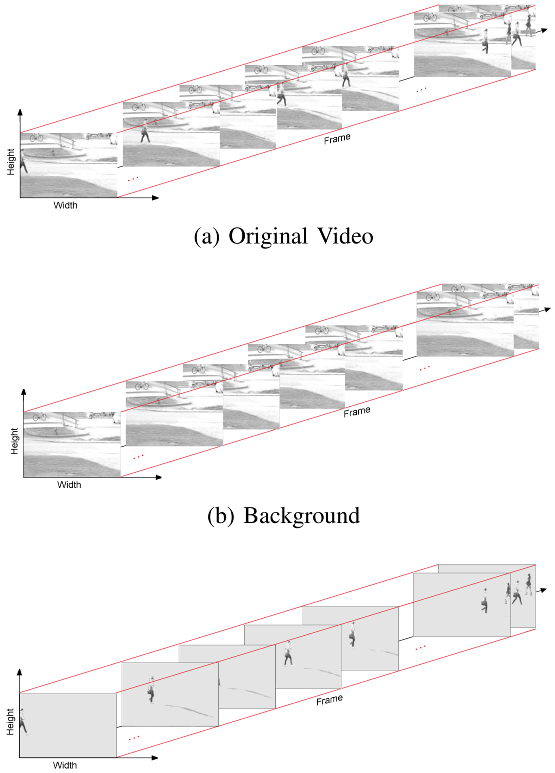
Algorithms and Tools for Big Data Application : Comparison of Tensor and Matrix based Approach in Foreground\Background Separation

Suhail Ahamed
Communication Research Laboratory
Ilmenau University of Technology
Ilmenau, Germany
Email: suhail.ahamed@tu-ilmenau.de

***Abstract* - Identifying moving objects in a video sequence is a fundamental and critical task in many computer-vision applications. This paper evaluates and compares the Matrix based compressed sensing approach and Tensor based Compressed Sensing approach of separating a video sequence into a background sequence and foreground sequence that consists of one or more moving objects/regions. Simulation results show that the Tensor based approach has better Background subspace estimation than the Matrix based approach. However, the Matrix based approach has better Foreground separation.**

I. Introduction

Identifying and tracking moving objects in a video sequence is a fundamental and critical task in many video applications. In surveillance, it enables us to identify and track foreign objects. This is important for security in train stations and airports, where unattended luggage can be a major hazard. In Sports, background subtraction is used when important decisions need to be made quickly. In tennis it is used for 'Hawk-Eye' which has become a key part of the game. In optical motion capture, background subtraction is used in animation to give a character life and personality. An actor is filmed on up to 200 cameras which monitor their movements precisely, using background subtraction these movements can then be extracted and translated onto the character. Since background subtraction is often the first step in many applications, it is important that the extracted foreground pixels are accurately separated from the background image. A good background subtraction algorithm should include fast adaptation to changes in environment, robustness in detecting objects moving at different speeds, and low implementation complexity.



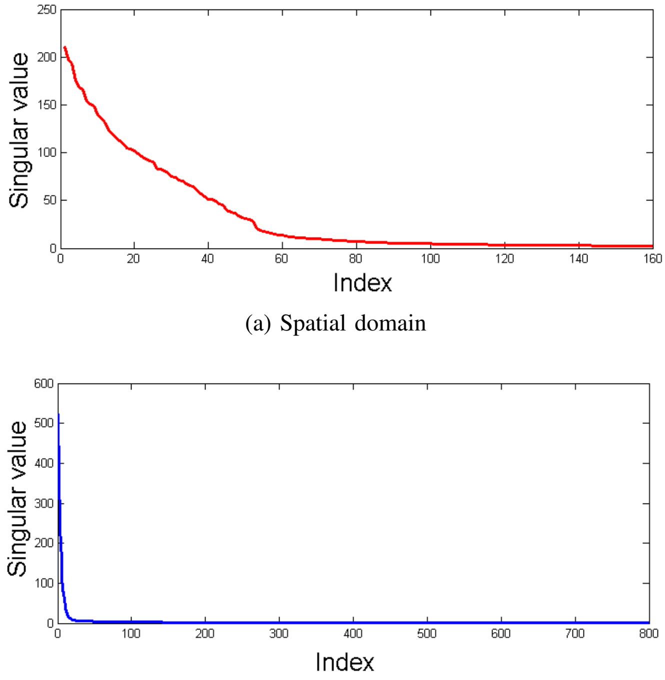
(a) Original Video

(b) Background

(c) Foreground

Fig. 1

A classical approach to identify the moving objects is to compare video frames to their background image. This background image can be obtained using several approaches. For eg., the mean and median filters can



(a) Spatial domain

(b) Temporal domain

Fig. 2: (a) exhibits the distribution of the singular values of one frame image. (b) exhibits the distribution of the singular values of the matrix obtained by vectorizing each frame in the background volume.

be used to extract the background image. Pixels in the current frame that deviate significantly from the background are considered to be moving objects. The basic idea of these statistical approaches have been discussed in [5]. These statistical approaches were adopted in our previous work [8] and their performance was compared to the compressed sensing based approach [1], via numerical simulations. It was shown that the compressed sensing based approach outperforms the classical statistical approaches.

In compressed sensing, it is assumed that the video is a sum of sparse components (foreground image) and dense components (background image). The spatio-temporal correlation of the background is exploited to enhance these approaches. As shown in fig. 2, the singular values have a decaying property, implying the intrinsic spatial as well as temporal low-dimensional structure over the frame. The state of the art method in matrix based compressed sensing approach has been designed in [1].

Tensors have shown to have a significant improvement in signal subspace estimation [3]. They can be used to further enhance the estimation of background subspace. In this paper we implement the Tensor based compressed sensing algorithm and compare the results to the Matrix based approach.

TABLE I: Notations

$\mathcal{X}$, X, $\mathbf{x}$, x	Tensor, Matrix, vector, scalar
$[\mathcal{X}]_{(n)}$	n-mode unfolding (Forward / MATLAB) of the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2, \cdots, \times I_N}$, obtained by arranging the n-mode vectors as the columns of the resulting matrix $\in \mathbb{R}^{I_n \times \Pi_{k \neq n} I_k}$
$r_1, r_2, \cdots, r_N$	multilinear rank, where r_n = Rank$\Big([\mathcal{X}]_{(n)}\Big)$
$\mathcal{Y} = \mathcal{X} \times_n U$	n-mode multiplication of $\mathcal{X}$ and U with matrix representation $[\mathcal{Y}]_{(n)} = U \cdot [\mathcal{X}]_{(n)}$
$\mathcal{X}^{[s]}, X^{[s]}$	low-rank approximated versions of tensor $\mathcal{X}$ and matrix X
$Z = X \otimes Y$	Kronecker product of matrices $X \in \mathbb{R}^{m \times n}$ amd $Y \in \mathbb{R}^{p \times q}$, resulting in the matrix $Z \in \mathbb{R}^{mp \times nq}$

Paper Organization: We give the system model of all the approaches Sec II. The simulation results are discussed in in Sec III. It also evaluates the performance of two approaches. Conclusions and future work are discussed in Sec IV.

II. System Model

Both the algorithms include a training phase, followed by an data processing phase. The algorithms differ only in the training phase and are similar in the data processing procedure. During training, a given sequence consisting only of the background, is used to estimate the background subspace. The dimensions of the background is $w \times h \times T$, where w is the width, h is the height and T is the number of frames in the training video sequence. This can be thought of as a 3^{rd} order tensor $\mathcal{X} \in \mathbb{R}^{w \times h \times T}$.

A. Matrix based subspace estimation

The matrix based method reshapes the 3-dimensional initial training sequence $\mathcal{X} \in \mathbb{R}^{w \times h \times T}$ into a 2-dimensional matrix $X \in \mathbb{R}^{(w \cdot h) \times T}$, which can be seen as the 3-mode unfolding (Forward/MATLAB) transpose of the tensor $\mathcal{X}$. This unfolded matrix X is used to compute U_s, the approximate basis for the background, i.e. $U_s = approx - basis(X, b\%)$. This is done by first computing the Singular Value Decomposition (SVD) of X, i.e.

$$X = U \cdot \Sigma \cdot V^H \tag{1}$$

where $U \in \mathbb{R}^{(w\cdot h)\times(w\cdot h)}$ and $V \in \mathbb{R}^{T\times T}$ are unitary matrices and $\Sigma \in \mathbb{R}^{(w\cdot h)\times T}$ is a diagonal matrix containing the singular values. Low-rank approximation is then computed by keeping the singular values that correspond to $b\%$ of the energy.

$$X \approx U_s \cdot \Sigma_s \cdot V_s^H \tag{2}$$

where $U_s \in \mathbb{R}^{(w\cdot h)\times \hat{r}}$, $V \in \mathbb{R}^{T\times \hat{r}}$ and $\Sigma \in \mathbb{R}^{\hat{r}\times \hat{r}}$. Note that $\hat{r} = rank(U_s)$, is the approximated low rank of$X$. We use b% = 95% or b% = 99.99% depending on whether the low-rank part is approximately low-rank or almost exactly low-rank.

B. Tensor based subspace estimation

The tensor based approach computes the Higher-order Singular Value Decomposition (Tucker3) of $XX \in \mathbb{R}^{w\times h\times T}$ and unitary matrices are used to estimate the background subspace. The "1-space", "2-space" and "3-space" unitary bases of XX are obtained from the (matrix) SVDs of the n-mode unfoldings of XX.

$$[\mathcal{X}]_{(1)} = U_1 \cdot \Sigma_1 \cdot V_1^H \tag{3}$$

$$[\mathcal{X}]_{(2)} = U_2 \cdot \Sigma_2 \cdot V_2^H \tag{4}$$

$$[\mathcal{X}]_{(3)} = U_3 \cdot \Sigma_3 \cdot V_3^H \tag{5}$$

where, $U_1 \in \mathbb{R}^{w\times w}$, $U_2 \in \mathbb{R}^{h\times h}$ and $U_3 \in \mathbb{R}^{T\times T}$. The HOSVD of $\mathcal{X}$ is given by,

$$\mathcal{X} = \mathcal{S} \times_1 U_1 \times_2 U_2 \times_3 U_3 \tag{6}$$

The core tensor is calculated by the following equation,

$$\mathcal{S} = \mathcal{X} \times_1 U_1^H \times_2 U_2^H \times_3 U_3^H \tag{7}$$

Here SS is a full-rank 3^{rd} order tensor. We then compute the low-rank approximation (truncated HOSVD) of XX by keeping $b\%$ energy of Σ_1, Σ_2 and Σ_3.

$$\mathcal{X} \approx \mathcal{S}^{[s]} \times_1 U_1^{[s]} \times_2 U_2^{[s]} \times_3 U_3^{[s]} \tag{8}$$

where, and $U_1^{[s]} \in \mathbb{R}^{w\times r_1}$, $U_2^{[s]} \in \mathbb{R}^{h\times r_2}$ and $U_3^{[s]} \in \mathbb{R}^{T\times r_3}$. The subspace estimation is given by,

$$\mathcal{U}^{[s]} = \mathcal{S}^{[s]} \times_1 U_1^{[s]} \times_2 U_2^{[s]} \times_3 \left(\Sigma_3^{[s]}\right)^{-1} \tag{9}$$

A link between the SVD-based and HOSVD-based subspace estimation enables us to calculate the subspace easily, without computing core tensor, SS. It is given by,

$$[\mathcal{U}^{[s]}]'_{(3)} = (T_1 \otimes T_2) \cdot U_s \tag{10}$$

where,

$$T_i = U_i^{[s]} \cdot {U_i^{[s]}}^H \tag{11}$$

The tensor based estimated subspace, $[UU^{[s]}]^T_{(3)} \in \mathbb{R}^{(w\cdot h)\times r_3}$ represents an improved background subspace estimate. Note that the condition $r_3 < max(w, h)$ has been satisfied.

C. Data processing

After the subspace estimation via the matrix and tensor based approaches, each frame of the real video sequence containing the foreground is projected onto the estimated subspaces (U_s and $[UU^{[s]}]^T_{(3)}$). To simplify our discussion, we use U_0 to represent the subspace of both the approaches.

Here, the video is treated to be composed of a slowly changing low-dimensional subspace (foreground) and a sparse component (background). Another way to see this is that the frame at time t ($\mathbf{x}_t$) is an $(w \cdot h)$ dimensional vector which can be decomposed as

$$\mathbf{x}_t := \mathbf{b}_t + \mathbf{f}_t \tag{12}$$

where $\mathbf{b}_t \in \mathbb{R}^{(w\cdot h)}$ is the low-dimensional subspace and $\mathbf{f}_t \in \mathbb{R}^{(w\cdot h)}$ is the sparse component. Note that $\mathbf{x}_t \in \mathbb{R}^{(w.h)}$ is the vectorized version of the matrix $X_t \in \mathbb{R}^{(w\times h)}$. The goal is to recursively estimate $\mathbf{b}_t$ and $\mathbf{f}_t$. Recursively meaning: use $\mathbf{f}_{t-1}$, $\mathbf{b}_{t-1}$ and the subspace to estimate $\mathbf{b}_t$ and $\mathbf{f}_t$. Let Φ_t be space orthogonal to the subspace, given by :

$$\Phi_t := (I - U_0 U_0') \tag{13}$$

At each time t, compressed sensing involves 3 steps:

(a) **Perpendicular Projection:** In the first step, project the vector $\mathbf{x}_t$, into the space orthogonal to range (U_0) to get the projected vector,

$$\mathbf{y}_t := \Phi_t \mathbf{x}_t. \tag{14}$$

(b) **Sparse Recovery (recover $\mathbf{f}_t$) :** Multiplying the obtained orthogonal complement to the current frame (x_t) will eliminate the background since $(I - U_0 U_0')\mathbf{b}_t$ is small. $\mathbf{f}_t$ can be recovered by solving :

$$min_{\mathbf{z}} \|\mathbf{z}\|_1 s.t. \|\mathbf{y}_t - \Phi_t \mathbf{z}\|_2 \leq \xi \tag{15}$$

After this, compressed sensing algorithm, e.g., basis pursuit, is used to recover the s_t. ξ here is the constraint equal to $\|\hat{\beta}_t\|_2$. $\hat{\beta}_t = \Phi_t \mathbf{b}_{t-1}$.

(c) **Recover $\mathbf{l}_t$**: The estimate $\mathbf{f}_t$ is used to estimate $\mathbf{b}_t$ as $\mathbf{b}_t = \mathbf{x}_t - \mathbf{f}_t$. Thus, if $\mathbf{f}_t$ is recovered accurately, so will $\mathbf{b}_t$.

III. Simulation and Experimental Results

In this section, we conduct simulations to compare the performance of the Matrix based Compressed Sensing and the Tensor based Compressed sensing. The comparison has been performed in two ways. Firstly, we analyze the ability of the algorithms to precisely estimate the background. Next we compare their abilities to extract the foreground from the video sequence. All the experiments are performed using MATLAB (R2014a) on a workstation with i7 Intel processor of 2.00 GHz and 8 GB of RAM equipped with Windows 8 OS.

We measure the accuracy of background estimation by the Peak Signal-to-Noise Ratio (PNSR) [7], which measures the structural similarity of two images; the higher the PSNR value, better the estimation result. PSNR is defined via the mean square error (MSE). Given a noise-free $w \times h$ gray-scale image I and its noisy approximation K, MSE is defined as:

$$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{w}\sum_{h=1}^{h}[I(i,j) - K(i,j)]^2 \qquad (16)$$

The PSNR in dB is defines as,

$$\text{PSNR} = 10 \cdot log_{10}\left(\frac{MAX_I^2}{MSE}\right) \qquad (17)$$

MAX_I is the maximum possible pixel value of the image, i.e. 255.

To analyze the foreground extraction of the algorithms, we use the notions of precision and recall, which are defines as:

$$\text{precision} := \frac{\text{TP}}{\text{TP+FP}}, \text{recall} := \frac{\text{TP}}{\text{TP+FN}},$$

where TP, FP and FN mean the numbers of true positives, false positives and false negatives, respectively. For simplicity, instead of plotting precision/recall, we use a single measurement called F-measure that combines precision and recall together:

$$\text{Fmeasure} := 2 \times \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

A higher F-measure means better extraction of the foreground image.

A. Simulations

The testing data is extracted from SABS (Stuttgart Artificial Background Subtraction) dataset [6], an artificial dataset for pixel-wise evaluation of background models. We collect 10 - 800 frames (NoForegroundDay0001 - NoForegroundDay0800) as the training video sequence with just the background and 300 frames (Basic0001 - Basic0300) as the video sequence containing both background anf foreground. Then we scale each frame into an image with size of 120×160 for speeding the simulations. Similarly, we choose 300 frames (say, GT0801-GT1100) as the foreground from SABS-GT data as the given true background.

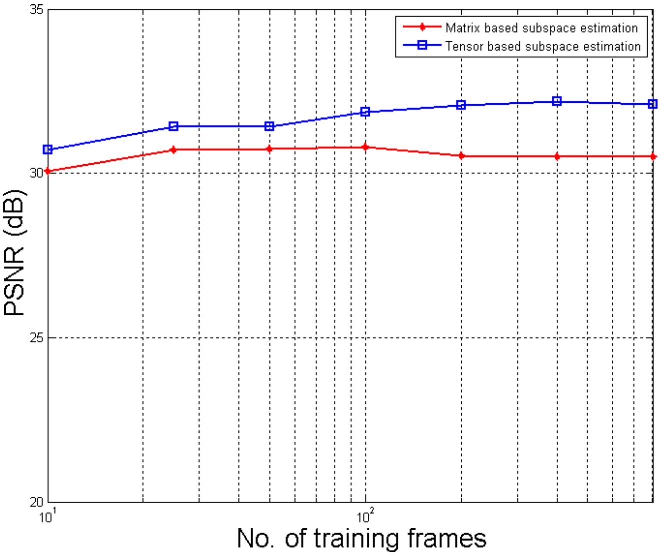


Fig. 3: PSNR values for different training lengths

Fig. 3 shows the performance comparison between the Matrix based subspace estimation and the Tensor based approach. The x-axis shows the number of training frames (10 - 800) and the y-axis shows the PSNR (in dB). The Tensor based subspace estimation clearly outperforms the Matrix based approach.

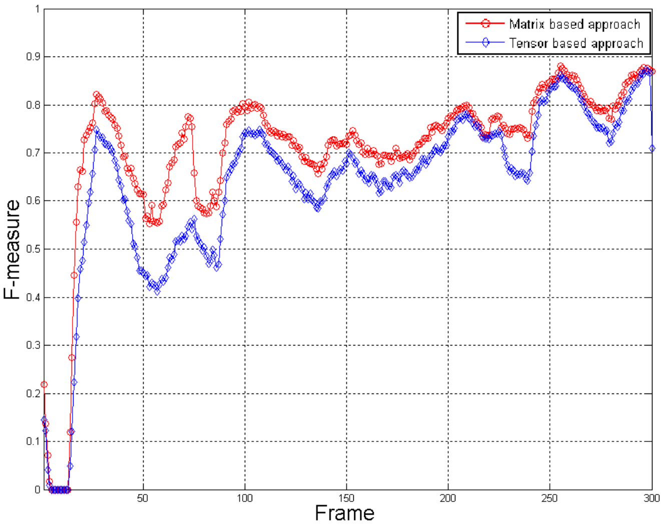


Fig. 4: F-measure for training length = 200 and videolength = 300

Next we compare the ability of the two approaches to separate the foreground from the video sequencee. Figures 4 and 5 show the performance comparison between two approaches for solving video background/foreground separation problem. The comparisons are in terms of the Fmeasure. In fig. 4 we see that the matrix based approach has a better performance. But, when we reduce the number of training frames (<50), the tensor based approach out-performs the matrix based one, as seen in fig. 5.

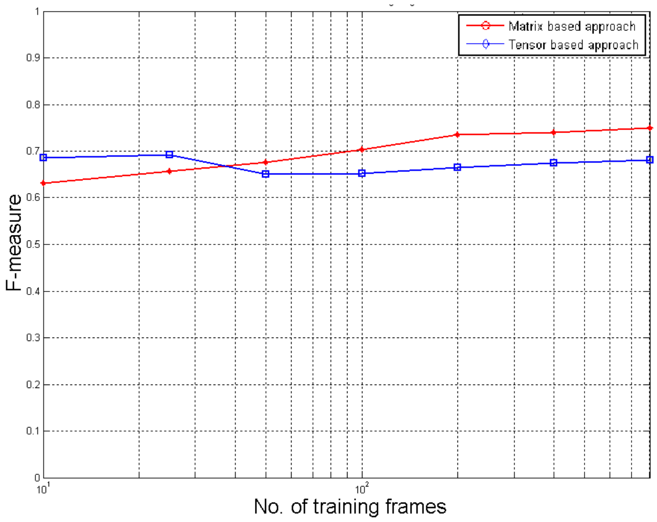


Fig. 5: Mean F-measure values for different training lengths

Fig. 6 shows 11 sampled frames of background and foreground from both the approaches. 200 frames (NoForegroundDay0001 - NoForegroundDay0200) of training sequence and 300 frames (Basic0001 - Basic0300) of the video sequence is used from the SABS training set.

IV. Conclusion

In this work we have compared the Matrix based compressed sensing approach and the Tensor based compressed sensing approach in the applications of foreground subspace estimation and background separation. It is shown that the Tensor based approach outperforms the Matrix based approach in estimating the background subspace for the simulated dataset. This improvement results from the fact that in tensors we take the special structure of the 3-dimensional structure into account while computing a low-rank approximation based on the HOSVD of the training sequence.

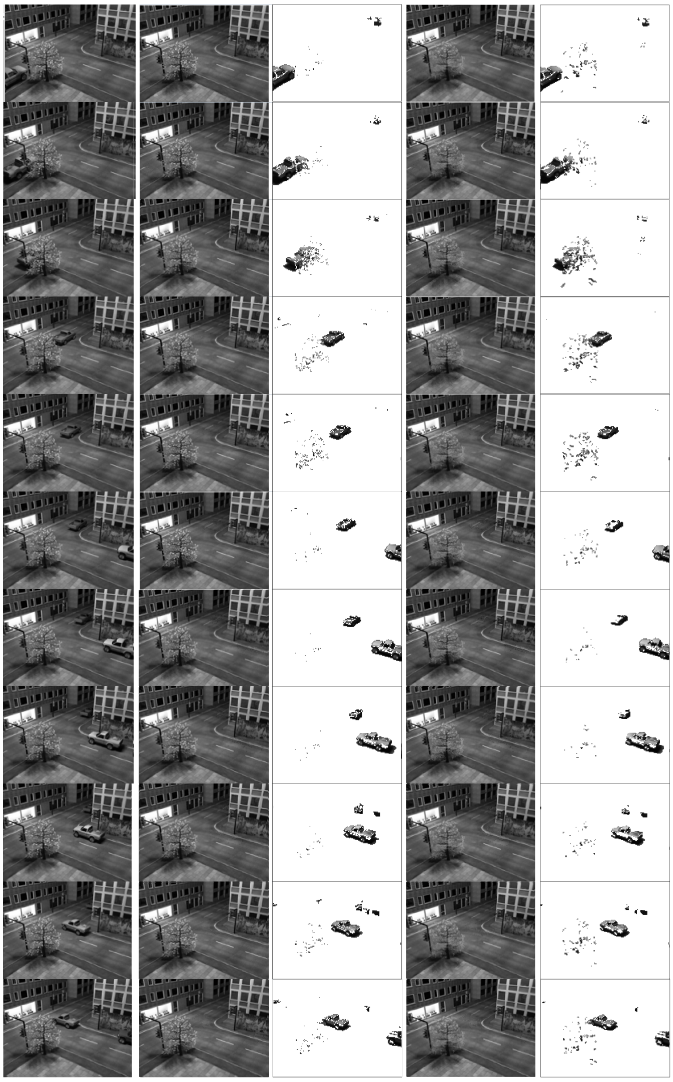


Fig. 6: Comparison of the visual results from the two methods. The first column is the real video sequence. The second and third column show the foreground and background from the matrix based approach. The fourth and fifth column show the foreground and background from the tensor based approach.

However, the matrix based approach performed better in separating the foreground from the background. But, when the number of training frames is reduced, the tensor based approach yields better results.

Tensors were used to estimate the background subspace, but the foreground detection algorithm was based on matrices. We can improve upon these methods by using tensors to separate foreground.

References

[1] Han Guo, Chenlu Qiu, Namrata Vaswani, ”An Online Algorithm for Separating Sparse and Low-dimensional Signal Sequences from their Sum”, *IEEE Transactions Signal Processing*, vol. 62, pp. 4284-4297, Jun. 2014

[2] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, ”Recursive robust PCA or recursive sparse recovery in large but structured noise”, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.5954-5958, May 2013

[3] Martin Haardt, Florian Roemer, and Giovanni Del Galdo, ”Higher-Order SVD-Based Subspace Estimation to Improve the Parameter Estimation Accuracy in Multidimensional Harmonic Retrieval Problems”, *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, VOL. 56, NO. 7, JULY 2008

[4] Andrew E. Waters, Aswin C. Sankaranarayanan, Richard G. Baraniuk, ”SpaRCS: Recovering Low-Rank and Sparse Matrices from Compressive Measurements,” *Neural Information Processing Systems (NIPS)*, Dec. 2011

[5] Birgi Tamersoy, ”Background Subtraction”, Sep. 2009

[6] Brutzer, Sebastian and Höferlin, Benjamin and Heidemann, Gunther, ”Evaluation of Background Subtraction Techniques for Video Surveillance”, in *IEEE Computer Vision and Pattern Recognition (CVPR)*, S.1937-1944, 2011

[7] Q. Huynh-Thu and M. Ghanbari, ”Scope of validity of PSNR in image/video quality assessment”, *ELECTRONICS LETTERS*, Vol. 44 No. 13, June 2008

[8] S. Ahamed, ”Algorithms and tools for Big Data Application”, TU Ilmenau, 2015 (No public release).

[9] Software: Ilmenau Tensor-Toolbox, contributors include: M. Weis, G. Del Galdo, F. Roemer, TU Ilmenau, CRL, (no public release).