CEIG - Spanish Computer Graphics Conference (2015) , pp. 1–10
Jorge Lopez-Moreno and Mateu Sbert (Editors)

Compressive High Speed Video Acquisition

Ana Serrano[1] Diego Gutierrez[1] Belen Masia[1,2]

[1] Universidad de Zaragoza [2] Max Planck Institute for Informatics

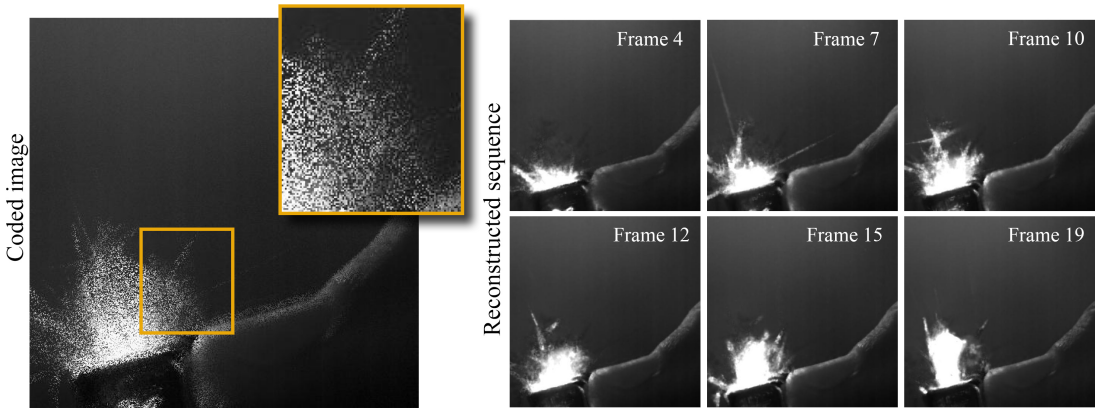

Figure 1: *Reconstruction of a high speed video sequence from a single, temporally-coded image using compressive sensing and an overcomplete learned dictionary. The sequence shows a lighter igniting. Left: Coded image, from which 20 individual frames will be reconstructed; inset shows a close-up of the coded temporal information. Right: Six of the 20 reconstructed frames.*

Abstract

Traditional video capture is limited by the trade-off between spatial and temporal resolution. When capturing videos of high temporal resolution, the spatial resolutions decreases due to bandwidth limitations in the capture system. Achieving both high spatial and temporal resolution is only possible with highly specialized and very expensive hardware; although the bandwidth is higher, the same basic trade-off remains. In this paper, we make use of a single-shot, high-speed video capture system, in order to overcome this limitation. It is based on compressive sensing, and relies on dictionary learning for sparse video representation. This allows capturing a video sequence by coding the temporal information in a single frame, and then reconstructing the full video sequence from this single coded image. We perform an in-depth analysis of the parameters of influence in the system, providing insights for future developments of similar systems.

Categories and Subject Descriptors (according to ACM CCS): I.4.1 [Computer Graphics]: Digitization and Image Capture—Sampling

1. Introduction

During the last years, high speed video capture technologies have been growing due to the neccessity of capturing information at temporal and spatial high resolution in scientific imaging, or industrial processes, among other areas. However, traditional cameras face a trade-off between these two resolutions, making it very difficult to capture high speed video at high spatial resolutions. This trade-off is determined by hardware restrictions, such as readout and analog-to-digital conversion times of the sensors; despite recent improvements in capture devices, this trade-off still represents a drawback.

Recent work tries to overcome these hardware limitations either with hardware-based approaches such as the camera

array prototype proposed by Willburn et al. [WJV*04], or with software-based approaches, like the work of Gupta et al. [GBD*09], where they propose to achieve high resolution videos with a combination of low resolution videos and a few key frames at high resolution. Recently, Liu et al. [LGH*13, HGG*11] presented a novel approach based on the emerging field of compressive sensing. This technique allows to fully recover a signal even when sampled at rates lower than the Nyquist-Shannon theorem, provided the signal is sufficiently sparse in a certain domain. They rely on this technique to selectively sample pixels at different time instants, thus coding the temporal information of a frame sequence in a single image. They then recover the full frame sequence from that image. Their key assumption is that the time varying appearance of scenes can be represented as a sparse linear combination of elements of an overcomplete basis.

In this paper we build on this work, extending their simulation results and performing an in-depth analysis of several parameters of influence in their framework, providing insights on design choices for improved performance of the framework. We further explore the existence of a good predictor of the quality of the reconstructed video. In particular, we make the following contributions:

- We introduce the Lars/lasso algorithm for the training and reconstruction, and show how it improves the quality of the results
- We present a novel algorithm for choosing the training blocks, which improves performance as well as reconstruction time
- We show a novel analysis of the input videos, which identifies which characteristics of the input videos will penalize performance, and provides insights about how to modify the framework accordingly

2. Related Work

The theory of compressive sensing has raised interest in the research community since its formalization in the seminal works of Candes et al. [CRT06] and Donoho [Don06]. Numerous works in recent years have been devoted to applying this theory to several fields. One of these fields of application is image and video acquisition, being one of the most significant works the *Single Pixel Camera* of Wakin et al. [WLD*06], where they propose a camera prototype with only one pixel that allows the reconstruction of complete images acquired with several captures under different exposition patterns. Other examples in imaging include the work of Marwah et al. [MWBR13], in which they achieve *light field* acquisition from a single coded image; or the capture of hyperspectral high resolution images proposed by Lin et al. [LLWD14], again coding the hyperspectral information within a single image by using compressive sensing.

High temporal and spatial resolution video acquisition has been attempted with several approaches. Gupta et al. [GBD*09] propose a method to recover high spatial resolution videos from low resolution sequences and a few key frames captured at a higher resolution but in order to do this they need two sequences, one with high spatial resolution and low temporal resolution and vice versa. Wilburn et al. [WJV*04] propose in their work a hardware-based approach with a dense camera array prototype. They set different time windows for every camera of the array and then align all the different views, however this approach is limited by the size and the complexity of the hardware.

On the other hand, coded exposures have been not only used in compressive sensing, but also widely as a way to improve some aspects of image and video acquisition in the field of computational photography. The objective is to code the light before it reaches the sensor either with coded apertures or shutter functions. Raskar et al. [RAT06] propose the use of a *flutter shutter* to avoid motion-blur in image capture. With the same purpose Gu et al. [GHMN10] propose the *coded rolling shutter*, an improvement to the conventional *rolling shutter*. Alternatively, codes in the spatial domain have been used to avoid defocus blur [MCPG11, MPCG12], or recover depth information [LFDF07, ZLN09].

Finally, of particular interest to our work is that of Liu et al. [LGH*13, HGG*11] who introduce two key insights for capturing high speed video with a compressive sensing framework. On the one hand they propose a system able to recover a video sequence from a single image with coded exposure through dictionary learning. On the other, they present a new shutter function based in a pseudo-random sampling pattern. We build on their work, analyze their system, and propose improved design choices.

3. Background on Compressive Sensing

The basic idea of compressive sensing states that under certain conditions, a signal can be completely captured even when sampled at rates lower than what the Nyquist-Shannon theorem dictates. In order to accomplish this, two conditions need to be satisfied, which are *Sparsity* and *Incoherence*. *Sparsity* means that the signal can be represented in some domain with only a few coefficients. This can be represented as:

$$X = \sum_{i=1}^{N} \psi_i \alpha_i \tag{1}$$

where X is the signal, in this case a video sequence, in its original domain; ψ_i are the elements of the basis that form the alternative domain; and α_i are the coefficients, which are in their majority zero or close to zero if the signal is sparse. Many natural signals, such are images or audio, can be considered sparse if represented in an adequate domain.

In order to satisfy the *Incoherence* condition, it is necessary to sample the signal with a particular pattern that

guarantees incoherence of such pattern with the chosen basis. The coherence between two pairs of bases measures the largest correlation between any two elements of those bases. For this purpose it has been demonstrated that a random sampling, in particular for Gaussian and binary distributions, yields a good grade of incoherence in an overall system formed by the sampling pattern and any basis.

The sampling process can be represented as:

$$Y = \phi X \tag{2}$$

where the video sequence is represented by $X \in \mathbb{R}^m$, the captured image is $Y \in \mathbb{R}^n$, with $n \ll m$, and $\phi \in \mathbb{R}^{n \times m}$ contains the sampling pattern which is called *measurement matrix*.

Finally, if both aforementioned conditions are fulfilled, the theory states that it is possible to perfectly reconstruct the original signal from the undersampled one acquired under an ideal scenario. For this step, we jointly consider the sampling process (Equation 2) together with the representation in the sparse dictionary (Equation 1), yielding the following formulation:

$$Y = \phi X = \phi \psi \alpha \tag{3}$$

with $\psi \in \mathbb{R}^{m \times q}$ being an overcomplete basis (also called *dictionary*) with q elements. If the original sequence X is $k-sparse$ in the domain of the basis formed by the *measurement matrix* ϕ and the dictionary ψ, it can be well represented by a linear combination of at most k coefficients in $\alpha \in \mathbb{R}^q$. Note that we are looking for an sparse solution; therefore, the search of the coefficients α has to be posed as a minimization problem. This optimization will search for the unknown α coefficients, seeking a sparse solution to Equation 3. This is typically formulated in terms of the L_1 norm, since L_2 does not provides sparsity and L_0 presents an ill-posed problem which is difficult to solve:

$$\min_{\alpha} \|\alpha\|_1 \; subject\ to \; \|Y - \phi\psi\alpha\|_2^2 \leq \varepsilon \tag{4}$$

where ε is the residual error. Once the α coefficients are known, we use them in Equation 1, together with the dictionary ψ, to recover the original signal.

The three key components in a compressive sensing framework are therefore the dictionary, the measurement matrix, and the reconstruction algorithm, we will analyze them in Section 4.

4. High speed video acquisition system

In this section we introduce the pipeline of a system based on compressive sensing applied to high speed video capture, including the reconstruction of a video from a single coded image, and the process of building a dictionary appropriate for high speed video representation. The pipeline is presented in Figure 2.

4.1. Learning high speed video dictionaries

We need a dictionary in which the signals of interest, in this case high speed videos, are sparse. The advantage of choosing an already existing basis is that usually these bases are mathematically well defined and their properties are known. However, since they are designed to be generic, they usually do not provide an optimal sparse representation for a specific set of signals of interest. A way to ensure that our set will be sparse in the basis domain is to train a dictionary specifically adapted for sparse video representation.

We learn fundamental building blocks (atoms) from high speed videos and create an overcomplete dictionary. For this purpose, we use the DLMRI-Lab implementation [RB11] of the K-SVD [AEB06] algorithm, which has been widely used in the compressive sensing literature, to train a high speed video basis with a varied set of videos of interest.

We train our dictionary with an acquired high temporal resolution video database with varied scenes recorded at 1000 frames per second. From them we have obtained a training set by splitting some of these videos into blocks of size $n = p_x \times p_y \times p_t$. Given a large collection of blocks, we have to choose a computationally affordable number of blocks as a training set. Most of these blocks will not have interesting features such as gradients or temporal events, therefore we would like to discard some of them while ensuring the presence of blocks with relevant information. In this work we propose an alternative to random selection that enforces this by giving certain priority to blocks with high variance. This is further explained in Section 5, where we also analyze the alternative choice of an existing base (DCT Type-II) instead of a trained dictionary.

4.2. Capturing coded images from high speed sequences

The measurement matrix introduced in Section 3, for the particular case of video sampling, consists on a coded exposure implemented as a shutter function that samples different time instants for every pixel. The final image is thus formed as the integral of the light arriving to the sensor for all the temporal instants sampled with the shutter function. This can be expressed with the following equation:

$$I(x,y) = \sum_{t=1}^{T} S(x,y,t)X(x,y,t) \tag{5}$$

where $I(x,y)$ is the captured image, S the shutter function and X the original scene. In a conventional capture system $S(x,y,t) = 1 \; \forall \, x,y,t$ but in this case the goal is to achieve a S function that fulfills the properties of a measurement matrix suitable for compressive sensing reconstruction. As mentioned in Section 3, an easy way to fulfill this requirements is to build a random sampling matrix. However, a fully-random sampling matrix cannot be implemented in current hardware, as explained below. Therefore, we use the shutter function proposed by Liu et al. [LGH*13, HGG*11] that

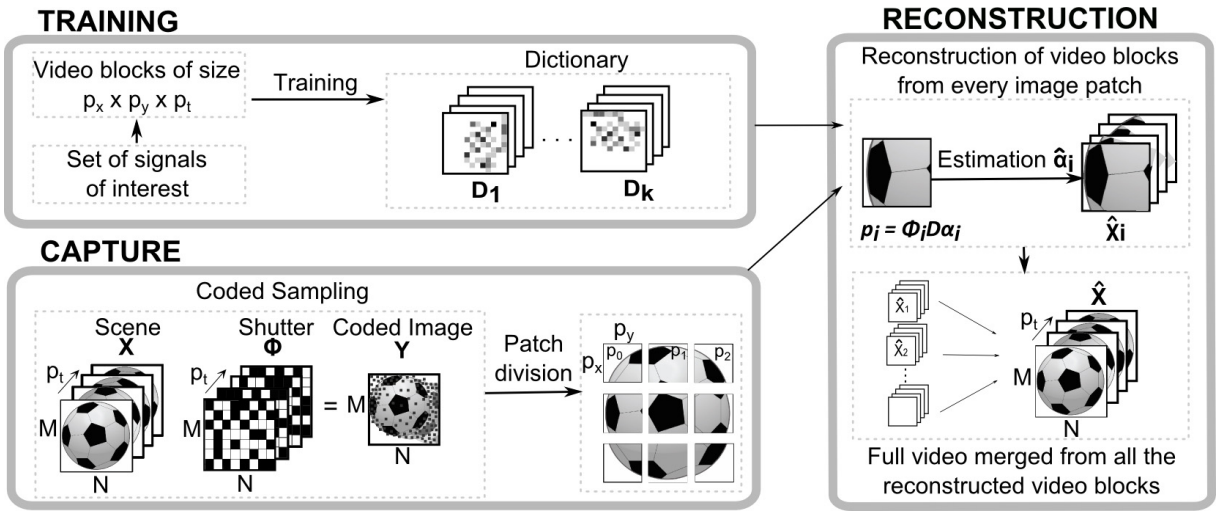

Figure 2: *Pipeline of the system for high speed high resolution video acquisition. Top left: Training of a dictionary D with k elements with a set of video blocks of size $p_x \times p_y \times p_t$ from the collection of signals of interest. Bottom left: Coded sampling of the original scene X with a measurement matrix ϕ resulting in the coded image Y, and division of this image in patches p_i appropriate for the size of the blocks used for training the dictionary. Right: Reconstruction of the video blocks $\hat{X}_i$ from each image patch p_i and merging of the blocks to obtain the full video $\hat{X}$. Adapted from [SGM15].*

tries to achieve randomness while imposing some restrictions to make a hardware implementation possible.

Hardware Limitations Image sensors are usually based on CMOS technology and a capture process comprises several processes requiring a certain amount of time, making difficult to build a function that samples randomly every pixel at every time instant. There are two key limitations: maximum integration time, i.e, the maximum time we can sample a pixel, which is limited by thermal noise in the sensor; and speed at which the shutter can be opened and closed, which is limited by the processing time of the camera (such as A-D conversion and readout time). Because of this, the shutter design for each pixel is limited to a single continous exposure and this exposure time has to be shorter than the integration time of the camera, that is, the shutter function design is limited to a single continuous integration bump for each pixel and this bump also has a limited duration. This function can be easily implemented in a *DMD* or an *LCoS* placed before the sensor, as proven by Liu et al. [HGG*11, LGH*13].

4.3. Reconstructing high speed videos from coded images

Once the dictionary and the measurement matrix are decided, we need to solve Equation 4 to estimate the α coefficients and thus be able to reconstruct the signal. Many algorithms have been developed for solving this minimization problem for compressive sensing reconstruction. In Section 5 we analyze the influence of this algorithm in the quality of the results by comparing two algorithms that had proven a good performance in similar problems: Orthogonal Matching Pursuit (OMP) [PRK93] and the LARS approximation for solving the Lasso [EHJT04]. We use the implementations available in the SPArse Modeling Software (SPAMS) [MBPS09, MBPS10].

5. Analysis of the system

In this section we perform an exhaustive analysis of the system presented in Section 4, exploring the parameters of influence. All the learned dictionaries presented in this section are trained with the same set of videos, which are included in the supplementary material. We analyze several parameters over a test set of six videos (see Figure 3) to find the parameter combination yielding the best results. Note that none of the testing videos are used during the training. In order to isolate the influence of each parameter in the framework, we maintain the rest of the parameters fixed while we vary the one being analyzed. The set of parameters derived from this analysis is presented in Section 6; these are also the parameters we use by default to perform every section of the analysis. We use as measures of quality the PSNR (Peak Signal to Noise Ratio), widely used in the signal processing literature. We also performed all tests with the MS-SSIM metric [Wan04], which takes into account visual perception, and found that it yielded results consistent with PSNR. Thus, for brevity, we only show results with PSNR.

5.1. Choosing a dictionary

In this section we compare our trained dictionary with a three dimensional DCT basis. We compare the two dictio-

Figure 3: *Sample frame extracted from each of the videos used in the analysis.*

naries under the same conditions, i.e., same number of elements of the dictionary and same block size. We can see in Figure 4 that the training dictionary performs better than the DCT basis except for the case of the video *Spring*. This video is consistently the result with worst quality, so the better performance of DCT can be arbitrary or due to different amounts of noise and artifacts in the reconstruction. Therefore, we choose the trained dictionary as the best alternative.

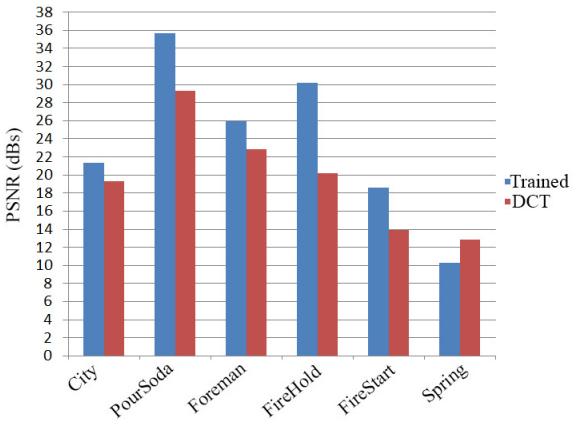

Figure 4: *Quality of the reconstructed videos (in terms of PSNR) for the set of analyzed videos reconstructed with a trained dictionary and with a three dimensional DCT basis under the same conditions (same dictionary elements and same block size). The trained dictionary consistently outperforms DCT, except for the Spring video, which can be considered an outlier (see text for details).*

5.2. Selection of training blocks

We use for the training a high speed video dataset and we divide each video into blocks. The size of these blocks is the size of the atoms of the resulting trained dictionary as well as the size of the video blocks the reconstruction is able to recover. The amount of blocks resulting from this division is unmanageable for the training algorithm; thus the dimensionality of the training set has to be reduced. The straightforward solution is to randomly choose a manageable amount of blocks. However, a high percentage of these blocks do not contain information about the scene (such as blocks corresponding to plain surfaces with no movement across the video). Aiming to improve the trained dictionaries, we explore several ways to select the blocks to use during the training, with the aim to avoid most of the blocks not containing any useful information. The methods analyzed are the following:

- **Random sampling**: The amount of training blocks are randomly selected from the original set.
- **Variance sampling**: We calculate the variance for each block and bin them in three categories (high, medium and low variance). Then we randomly select the same amount of blocks for every bin aiming to ensure the presence of high variance blocks in the resulting set.
- **Stratified gamma sampling**: We sort the blocks by increasing variance and sample them with a gamma curve ($f(x) = x^{\gamma}$). We analyze the effect of two possible curves: $\gamma = 0.7$ which yields a curve closer to a linear sampling, and $\gamma = 0.3$. The objective of the stratification is to ensure the presence of all the strata in the final distribution. We divide the range uniformly in the amount of desired final samples and we calculate thresholds for the strata applying the gamma function. Then we randomly choose a sample from every strata and remove that sample from the original set. Given some strata will be empty, this process repeats iteratively until the number of desired samples is reached.
- **Gamma sampling**: We choose directly samples from the original set following a gamma curve sampling. We also test two values for γ, $\gamma = 0.3$ and $\gamma = 0.7$.

We show in Figure 5 results from one of the six tested videos reconstructed with different dictionaries learned from blocks dragged from the same set of videos but with the different selection methods. Results for all the videos tested were consistent. *Random* and *Variance sampling* clearly outperform the other methods, with the *Variance sampling* yielding slightly better results. Additionally, as shown in Table 1, *Variance sampling* achieves a faster reconstruction time.

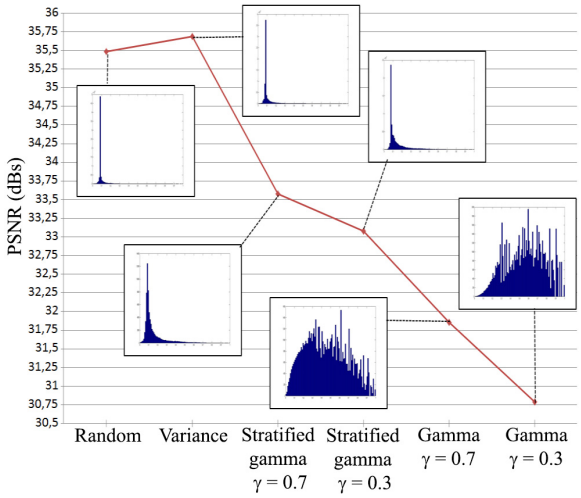

Figure 5: *Quality of the reconstruction (in terms of PSNR) for a sample video (PourSoda) as a function of the method used to select training blocks for learning the dictionary. For each method we show an inset with the histogram of variances of video blocks of the resulting training set.*

Method	**Time** (seconds)
Random	868.94
Variance	781.98
Stratified gamma 0.7	1295.61
Stratified gamma 0.3	971.25
Gamma 0.7	1001.48
Gamma 0.3	976.73

Table 1: *Mean reconstruction times obtained for each of the methods used to select the set of blocks for learning the dictionary. Times shown are the average across all six test videos.*

5.3. Reconstruction algorithm

As explained in Section 4, we need a reconstruction algorithm to recover the α coefficients that are multiplied by the dictionary to obtain the original scene from the coded image; this is posed as a minimization problem (see Equation 4). This algorithm is used in the reconstruction as well as in the training, since the training algorithm K-SVD solves the minimization problem as a step to update the dictionary. In this work, we analyze two algorithms for solving the minimization: Orthogonal Matching Pursuit [TG07] and the LARS solver for the Lasso [EHJT04]. In Figure 6 we show combinations of these two algorithms in the training and reconstruction steps. It can be seen that the reconstruction algorithm has a significant influence in the reconstruction step, with LARS-Lasso yielding the best performance.

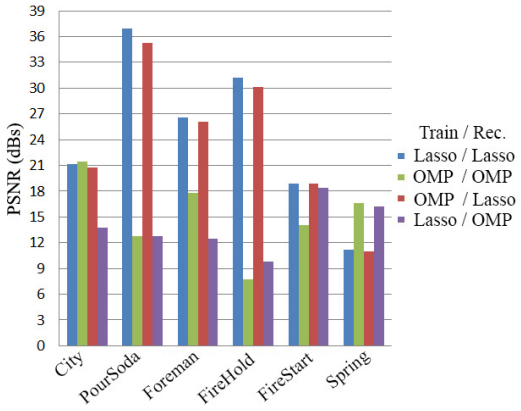

Figure 6: *Quality of the reconstruction (in terms of PSNR) for different combinations of the algorithms OMP and LARS-Lasso for the training/reconstruction steps for each of the analyzed videos. The best combination is obtained using the LARS-Lasso both for training and reconstruction.*

5.4. Measurement matrix

Choosing the best measurement matrix possible is crucial to achieve good results in a compressive sensing framework. As explained in Section 3, some properties need to be fulfilled by this matrix and at the same time we want it to be implementable in hardware. In this section we compare several measurement matrices [LGH*13, HGG*11] (see Figure 7):

- **Global shutter**: All the pixels are sampled over the integration time of the camera.
- **Flutter shutter** [RAT06]: The shutter entirely opens and closes over the integration time of the camera.
- **Rolling shutter**: Pixels are sampled sequentially by rows through time.
- **Coded rolling shutter** [GHMN10]: Variant of the rolling shutter. In this case the image is sub-divided and each part sampled with rolling shutter independently. For example, for an image sub-divided in two, first odd rows are sampled with rolling shutter, and then even rows.
- **Pixel-wise shutter** [LGH*13, HGG*11]: Each pixel is sampled over a fixed bump time shorter than the integration time of the camera, starting at different time instants. However, in order to obtain enough samples for the reconstruction a condition is imposed: Taking into account the size of the blocks we want to reconstruct, for every temporal instant at least one of the pixels from each block must be sampled. This is a way to ensure that every temporal instant is represented in every patch of the captured image. This can be expressed as:

$$X = \sum_{(x,y)\in p_j} S(x,y,t) \geq 1 \; for \; t = 1..T \qquad (6)$$

with T the temporal instants (or frames) to sample.

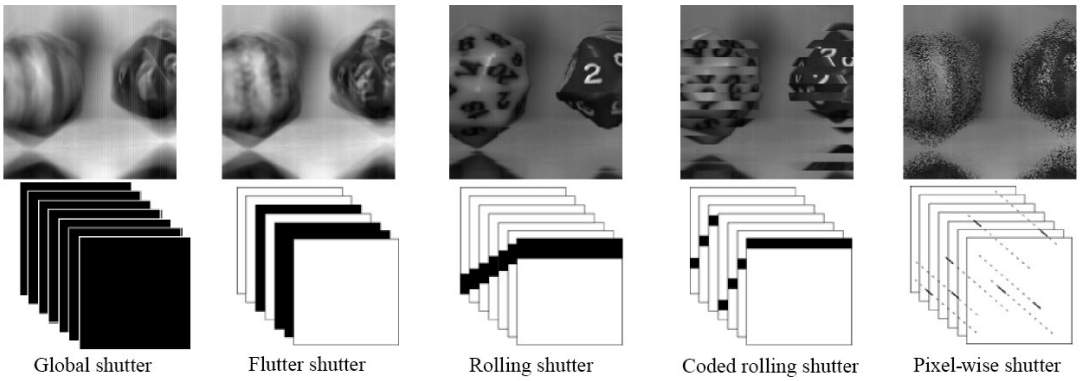

Figure 7: *Shutter functions and resulting coded images when sampling with them. Black pixels correspond to pixels being sampled (on) while white pixels correspond to pixels that are off.*

The results of the analysis of these measurement matrices are shown in Figure 8. The best results are obtained with the pixel-wise shutter, since it is specifically designed for the compressive sensing framework, as it guarantees that at least one pixel per patch is sampled for every frame.

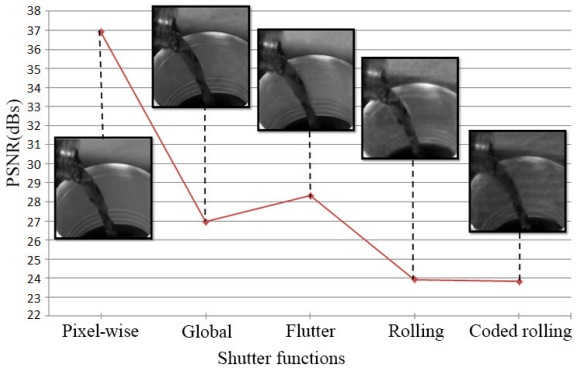

Figure 8: *Quality of the reconstruction (in terms of PSNR) using several measurement matrices. The insets show a reconstructed frame for every measurement matrix.*

6. Results and evaluation

In this section we present additional results, and also include a characterization of the input videos where we aim to identify which characteristics of a video may have an influence in the quality of the results of the reconstruction.

6.1. Characterization of input videos

We have performed a characterization over the set of videos we use to perform all the analysis aiming to find some insights about how the characteristics of these input videos influence the quality of the results. We have explored the following methods (see Figure 9):

- **Histogram of the high frequency energy density ratios** (H_{ratios}) for every video block. It is calculated by performing a one-dimensional DFT along the temporal dimension for every pixel of the block. Then, for each block, the DFTs corresponding to all its pixels are added and the ratio is calculated as the high frequency energy of the signal divided by the total energy. A threshold is used to determine what is high-frequency. We set this threshold experimentally to classify our set of signals correctly. Finally, we obtain the histogram with the values given for every block of the video and we calculate some standard statistical descriptors: mean, standard deviation, skewness and kurtosis. This ratio represents the percentage of high frequency energy to the total energy of the signal.
- **Histogram of variances** (H_{var}) for each video block. We calculate the variance for each 3-D block and its histogram, together with the same statistical descriptors as before: mean, standard deviation, skewness and kurtosis. We aim to obtain joint information about the temporal and spatial variation.
- **Sum of the difference between frames**. We sum the difference frame between all the consecutive frames in the video obtaining a single frame. Then we sum all the pixels of that image to obtain a single value. This values gives us an estimate of the amount of movement happening in the whole scene.

We have performed a correlation analysis with between the quality of the final result and all these characterization values using Pearson's [Pea95] and Spearman's [Spe04] correlation. The characterization method yielding the higher correlation is the standard deviation of the histogram of high frequency energy density ratios σ_{ratios}, with a Pearson's coefficient of $\rho_P = -0.9636$ and a p-value of 0.002 and a Spearman's coefficient of $\rho_S = -1$ and a p-value of 0.0028.

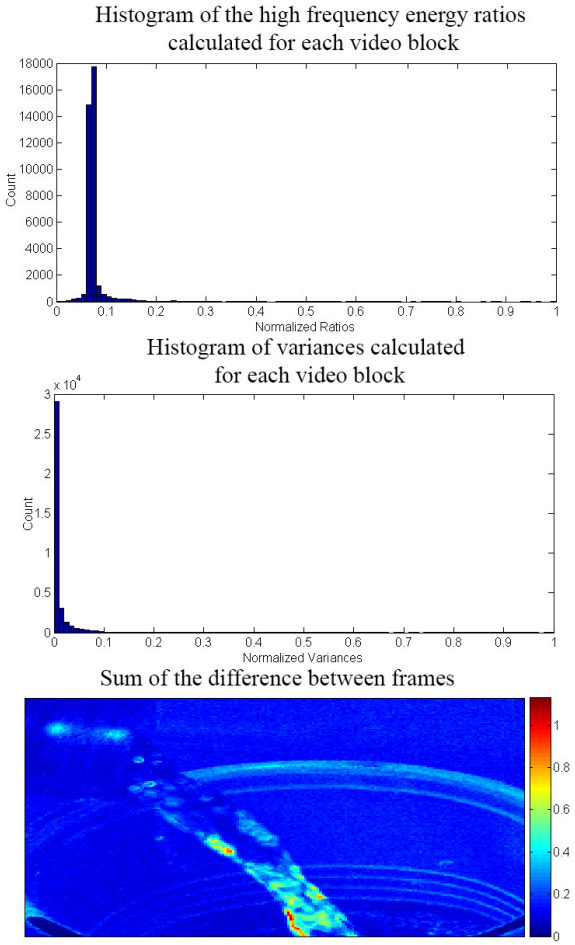

Figure 9: *Characterization of a sample video (PourSoda). We compute different statistics from the input videos and analyze their correlation with the quality of the reconstruction. From top to bottom: Histogram of the variances computed for every video block, histogram of the high frequency energy density ratios calculated for every video block, and sum of the difference between all consecutive frames of the video. Please refer to the text for details.*

Given these values, we consider it to be a good predictor of the quality of the reconstruction. Intuitively, this statistic tells us whether there is a high variance in the proportion of high temporal frequencies across the frames of the video. If this variance is high, then the reconstruction results are poorer, since temporal information with very different frequencies has to be coded within the same samples in the coded image.

6.2. Additional results

We present some results for three reconstructed sample videos. In Figure 10 we show 10 coded images, each one containing the necessary information to recover 20 frames of a sequence, therefore together forming a 200 frames video coded within the 10 frames. In Figures 1 and 11 we show some key frames of the reconstructed sequence corresponding to two of these coded images. Another result is presented in Figure 12, the sequence represents a flower toy moving. For this video we show the coded image together with three reconstructed frames.

For the reconstruction of all the video sequences presented in this section we use the combination of parameters that yield better results according to our analysis. These parameters are chosen as follows:

- Dictionary: Trained with the K-SVD algorithm, making use of the LARS-Lasso. The size of the atom is 7 $pixels \times$ 7 $pixels \times 20$ $frames$ and the training set is chosen with the *Variance sampling* method.
- Measurement matrix: We use the *pixel-wise* shutter function.
- Reconstruction: We solve the reconstruction minimization problem with the LARS-Lasso algorithm.

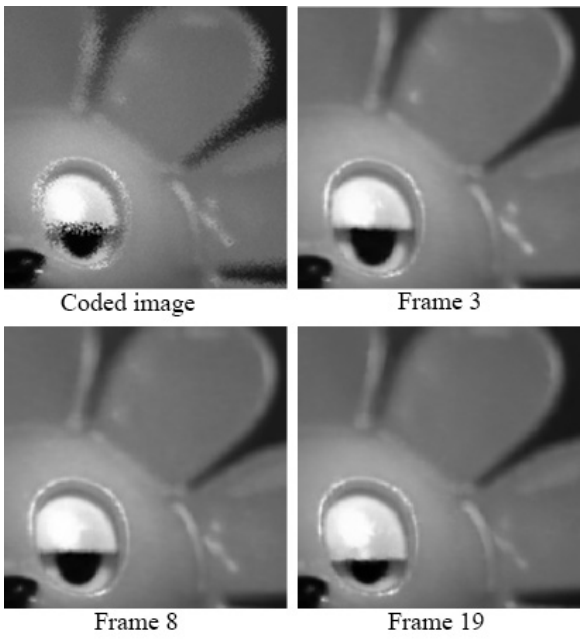

Figure 12: *Close-ups of coded image and three reconstructed frames (out of 20) for a video of a moving flower toy closing its eyes. The PSNR of the reconstructed video sequence is* 33.53 *dBs.*

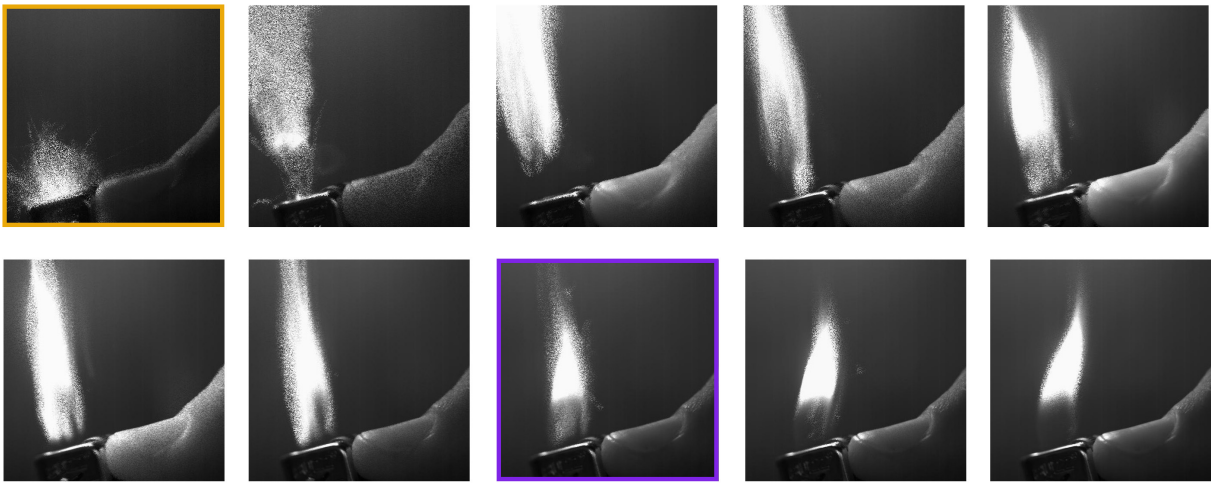

Figure 10: *Coded images for the reconstruction of a sequence of 200 frames. Each coded image contains the necessary information to recover 20 frames of a video sequence. In Figures 1 and 11 we show the reconstructed sequence frame by frame for the marked coded images.*

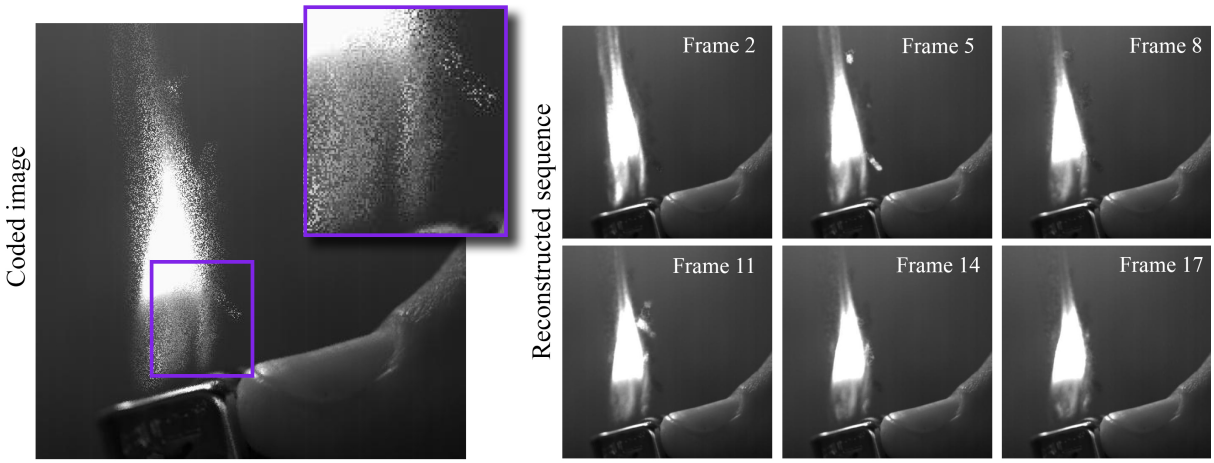

Figure 11: *Left: Coded image of a flame burning in a lighter, inset shows a close-up of the temporal information coded. Right: Sample frames (out of 20) of the reconstructed sequence. The PSNR of the reconstructed video sequence is* 29.09 *dBs.*

7. Conclusions

In this work, we have performed an in-depth analysis of the system proposed by Liu et al. [LGH*13, HGG*11], running extensive tests over a range of parameters of influence, possible algorithms of choice, and characteristics of the input videos. We have shown that there is room for improvement in a number of aspects of their framework, and we proposed alternatives for some of them that perform better, in particular the method for choosing a training set for the dictionary, and the reconstruction algorithm used. For the first one, we found that carefully choosing the training set can improve the performance of the framework, and for the second one, we have proven that the LARS-Lasso algorithm performs better than the OMP.

We have also computed a series of statistics from a collection of videos and provided some insights about the correlation of these statistics with the quality of the reconstruction obtained for those videos, in particular, we have shown that the standard deviation of the histogram of high frequency energy density ratios has a high correlation with the quality of the results.

8. Future Work

Our work shows that there is still room for improvement in several aspects of the framework. In particular, we would like to test some others methods for the selection of blocks for the training and analyze their influence in our framework, such as the *coresets* proposed by Feldman et al. [FFS13].

On the other hand, we would like to explore more deeply on the influence of the characteristics of input videos in the system, since this can provide insights about how to improve both training and reconstruction algorithms.

9. Acknowledgements

We would like to thank the Laser & Optical Technologies department from the Aragon Institute of Engineering Research (I3A), as well as the Universidad Rey Juan Carlos for providing a high-speed camera and some of the videos used in this paper. This research has been partially funded by Spanish Ministry of Science and Technology (project LIGHTSLICE), the BBVA Foundation, the Max Planck Center for Visual Computing and Communication, and a gift from Adobe. Diego Gutierrez is additionally supported by a Google Faculty Research Award.

References

[AEB06] AHARON M., ELAD M., BRUCKSTEIN A.: K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing 54* (2006), 4311–4322. 3

[CRT06] CANDES E., ROMBERG J., TAO T.: Robust incertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory 52* (2006), 489–509. 2

[Don06] DONOHO D.: Compressed sensing. *IEEE Transactions on Information Theory 52* (2006), 1289–1306. 2

[EHJT04] EFRON B., HASTIE T., JOHNSTONE I., TIBSHIRANI R.: Least angle regression. *Annals of statistics 32* (2004), 407–499. 4, 6

[FFS13] FELDMAN D., FEIGIN M., SOCHEN N.: Learning big (image) data via coresets for dictionaries. *Journal of Mathematical Imaging and Vision 46*, 3 (2013), 276–291. 10

[GBD*09] GUPTA A., BHAT P., DONTCHEVA M., CURLESS B., DEUSSEN O., COHEN M.: Enhancing and experiencing spacetime resolution with videos and stills. In *IEEE International Conference on Computational Photography* (2009). 2

[GHMN10] GU J., HITOMI Y., MITSUNAGA T., NAYAR S.: Coded rolling shutter photography: Flexible space-time sampling. In *IEEE International Conference on Computational Photography* (2010). 2, 6

[HGG*11] HITOMI Y., GU J., GUPTA M., MITSUNAGA T., NAYAR S.: Video from a Single Coded Exposure Photograph using a Learned Over-Complete Dictionary. In *IEEE International Conference on Computer Vision (ICCV)* (2011), pp. 287–294. 2, 3, 4, 6, 9

[LFDF07] LEVIN A., FERGUS R., DURAND F., FREEMAN W. T.: Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics 26* (2007). 2

[LGH*13] LIU D., GU J., HITOMI Y., GUPTA M., MITSUNAGA T., NAYAR S.: Efficient Space-Time Sampling with Pixel-wise Coded Exposure for High Speed Imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36* (2013), 248–260. 2, 3, 4, 6, 9

[LLWD14] LIN X., LIU Y., WU J., DAI Q.: Spatial-spectral encoded compressive hyperspectral imaging. *ACM Transactions on Graphics 33* (2014), 1–11. 2

[MBPS09] MAIRAL J., BACH F., PONCE J., SAPIRO G.: Online dictionary learning for sparse coding. In *International Conference on Machine Learning* (2009), pp. 689–696. 4

[MBPS10] MAIRAL J., BACH F., PONCE J., SAPIRO G.: Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research 11* (2010), 16–60. 4

[MCPG11] MASIA B., CORRALES A., PRESA L., GUTIERREZ D.: Coded apertures for defocus blurring. In *Ibero-American Symposium in Computer Graphics* (2011). 2

[MPCG12] MASIA B., PRESA L., CORRALES A., GUTIERREZ D.: Perceptually-optimized coded apertures for defocus deblurring. *Computer Graphics Forum 31* (2012). 2

[MWBR13] MARWAH K., WETZSTEIN G., BANDO Y., RASKAR R.: Compressive light field photography using overcomplete dictionaries and optimized projections. *ACM Transactions on Graphics 32* (2013), 1–11. 2

[Pea95] PEARSON K.: Note on regression and inheritance in the case of two parents. In *Proceedings of the Royal Society of London* (1895). 7

[PRK93] PATI Y., REZAIIFAR R., KRISHNAPRASAD P.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Conference on Signals, Systems and Computers* (1993), vol. 1, pp. 40–44. 4

[RAT06] RASKAR R., AGRAWAL A., TUMBLIN J.: Coded exposure photography: Motion deblurring using fluttered shutter. *ACM Transactions on Graphics 25* (2006), 795–804. 2, 6

[RB11] RAVISHANKAR S., BRESLER Y.: Mr image reconstruction from highly undersampled k-space data by dictionaty learning. *IEEE Transactions on Medical Imaging 30* (2011), 1028–1041. 3

[SGM15] SERRANO A., GUTIERREZ D., MASIA B.: An in-depth analysis of compressive sensing for high speed video acquisition. In *IEEE International Conference on Computational Photography. Poster* (2015). 4

[Spe04] SPEARMAN C.: The proof and measurement of association between two things. *The American Journal of Psychology 15* (1904), 72–101. 7

[TG07] TROPP J., GILBERT A.: Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory 53* (2007), 4655–4666. 6

[Wan04] WANG Z.: Multiscale structural similarity for image quality assessment. In *Conference on Signals, Systems and Computers* (2004), vol. 2, pp. 1398–1402. 4

[WJV*04] WILBURN B., JOSHI N., VAISH V., LEVOY M., HOROWITZ M.: High-speed videography using a dense camera array. In *Computer Vision and Pattern Recognition* (2004), vol. 2, pp. 294–301. 2

[WLD*06] WAKIN M., LASKA J., DUARTE M., BARON D., SARVOTHAM S., TAKHAR D., KELLY K., BARANIUK R.: Compressive imaging for video representation and coding. In *Proceedings of the Picture Coding Symposium* (2006). 2

[ZLN09] ZHOU C., LIN S., NAYAR S. K.: Coded Aperture Pairs for Depth from Defocus. In *IEEE International Conference on Computer Vision (ICCV)* (Oct 2009), pp. 325–332. 2