

# COMPRESSED SENSING AND SOME IMAGE PROCESSING APPLICATIONS

Aleš Hladnik<sup>1</sup> , Pavle Saksida<sup>2</sup> 

<sup>1</sup> University of Ljubljana, Faculty of Natural Sciences and Engineering,  
Department of Textiles, Graphic Arts and Design, Ljubljana, Slovenia

<sup>2</sup> University of Ljubljana, Faculty of Mathematics and Physics, Ljubljana, Slovenia

**Abstract:** We live in a digital media-overloaded world. An enormous number of images, sound and video files are continuously being created and either transmitted over the internet or stored on hard drives or portable storage devices. During their transmission or storage, however, such digital files are almost without an exception subjected to a process of discarding most of their original information, since e.g. fast opening of a web site image or a small audio file size are today of utmost importance. Loss in redundant or imperceptible information is therefore inevitable and incorporated in lossy compression algorithms such as JPEG, MPEG or MP3, but to record raw video or audio data only to be, in large part, soon discarded during the process of sending it to a receiver is obviously not an optimum approach. Compressed sensing is a signal processing technique that provides one solution to the above problem. Rather than performing acquisition followed by compression of a signal, it combines both steps in a single sensing – or sampling – operation. In other words, compressed sensing allows acquiring signals while taking only a few samples. One of the underlying assumptions of the signal is that it is sparse, i.e. it should be possible to represent it with a matrix, consisting of a large number of zero – or close to zero – coefficients. Images, when represented in a non-spatial domain, such as discrete cosine- or wavelet-domain, often comply with such a requirement.

Theory behind the compressed sensing will be presented briefly together with several examples of successful implementation of this method in the field of signal – mainly image – processing.

**Key words:** compressed sensing, image acquisition, sparsity, signal processing

## 1. INTRODUCTION

The number of digital media files, such as still images, sound and video clips that are created every day is astonishing. For example, in 2017 every minute more than 400 hours of video were uploaded to YouTube, and 136,000 and 46,740 photos were uploaded on Facebook and Instagram, respectively (Schultz, 2017). However, the vast majority of this data is discarded during the process of its compression in order to secure its fast network transmission or a compact file storage on hard drives or portable storage devices. Loss in redundant or imperceptible information is therefore inevitable and incorporated in widely used lossy compression algorithms such as JPEG, MPEG or MP3. This traditional approach consisting of a raw data acquisition followed by its compression therefore seems wasteful; one can argue whether it would be possible to directly acquire just the useful part of a signal?

### 1.1 Compressed sensing

*Compressed sensing* (CS), also known as *compressive sensing/sampling* and *sparse sampling*, is a signal processing technique developed in years 2004-2006 (Candes et al, 2006) that attempts to solve this problem. In the conventional signal acquisition and compression scheme (Figure 1 left), a signal consisting of  $N$  samples (e.g. pixels in an image) is compressed producing a much smaller number of  $K$  elements (e.g. discrete Fourier transform (DFT), cosine transform (DCT) or wavelet transform (DWT) coefficients), which are transmitted from the sender to the receiver where a decompression – reconstruction – takes place, resulting in an approximation of the original signal, with a hopefully no significant perceptual difference to the original. In the CS workflow (Figure 1 right), on the other hand, signal acquisition and compression operations are merged – or rather substituted by – a single sensing (sampling) step. Here  $N$  samples are replaced by  $M$  measurements, which are then, similar to the conventional workflow, sent over the network and used to perform the reconstruction of the signal. Note that  $K < M \ll N$ . It can be shown (Candes et al., 2006) that the reconstruction is perfect (exact) as long as  $M = O(K \cdot \log(N/K))$ ; see Figure 2. Two important prerequisites for CS to work are *sparsity* and *incoherence* and are discussed below.

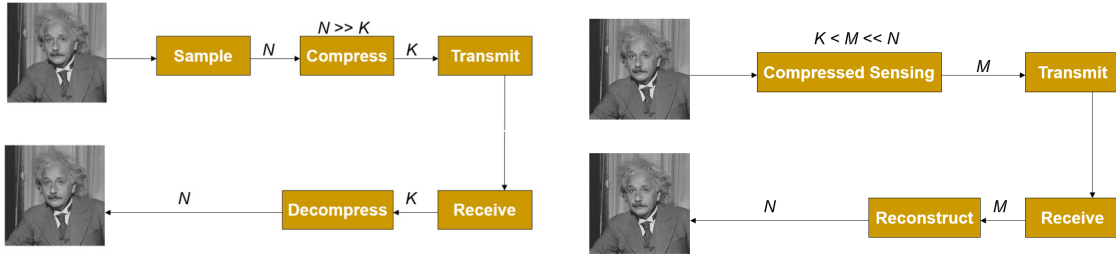


Figure 1: Conventional signal acquisition and compression scheme (left) and CS workflow (right) (Mancera, 2008).

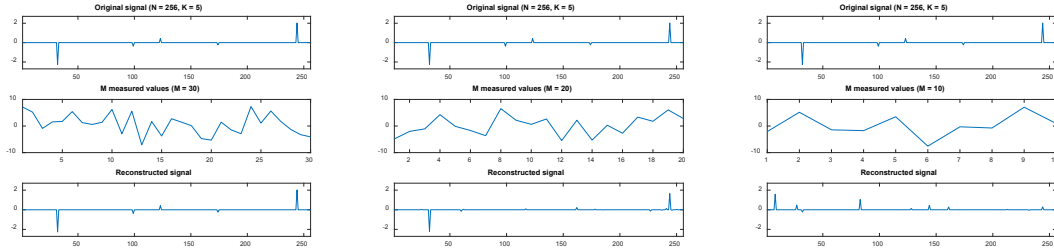


Figure 2: Original 1D signal has 5 non-zero values ( $K = 5$ ). As the number of random measurements  $M$  decreases ( $30 \rightarrow 20 \rightarrow 10$ ), signal reconstruction deteriorates and becomes suboptimal when  $M < 20$ . This is in agreement with the formula  $M = O(K \cdot \log(N/K))$ , since the computed value for  $M$  is 19.68.

### 1.2 Signal reconstruction

In principle, signal reconstruction is based on the idea of finding a solution to an underdetermined linear system  $y = \Phi x$  (Figure 3 left) (Milliarde, 2016), where for the CS purposes  $y$  and  $x$  represent measurement and signal vectors, respectively, while  $\Phi$  is a sampling/sensing matrix, also known as a *dictionary*. Such a system is characterized by having fewer equations than unknowns –  $M < N$  – and generally has an infinite number of solutions. However, by imposing certain constraints to the system, the CS framework allows obtaining one particular solution. One such constraint is that the signal  $x$  is  $K$ -sparse, i.e., it must be possible to represent it with a vector, consisting of a large number of zero – or close to zero – coefficients and only  $K$  non-zero values. Digital images, when represented in a non-spatial, i.e. transform domain, such as discrete cosine- or wavelet-domain, often comply with such a requirement. If this is the case, the scheme shown in Figure 3 (right) applies. Here  $\Psi$  denotes a representation matrix corresponding to – when the signal is an image – DFT, DWT or DCT matrix coefficients.  $\Phi$  is, as mentioned above, a dictionary consisting of randomly distributed values or, less often, of predetermined sequences, such as pseudo-random codes, binary codes, noiselets, etc. The second CS prerequisite is that the two orthonormal bases  $\Phi$  and  $\Psi$  are incoherent, i.e. poorly correlated to each other.

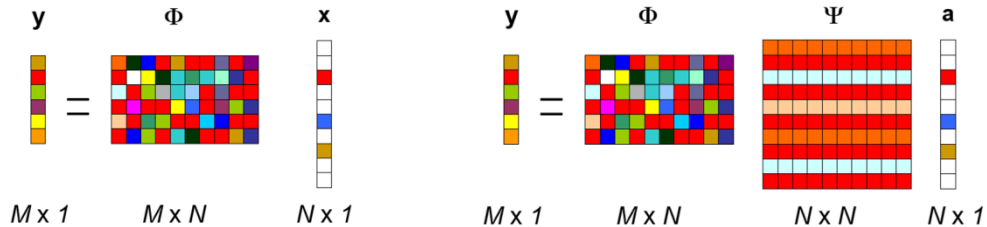


Figure 3: A general underdetermined linear system of equations  $y = \Phi x$  having an infinite number of solutions (left) and a CS scheme  $y = \Phi \Psi a$  where it is possible to obtain  $\hat{a}$  using numerical optimization routines (right).

From  $y = \Phi \Psi a$  one can recover  $a$ , i.e. compute  $\hat{a}$ , by solving Eq. 1:

$$\hat{a} = \arg \min_{\mathbf{a}} \{ \|\mathbf{a}\|_0 \text{ s.t. } \mathbf{y} = \Phi \Psi \mathbf{a} \} \tag{1}$$

Since this is a computationally complex combinatorial (NP-hard) problem, various approximations using rather involved numerical non-linear optimization algorithms have been implemented (Adcock, 2015). Going into these techniques is beyond the scope of this introductory paper. One of the widely adopted approaches is the minimization of  $L_1$ -norm (known as *Basis pursuit*), which leads to the sparsest solution. Vector  $\hat{a}$  obtained through this procedure has to be transformed back into the spatial domain, to obtain the conventional image representation.

## 2. RESULTS AND DISCUSSION

### 2.1 1D signal reconstruction

First, let's take a look at how we can reconstruct a 1D signal of length  $N = 256$ , such as a sound wave (Figure 4a) by means of CS. Unlike in Figure 2, the signal of interest is now in the frequency – DFT – domain (Figure 4b). It has six peaks, corresponding to the three frequencies ( $K = 3$ ) contained in the original, time domain, signal: 31, 98 and 122. This is due to the fact that each real-valued sinusoid is equivalent to two complex sinusoids of frequency  $f$  and  $-f$  and consequently has two peaks in the frequency domain. By performing 64 random measurements  $M$  (Figure 4c) it is possible to recover the signal in the frequency domain (Figure 4d) and – via inverse DFT – in the time domain (Figure 4e) perfectly. When the number of non-zero elements in the sparse representation of the same signal is raised to five ( $K = 5$ ; frequencies: 21, 31, 39, 98, 122), however, the reconstruction with 64 measurements is no longer satisfactory (Figure 5) and  $M$  has to be increased as well to secure a perfect recovery.

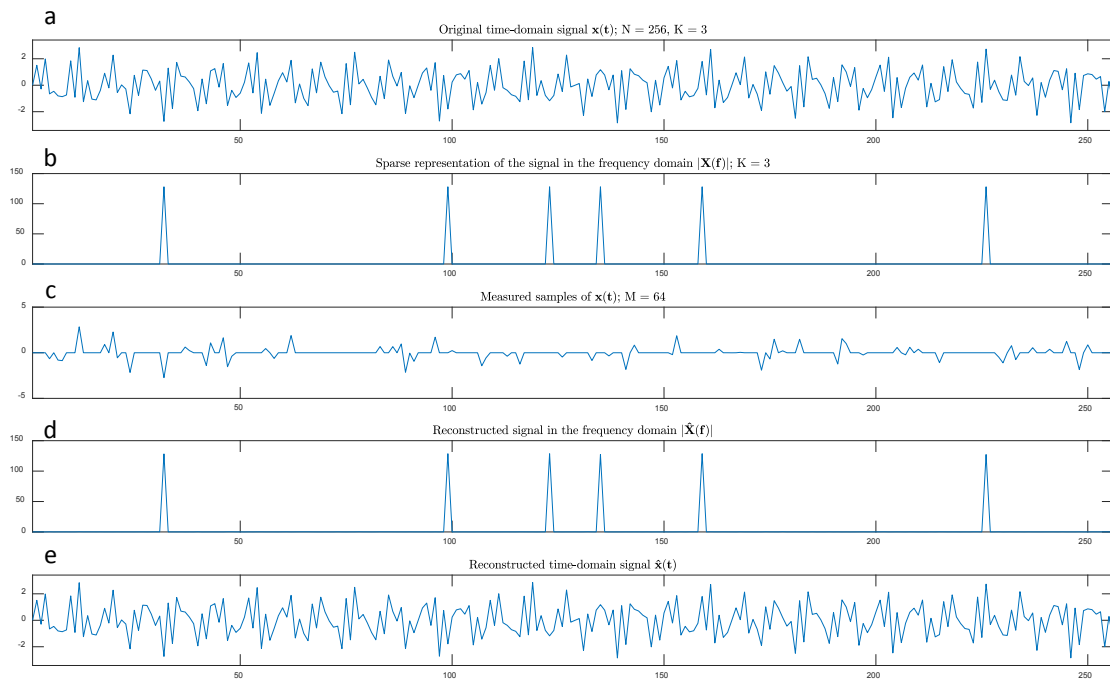


Figure 4: Perfect reconstruction of a 1D signal with  $N = 256$ ,  $M = 64$ ,  $K = 3$ .

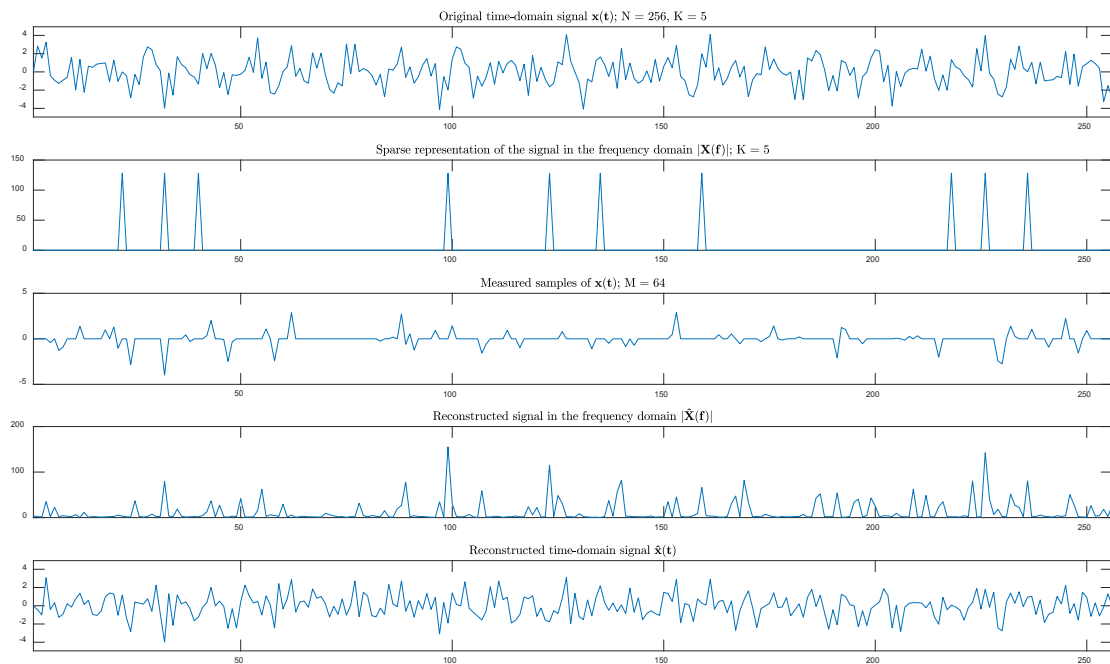


Figure 5: Approximate reconstruction of a 1D signal with  $N = 256$ ,  $M = 64$ ,  $K = 5$ .

## 2.2 2D signal reconstruction

The procedure for reconstructing a 2D spatial domain signal, such as a digital photograph, is in principle similar to the recovery of a 1D time domain signal shown above and is demonstrated here using a MATLAB code (Gibson, 2013) and a well known *cameraman* test image (Figure 6). For the sake of simplicity it is assumed that the image is sparse – strictly speaking, *compressible* – in the DCT domain ( $\Psi$  is a DCT matrix); note that in general, it is more likely for a DWT-, rather than a DCT-, representation of a real-scene image that the assumption of sparsity is met. Dictionary  $\Phi$  is a matrix of normally distributed random measurements. Due to certain computational issues, instead of reconstructing the whole image, only the two  $50 \times 50$  px regions, i.e. subimages, were processed ( $N = 2500$ ). The number of measurements  $M$  was  $N/2$ ,  $N/4$  and  $N/6$ , respectively. For obtaining  $\hat{a}$ , two approaches were adopted:  $L_1$ -norm minimization (Basis pursuit; BP) and  $L_2$ -norm minimization (Least squares solution; LS). Results are displayed in Figure 6.

It is evident that the Basis pursuit algorithm outperforms the Least squares approach by a large margin. Contrary to the latter one where even with  $M = N/2 = 1250$  the reconstructed image is extremely noisy and of a poor quality, the application of the former one leads to an acceptable reconstruction of the image regions – especially where lower frequencies dominate, such as in the subimage displaying the man's coat – even with  $M = N/4 = 625$ . Of course, when the number of measurements  $M$  decreases too much, it becomes unlikely that the  $K$ -sparse signal will be recovered with a high precision.

With an additional mathematical effort it is possible to improve CS image reconstruction results even further. By implementing a more sophisticated numerical optimization algorithm OWL-QN (Galen et al, 2003) in Python, the author (Pyrunner, 2016) was able to quite successfully reconstruct the  $1600 \times 1200$  px Escher-Wyss' image *Waterfall* by randomly taking only 10% of the samples (= measurements); see Figure 7 left. The image is recognizable even when starting from as little as 1% of the available data (Figure 7 right).

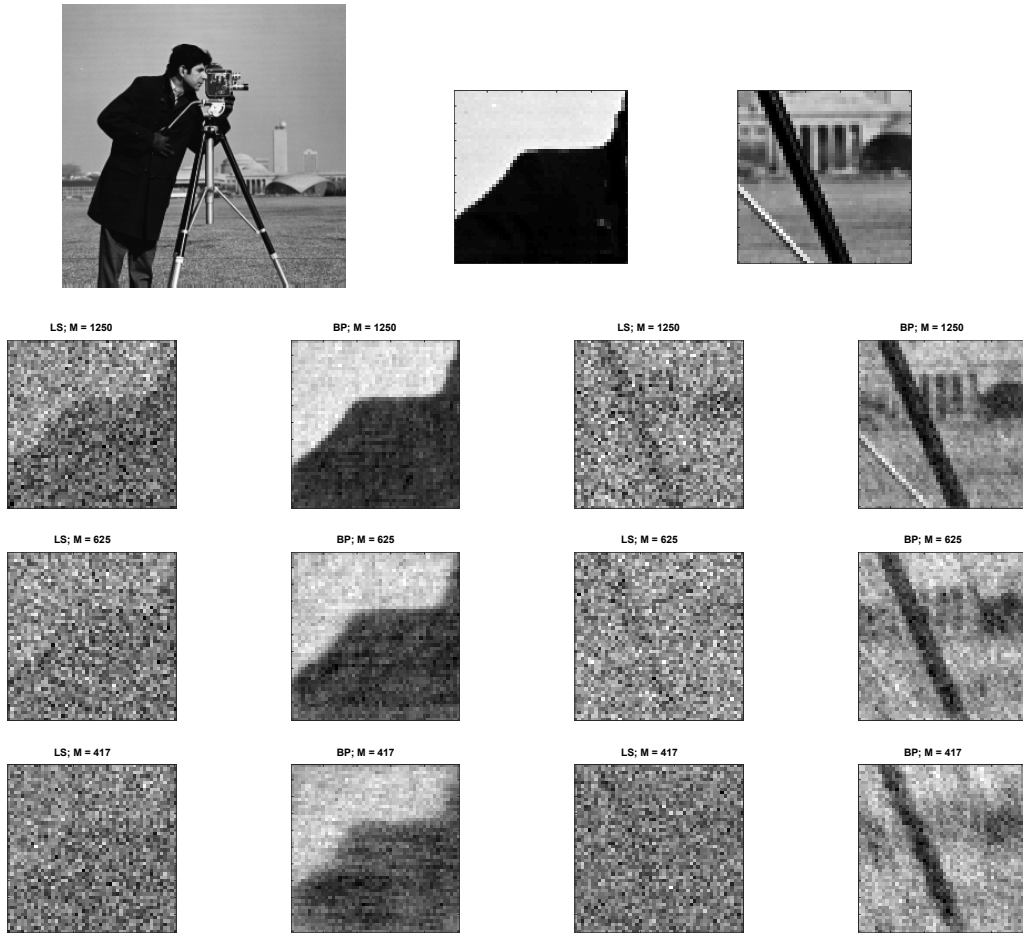


Figure 6: Image reconstruction results using minimization of  $L_1$ -norm and  $L_2$ -norm with  $N = 2500$  and three different  $M$  values:  $N/2$ ,  $N/4$  and  $N/6$ .

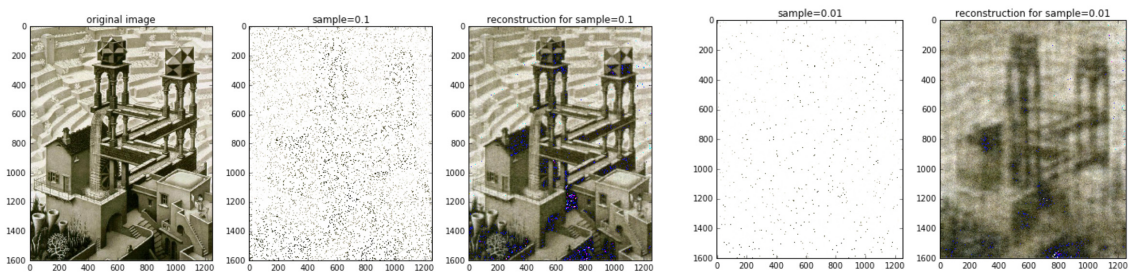


Figure 7: Image reconstruction results with advanced mathematical optimization: recovery from 10% (left) and 1% (right) of the original image data (Pyrunner, 2016).

As discussed above, one of the strengths of CS is that it allows acquiring signal  $a$  while taking only a few measurements  $y$  – see Figure 3b. This is especially beneficial and applicable in imaging techniques such as Magnetic resonance imaging (MRI). MRI uses magnetic fields to excite hydrogen atoms in the patient’s body, resulting – after modelling – in measurements of the DFT of the image. Since the number of measurements is roughly proportional to the scan duration, longer scans are unpleasant for the patient, more prone to motion artifacts and consequently virtually impossible for certain moving body parts, e.g. lungs (Adcock, 2015). It is therefore highly desirable that the number of measurements – i.e. the scanning time – is as low as possible. Figure 8 shows an early example of a CS reconstruction (Mancera, 2008). The recovered image is clearly sharper and has less artifacts compared to the result obtained using the alternative – back-projection reconstruction – algorithm (Wikipedia, 2018).

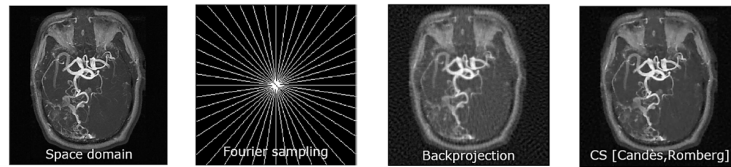


Figure 8: Magnetic resonance imaging – reconstruction results using backprojection- and CS approach.

### 3. CONCLUSIONS

CS is a signal processing technique that uses a sparse signal, which can be sampled at a rate less than the Nyquist-Shannon sampling rate, in order to reconstruct the signal via a constrained optimization method, such as  $L_1$ -norm. It has during the last ten years developed into a rich research area of itself spanning a wide variety of applications, such as photography, computer tomography, transmission electron microscopy, radio interferometry, and many others. Developing new applications as well as new recovery algorithms remains an active field of research.

### 4. REFERENCES

- [1] Adcock, B.: "An introduction to compressed sensing", URL: <http://benadcock.org/wp-content/uploads/2015/11/AdcockBeijingTalk1.pdf> (last request: 2018-09-05).
- [2] Candès, E. J., Romberg, J. K., Tao, T.: "Stable signal recovery from incomplete and inaccurate measurements", *Communications on Pure and Applied Mathematics*, 59 (8), 1207–1223, 2006. doi: 10.1002/cpa.20124.
- [3] Galen, A.; Gao, J.: "Scalable training of  $L_1$ -regularized log-linear models", *Proceedings of the 24th international conference on Machine learning 2007, (ICML, Oregon, USA, 2007.)*
- [4] Gibson, S.: "Simple Compressed sensing example", URL: <https://www.mathworks.com/matlabcentral/fileexchange/41792-simple-compressed-sensing-example> (last request: 2018-09-05).
- [5] Mancera, L.: "Compressed Sensing", URL: [decsai.ugr.es/vip/files/presentations/20080701-CS.pps](https://decsai.ugr.es/vip/files/presentations/20080701-CS.pps) (last request: 2018-09-10).
- [6] Milliarde: "Compressed Sensing Intro & Tutorial w/ Matlab", URL: <https://www.codeproject.com/Articles/852910/Compressed-Sensing-Intro-Tutorial-w-Matlab> (last request: 2018-09-10).
- [7] Pyrunner: "Compressed Sensing in Python", URL: <http://www.pyrunner.com/weblog/2016/05/26/compressed-sensing-python/> (last request: 2018-09-10).
- [8] Schultz, J.: "How Much Data is Created on the Internet Each Day?", URL: <https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/> (last request: 2018-09-10).
- [9] Wikipedia: "Tomographic reconstruction", URL: [https://en.wikipedia.org/wiki/Tomographic\\_reconstruction#Back\\_Projection\\_Algorithm\[2\]](https://en.wikipedia.org/wiki/Tomographic_reconstruction#Back_Projection_Algorithm[2]) (last request: 2018-09-05).



© 2018 Authors. Published by the University of Novi Sad, Faculty of Technical Sciences, Department of Graphic Engineering and Design. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license 3.0 Serbia (<http://creativecommons.org/licenses/by/3.0/rs/>).