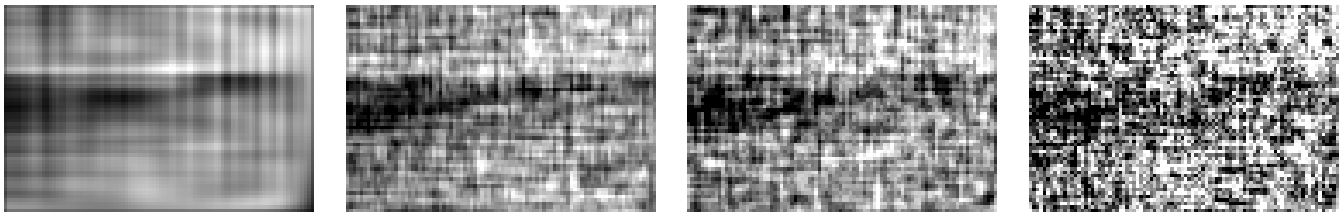
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 23: **3000 measurements**with noise,k=500



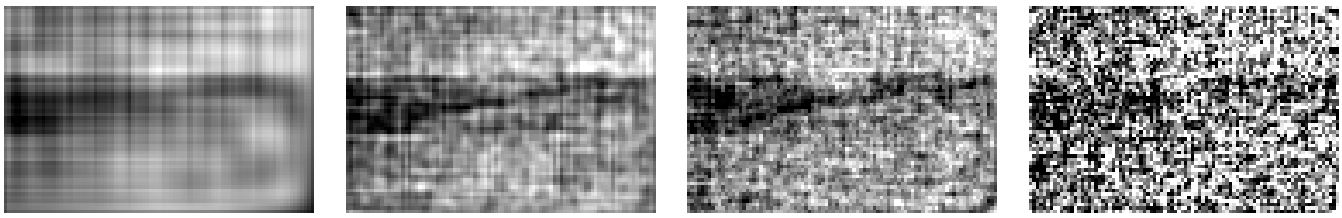
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 24: **1000 measurements**with noise,k=1000



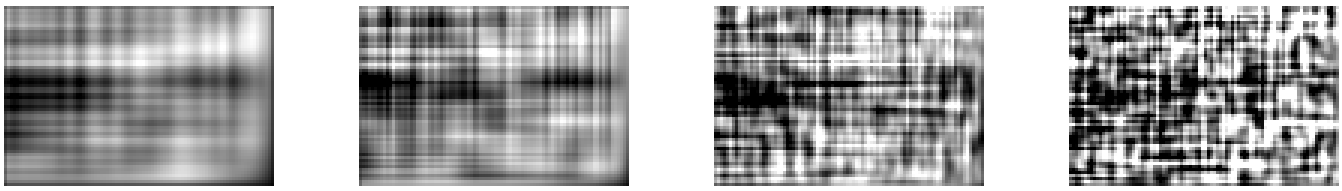
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 25: **2000 measurements**with noise,k=1000



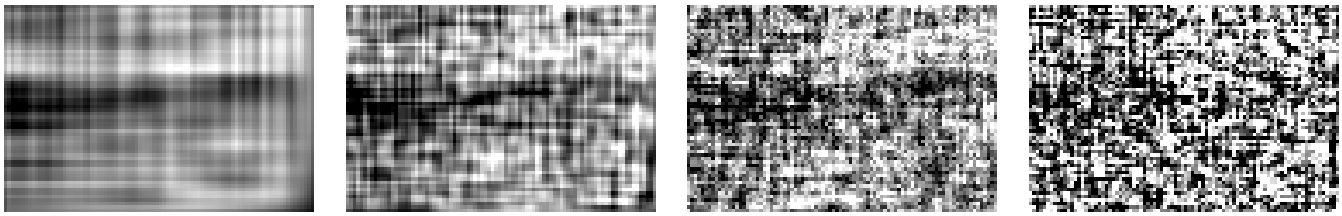
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 26: **3000 measurements**without noise



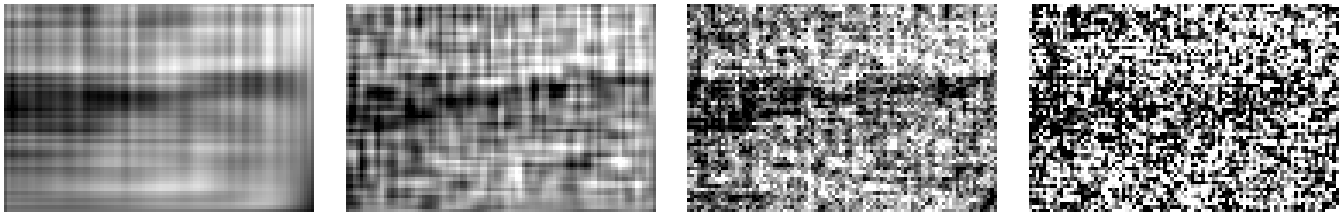
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 27: **1000 measurements**with noise,k=1500



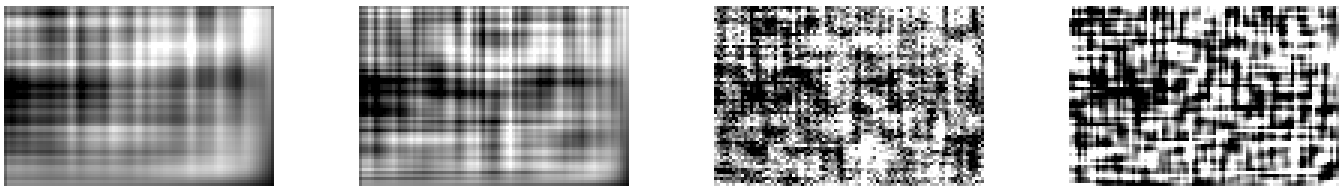
(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 28: **2000 measurements**with noise,k=1500



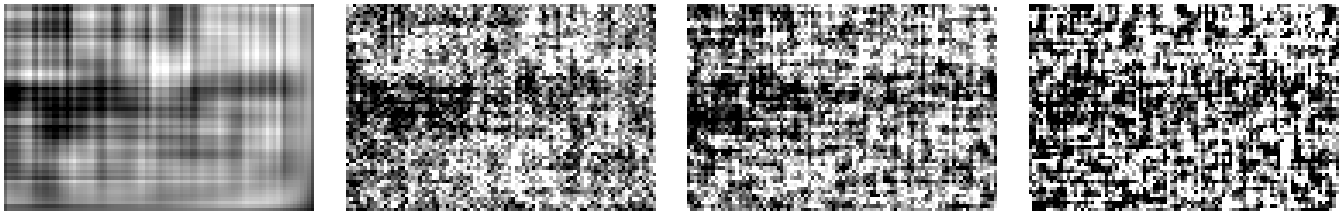
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 29: **3000 measurements**with noise,k=1500



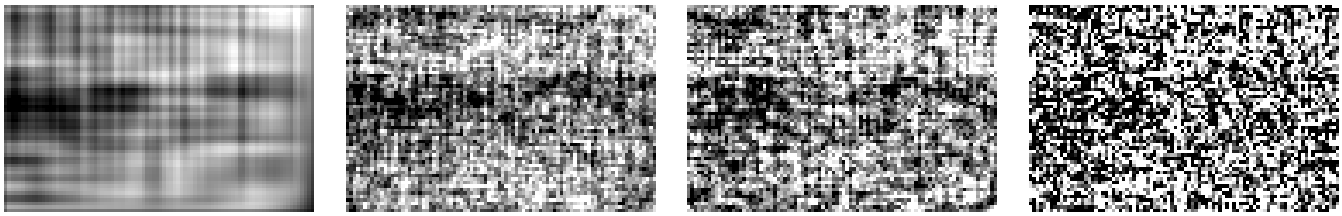
(a) s = 50 (b) s = 100 (c) s = 200 (d) s = 300

Figure 30: **1000 measurements**with noise,k=2000



(a) s = 100 (b) s = 300 (c) s = 400 (d) s = 600

Figure 31: **2000 measurements**with noise,k=2000



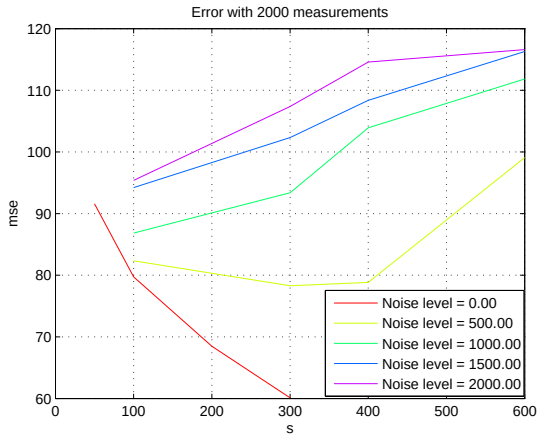
(a) s = 100 (b) s = 200 (c) s = 500 (d) s = 1000

Figure 32: **3000 measurements**with noise,k=2000

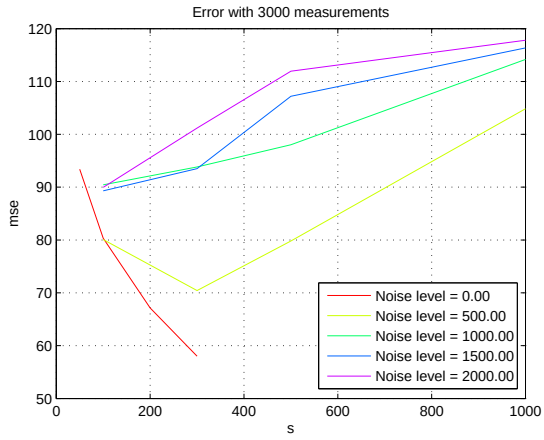
Appendix B

Appendix B

Graphs:

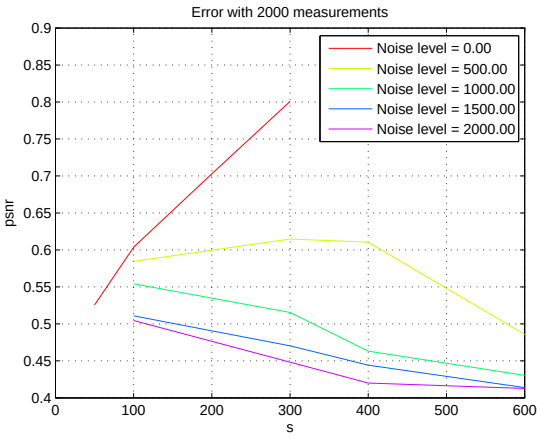


(a) 2000 measurements

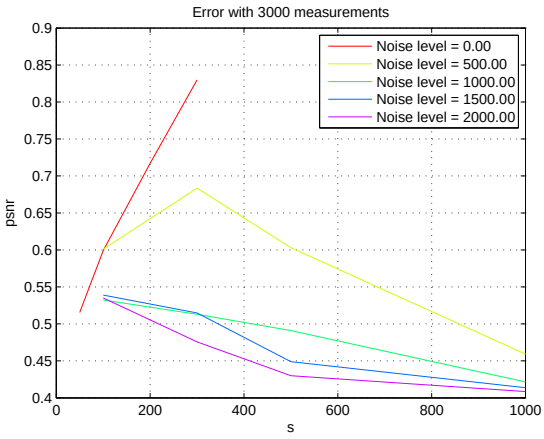


(b) 3000 measurements

Figure 33: MSE

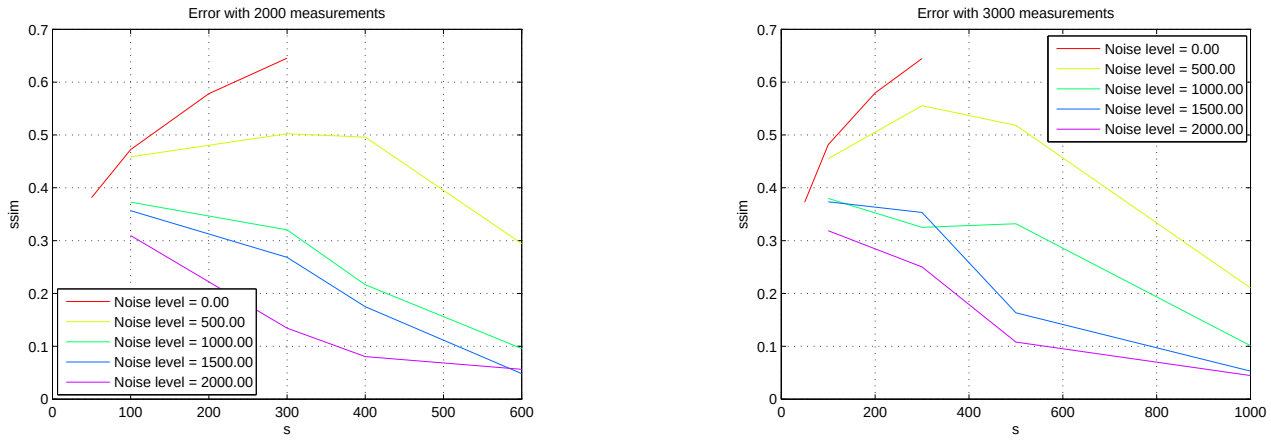


(a) 2000 measurements



(b) 3000 measurements

Figure 34: PSNR



(a) 2000 measurements

(b) 3000 measurements

Figure 35: SSIM

Appendix C

Appendix C

Simulation in Matlab software:

Main file for simulating CoSaMP algorithm:

```
close all;
clear all;



% loading picture
 picture = imread('geom.jpg');
 picture = rgb2gray(picture);
 x=double(picture(:));

%define imput parameters
s = 1000; %sparse coeffitients must be lower then 1/3 of measurements
number_of_measurements = 3000;

%caching matrices phi and theta
fileName = strcat( ...
    'matice/theta-', ...
    num2str(size(picture, 2)), ...
    'x', ...
    num2str(size(picture, 1)), ...
    '-', ...
    num2str(number_of_measurements), ...
    '.mat');
if exist(fileName, 'file')
    load(fileName);
else
    phi=(rand(number_of_measurements, length(x)));
    theta = kratDct(phi, size(picture,2),size(picture,1));
    if ~exist('matice', 'dir')
        mkdir('matice');
    end
```

```
    save(fileName, 'theta', 'phi');
end

% ------- simulation compressive sensing-----------------------------------

% simulation taking image aquisition
y=phi*x;
% simulation reconstructing the image by CoSaMP algorithm
[Sest,r,normR,residHist,errHist] = CoSaMP2(theta, y, s);

 reconstructedPicture=uint8((dctKrat(Sest, size(picture,2),size(picture,1))));
 outputPicture = vec2mat(reconstructedPicture,size(picture,1),size(picture,2))';

 figure(1)
 imshow(picture);

 figure(2)
 imshow(outputPicture,[]);
```

Functions for creating DCT matrices:

```
% a = b * dctBazickaMatice

function a = kratDct(b, width, height)

    if size(b, 2) ~= width*height
        throw(MException('argument:sizeCheck', 'Incorrect matrix size'));
    end

    a = zeros(size(b));

    spatialdomain = zeros(height, width);
    for rowDct = 1:height
        for columnDct = 1:width
            spatialdomain(rowDct, columnDct) = 1;
            dctfrequencyDomain = dct2(spatialdomain);
            dctVektor = dctfrequencyDomain(:);
            spatialdomain(rowDct, columnDct) = 0;

            columnA = (rowDct - 1) * width + columnDct;

            for rowA = 1:size(a, 1)
                a(rowA, columnA) = b(rowA, :) * dctVektor;
            end
        end
    end
end
```

```
% a = dctBazickaMatice * b

function a = dctKrat(b, width, height)

    if size(b, 1) ~= width*height
        throw(MException('argument:sizeCheck', 'Incorrect matrix size'));
    end

    a = zeros(size(b));

    spatialdomain = zeros(height, width);
    for rowDct = 1:height
        for columnDct = 1:width
            spatialdomain(rowDct, columnDct) = 1;
            dctfrequencyDomain = dct2(spatialdomain);
            dctVektor = dctfrequencyDomain(:);
            spatialdomain(rowDct, columnDct) = 0;

            columnBaze = (rowDct - 1) * width + columnDct;

            for columnA = 1:size(a, 2)
                a(:, columnA) = a(:, columnA) + dctVektor * b(columnBaze, columnA);
            end
        end
    end
end
```

```
close all; clear all;
% size of the taking image
 sz = 50;

dctbazickamaticepsi=dctbaze(sz,sz);
number_of_measurements=1000;
phi=(rand(number_of_measurements,(sz^2))));
theta=phi*dctbazickamaticepsi;

 y=zeros(size(theta,1));

 mkdir('test');
    cd test


for i=1:size(theta,1)
    thetaradek=theta(i,:);
    thetamat=vec2mat(thetaradek,sz);

```

```
20      pripona='.png';
21
22
23  %       jmeno =[i,pripona];
24      jmeno=[int2str(i),pripona];
25
26
27     B = imresize(thetamat,[600,600],'nearest');
28
29      alpha = 1*ones(size(B));
30      imwrite(B,jmeno,'Alpha',alpha);
31
32
33      pause(1e-2)
34      hodnota=1;
35      y(i)=hodnota;
36  end
37
38  cd('..')
39   system('python slm.py 1000')
40
41   vstup_pro_cosamp = csvread('measure.csv');
42
43  s = 300;
44  [Sest,r,normR,residHist,errHist] = CoSaMP2(theta, vstup_pro_cosamp, s);
45
46  zrekonstruovanyobrazek=((dctKrat(Sest, sz,sz)));
47  vystupniObrazek = vec2mat(zrekonstruovanyobrazek,sz,sz);
48
49  figure(2)
50  imshow(vystupniObrazek,[]);
```

Appendix D - contents of DVD

- **construction-measured data**: saved data, scripts and figures from measurements
  - `CoSaMP2.m` – implementation of CoSaMP algorithm (taken from [18])
  - `dctbaze.m` – generates DCT base matrix of the given dimensions
  - `dctKrat.m` – multiplies the given matrix by DCT base matrix from the left
  - `experiment1.m` – runs the measurement for the physical construction
  - `hybrid.m` – runs the CoSaMP for the physical construction, but with pixel-by-pixel measurement
  - `kratDct.m` – multiplies the given matrix by DCT base matrix from the right
  - `pix_bila_test.m` – measures pixel-by-pixel, but measures with all pixels on the SLM white each 50 measurements
  - `pix_po_pix0.m` – measures pixel-by-pixel on the physical system
  - `slm.py` – Python code for displaying matrices on the SLM modulator
- `simulation`: Matlab scripts for running the simulation. The main simulation is ran by running the file `example-run.m`
  - `CoSaMP2.m` – implementation of CoSaMP algorithm
  - `cosampkonstanta.m` – multiplying input to CoSaMP algorithm a constant
  - `dctbaze.m` – generates DCT base matrix of the given dimensions
  - `dctKrat.m` – multiplies the given matrix by DCT base matrix from the left
  - `example-run.m` – main script for running the simulation
  - `getPsf.m` – loads the file with measured PSF and scales (resamples) it to fit a specified size
  - `grafychyb.m` – draws error levels graphs for outputs of the simulation data
  - `kazdydruhy.m` – loads saved data from real measurements, proceeds like pixel-by-pixel but only uses each second pixel as input for CoSaMP
  - `kratDct.m` – multiplies the given matrix by DCT base matrix from the right
  - `loadData.m` – loads saved data from previous simulation
  - `measure.m` – runs simulation of one measure for specified parameters
  - `noiseLevelGraph.m` – draws one graph for error level metrics
  - `ssim.m` – computes SSIM, taken from [15]
- `prochazkova.pdf`: PDF text of this thesis