Non-Recursive Method for Motion Detection from a Compressive-Sampled Video Stream

Marco Trevisi, Ricardo Carmona-Galán, Ángel Rodríguez-Vázquez
Instituto de Microelectrónica de Sevilla (IMSE-CNM)
CSIC-Universidad de Sevilla, Spain.
E-mail: trevisi@imse-cnm.csic.es

This paper introduces a non-recursive algorithm for motion detection directly from the analysis of compressed samples. The objective of this research is to create an algorithm able to detect, in real-time, the presence of moving objects over a fixed background from a compressive-sampled greyscale video stream. Many difficulties arise using this type of algorithm because it violates the fundamental principles of compressive sensing reconstruction that lie beneath traditional recursive methods. Recursive reconstruction methods even if accurate need large amounts of time and resources because they aim to retrieve all of the information contained within a scene. Our method is based on two key considerations. The first is that the targeted information of a moving element compared to a fixed background is really small. The second is an appropriate choice of a sub-Gaussian compressive sampling strategy. Our aim is to reduce the focus of general reconstruction in order to retrieve only objects of interest. This algorithm can be used to process compressed samples derived from a video stream with a speed of 100fps. This makes possible to detect the presence of moving objects directly from compressed samples with limited resources.

Keywords— motion detection; compressive sampling; non-recursive.

I. Introduction

Compressive sensing (CS) is a signal processing technique that exploits signal compressibility to recover a signal from a small set of samples. CS core equation is [1]:

$$Y = \Phi X \tag{1}$$

where $\Phi \in \mathbb{R}^{m \times n}$ is the so-called measurement matrix, $Y \in \mathbb{R}^{m}$ are the compressed samples and $X \in \mathbb{R}^{n}$ is the signal to be recovered represented as a vector. If the original signal can be sparsely represented in some domain, like natural images [2], then it is possible to recover it from a much smaller number of samples than that indicated by Nyquist-Shannon theorem. The key requirement for achieving a successful reconstruction, i. e. approximating X given the much smaller set Y, is the sparsity of the input signal. The way in which the samples of the original signal are linearly combined to the compressed samples is encoded into the measurement matrix. The inverse problem defined by Eq. (1) is undetermined as long as $m < n$. Although underdetermined problems are considered ill-posed, as there is no univocal solution to them, compressive sensing theory can lead to a unique solution by means of convex optimization [3].

978-1-5090-0493-5/16/$31.00 ©2016 IEEE

In this paper, we are evaluating the efficiency of a lightweight non-recursive algorithm to process compressed samples of a greyscale video stream in order to detect moving objects over a fixed background. The difference of two sets of compressed samples, S_k and S_{k-1}, taken using the same measurement matrix Φ, from two consecutive frames I_k and I_{k-1}, of a given video stream, equals the set of compressed samples Y obtained by sampling X, if this is the difference of the two frames:

$$Y = S_k - S_{k-1} = \Phi(I_k - I_{k-1}) = \Phi X \tag{2}$$

thanks to the linearity of the compressive sampling process. Each element of Y maps the time evolution of a set of pixels of the original frames into the domain of compressed samples. If the background is fixed, a pixel by pixel difference of two consecutive frames will return values different from zero in those pixels that have changed. For this reason X, the difference of two consecutive frames, can be considered to be highly sparse. This sparseness will allow us to introduce a new method that will highlight the presence of moving objects without recurring to standard reconstruction algorithms.

Many methods of extracting information from compressed samples have been successfully implemented. Some of these works tackle the problem of feature extraction trying to adapt existing algorithms to the compressed domain [4]. Others focus on recursive reconstruction like algorithms that use trainable sparsifying dictionaries for object recognition [5]. Among these works, background subtraction from compressed samples using a standard reconstruction algorithm has been presented [6], but, to the best of our knowledge, there are no previous attempts to generate lightweight methods able to preprocess compressive-sampled videos in real-time.

II. Design of the Measurement Matrix

For the design of the measurement matrix we have chosen a sub-Gaussian distribution. Each element of this sub-Gaussian measurement matrix is picked at random between one and zero. This choice presents three major advantages: it is easily understandable; it is also readily implementable [7] and [8]; and its dual matrix $\overline{\Phi}$ will has the same structure as the original:

$$\bar{\Phi} = J - \Phi \tag{3}$$

being $J \in \mathbb{R}^{m \times n}$ a matrix of ones.

To obtain a single compressed sample from a sub-Gaussian measurement matrix we would simply have to add together the values of a set of randomly chosen pixels, e. g. by summing their current outputs in a common line [7]. The information about which pixels have been selected to compose the i-th compressed sample is explicitly included in the i-th row of Φ, from now on φ_i.

Instead of dumping the information contained in the discarded pixels, it is possible to generate a second compressed sample by adding these contributions as well. We call it a dual compressed sample. These dual samples can be obtained by sampling the same frames, I_k and I_{k-1}, using the dual matrix $\bar{\Phi}$. The subtraction of two sets of dual compressed samples $\bar{S}_k$ and $\bar{S}_{k-1}$ will give us $\bar{Y}$ the dual difference of compressed samples:

$$\bar{Y} = \bar{S}_k - \bar{S}_{k-1} \tag{4}$$

It will not be possible to use this dual difference, $\bar{Y}$, along with the original, Y, to double the amount of compressed samples for standard reconstruction purposes. The joint measurement matrix, $[\Phi^T \quad \bar{\Phi}^T]^T$, would generate a divergent solution in a recursive method. However, this description will be useful later on, in the non-recursive motion detection.

III. Motion Detection Algorithm

Let us consider the structure of the data generated by compressively sampling two consecutive frames and then subtracting one from the other Eq. (2).Each element of Y, which, from now on, we will call y_i, is the difference of two compressed samples generated by the same row of Φ:

$$y_i = \varphi_i X = \varphi_i (I_k - I_{k-1}) \tag{5}$$

If the pixels that contribute to y_i do not contain any changes, i. e. no object appears or disappears to or from them, then the value of y_i will be zero because that subset of I_k equals the corresponding subset of I_{k-1}. Likewise, given the presence of a moving object in that subset, the contribution to y_i of the pixels belonging to that subset, differs from zero. It means that the pixels selected by φ_i are experiencing changes between frames. The higher the value of y_i the higher is the amount of pixels selected by φ_i that contains differences between the two consecutive frames.

Based on these elements it is possible to create m contribution vectors, $c_i^+ \in \mathbb{R}^n \quad \forall i \in \{1,2,\dots,m\}$, by multiplying each row of the measurement matrix, φ_i, by the scalar y_i which is the value of the difference of compressed samples generated by said row:

$$c_i^+ = \varphi_i y_i \tag{6}$$

Notice that each contribution c_i^+ vector is a sparse vector that has the same size as the original frame difference X. All non-zero elements, which correspond to the positions of pixels selected by row φ_i, will have a value of y_i.

Thanks to the selection of a sub-Gaussian measurement matrix, each $\varphi_i$will have, by construction, a dual in $\bar{\Phi}$:

$$\bar{\varphi}_i = j_i - \varphi_i \tag{7}$$

being j_i a row of matrix J. If an object appears in a spot represented by a pixel chosen by φ_i as part of a given sample, that object disappears from the pixels included in $\bar{\varphi}_i$. Which means that the dual difference $\bar{y}_i$ will also experience a fluctuation. What is more is that φ_i together with its dual $\bar{\varphi}_i$ contain the contributions of all the pixels of the frame, but no contribution belonging to φ_i will be in $\bar{\varphi}_i$, nor vice versa.

Analogously, it is possible to define a dual contribution vector $c_i^- \in \mathbb{R}^n \quad \forall i \in \{1,2,\dots,m\}$ from the dual differences:

$$c_i^- = \bar{\varphi}_i \bar{y}_i \tag{8}$$

Adding together these two vectors it is possible to obtain $c_i \in \mathbb{R}^n \quad \forall i \in \{1,2,\dots,m\}$, a contribution vector of all pixels. These vectors c_i will no longer be sparse and the value of their elements will either be y_i or $\bar{y}_i$. The pixels selected by φ_i and $\bar{\varphi}_i$ have been chosen randomly. The amount of pixels selected by them is also random. This means that each pair of φ_i and $\bar{\varphi}_i$ will generate a different contribution vector c_i.

It is worth nothing that if an object moves within the subset of the image defined by φ_i, even if y_i differs from zero, its dual sample $\bar{y}_i$ will still be zero. For that reason only non-sparse contribution vectors c_i contain information on the motion of an object. A global contribution vector $C \in \mathbb{R}^n$ can be generated by superposing the contribution of all c_i's where both y_i and $\bar{y}_i$ are different from zero:

$$C = \sum_{\substack{\forall i \\ y_i \neq 0 \\ \bar{y}_i \neq 0}} c_i = \sum_{\substack{\forall i \\ y_i \neq 0 \\ \bar{y}_i \neq 0}} (\varphi_i y_i + \bar{\varphi}_i \bar{y}_i) \tag{9}$$

The elements of the global contribution vector C are linear combinations of compressed samples and every other their dual. Let us assume that they are very likely to be different form each other. Each pixel contribution will be accounted for the same exact number of time as any other. This means that each element of C is directly comparable to the others. Notice that this is true only because of φ_i and $\bar{\varphi}_i$ together address all the pixels of the frame. Thus, the higher the value associated to a given element of C the greater the contribution of the corresponding pixel. In a sense, the maxima and minima in C reflect those pixels where change happened, what means those pixels which contributed more notably to the compressed samples which addressed them. For that reason it is possible to associate those maxima and minima with the positions that the moving objects have occupied in the two consecutive frames.

This algorithm not only returns information on the presence of an object moving on a fixed background in two consecutive frames, it also gives information on the direction of that motion, i. e. where the object was and is likely to be.

At last it is important to notice that none of the steps taken for the generation of these contributions is recursive and, due to the randomness of Φ, its accuracy is proportional to the amount of compressed samples computed.

IV. Experimental Evaluation

To validate the presented method several simulations have been carried out. Some of the results, as well as the MATLAB code used to generate them, can be downloaded from: http://www.imse-cnm.csic.es/mondego/prime2016/

The performance of this method was studied creating various synthetic videos of 50 frames of 64×64 pixels each. Each video has a black background and one or more moving objects represented by 3×3-pixel white squares. Beside the number of objects another variable that we considered was the amount of compressed samples extracted. As a 64×64-pixel frame has a total amount of pixel of 4096, we took sets of samples reaching a compression ratio that ranged from 1/2, with 2048 compressed samples up to 1/32 with 128 compressed samples, halving the total amount of samples taken at every step.

To compare the results obtained with our method we compared each maxima and minima of the global contribution vector C with the difference of the original frames from which the compressed samples were derived. We also used the NESTA algorithm [9] on the same compressed samples sets to compare the performance of our method to one of the most effective convex optimization reconstruction algorithms currently presented in literature.

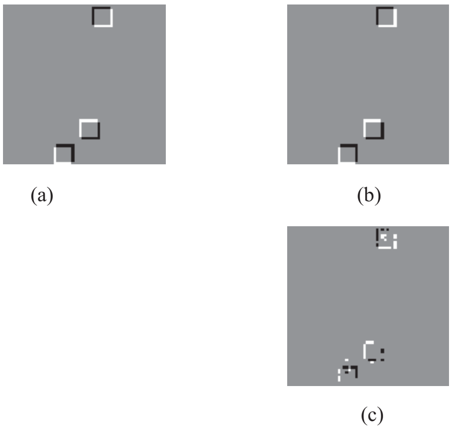

Fig. 1. (a) Difference of two original 64 × 64-pixel frames; (b) Filtered difference of two NESTA reconstructed 64 × 64-pixel frames; (c) Maxima and Minima of the contributions extracted from two sets of compressed samples following our method.

Fig.1 represents one example of our analysis. In this particular setup we used three moving objects and a set of 1024 compressed samples i. e. 1/4 compression. Fig.1(a) represents the difference of the original frames from which 1024 compressed samples and 1024 dual compressed samples have been simultaneously extracted. Fig.1(b) represents the reconstruction of the difference of said frames by applying a NESTA convex optimization algorithm over the set of compressed samples differences Y. This difference has been further filtered to remove noise. This has been done to easily compare it to the maxima and minima of the weight vector. It is possible to see that it resembles closely the difference of the two original frames shown in Fig.1(a). Lastly Fig.1(c) represents the location of the maxima (white) and minima (black) of the contribution vector produced by our proposed algorithm. These contributions have been scaled to fit a greyscale representation to easily compare them graphically with the original difference as well.

Two different parameters where considered to establish the reliability of this method. The first parameter is the pixel by pixel root mean square error (RMSE) of the scaled contribution vector defined in Eq. (9) using the original frame differences as ground truth Fig.2. The values that the RMSE of our method took were then compared to the RMSE through NESTA reconstruction using the same ground truth and the same set up. We recorded these values while varying the amount of objects moving in the scene and the amount of compressed samples taken Fig.3.

The second parameter considered to establish the reliability of this method is the time of reconstruction. We compared the time that the NESTA algorithm took to recover a difference Y and the time our method took to locate the maxima and minima of the global contribution vector C.

Comparing Fig.2 and Fig.3 it is possible to see that traditional reconstruction derives better results over our proposed methodology in all cases of study. This was to be expected because relying on convex optimization leads better results than relying on a method whose strength is based only on the sheer amount of samples taken.

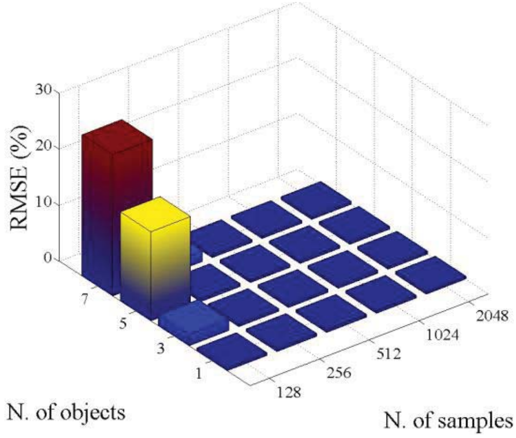

Fig. 2. RMSE of NESTA reconstruction (% of Full Signal Range).

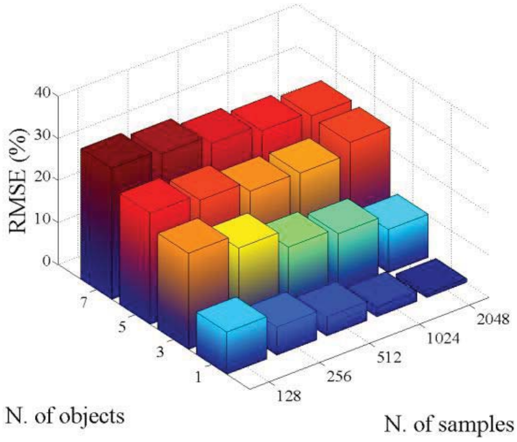

Fig. 3. RMSE of our method (% of Full Signal Range).

Even if that is the case it is worth noting that NESTA reconstruction fails in returning acceptable results in extreme cases i. e. high number of objects and low number compressed samples.

We have run the whole procedure on an Intel core i7-3740QM running at 2.7GHz having 24GB of RAM with an SSD. Comparing Fig. 4 and Fig.5 it is possible to see that the longest time our method took to estimate the position of a moving object was 2 to 3 orders of magnitude lower than the time it took for NESTA to deliver its results.

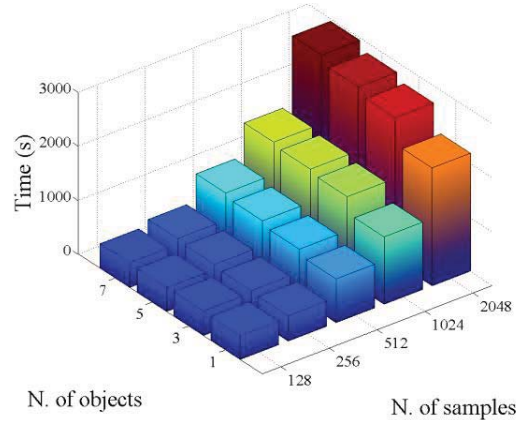

Fig. 4. Time needed for NESTA reconstruction (seconds).

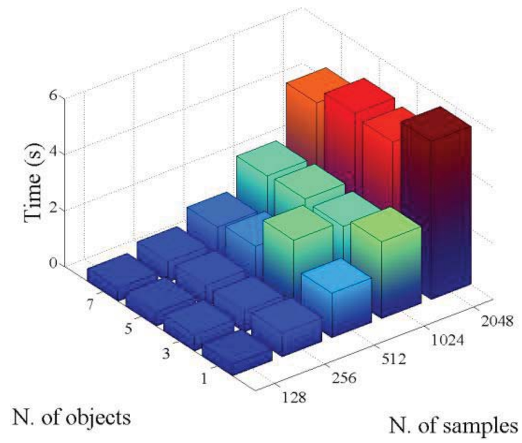

Fig. 5. Time needed for maxima and minima extraction (seconds).

Considering that the videos we were reconstructing had a total of 50 frames, in most cases it would have been possible to analyze said videos while streaming the results in real-time.

V. Conclusions

We have derived a new method to extract information from compressed samples without resorting to recursive reconstruction. This has resulted in a lightweight fast detection algorithm capable of analyzing a stream of compressed samples obtained by sampling frame by frame a greyscale video in order to detect the presence of moving objects over a fixed background.

We have compared the performance of our algorithm with a highly efficient reconstruction algorithm, NESTA, to understand which benefits and limitations we are facing while trying to handle compressive-sampled information without recurring to standard techniques. While trying to target specific information within the compressed samples thus not following a conventional reconstruction technique may deliver worse reconstruction errors it is also true that it can benefit from faster processing times opening the possibility to new applications of CS.

Our future works will focus on the application of these techniques for the detection of fast moving objects. To do that we will need to improve the frame rate. It will also be beneficial to reduce the root mean square error. Our final goal is to apply this algorithm to the reconstruction of 3D trajectories using compressed samples from multiple views of the scene

Acknowledgments

This work has been funded by the Spanish Government through projects TEC2015-66878-C3-1-R MINECO (European Region Development Fund, ERDF/FEDER), IPT-2011-1625-430000 MINECO and IPC-20111009 CDTI (ERDF/FEDER), by Junta de Andalucía through project TIC 2338-2013 CEICE and by the Office of Naval Research (USA) through grant N000141410355.

References

[1] E. Candès. "Compressive sampling". *Int. Congress of Mathematics*, pp. 1433-1452. Madrid, Spain, 2006.

[2] J. Romberg. "Imaging via Compressive Sampling". *IEEE signal Processing Magazine*, Vol. 25, No. 2, pp. 15-20. March, 2008.

[3] R. G. Baraniuk, V. Cevher, M. B. Wakin. "Low-Dimensional Models for Dimensionality Reduction and Signal Recovery: A Geometric Perspective". *Proc. of the IEEE*, Vol. 98, No. 6, pp. 959-971, Jun 2010.

[4] Davenport, M. Duarte, M. Wakin, J. Laskar, D. Takhar, K. Kelly, R. Baraniuk. "The Smashed Filter for Compressive Classification and Target Recognition". *Proc. SPIE Computational Imaging V*, San Jose, California, Jan 2007.

[5] P. Nagesh and B. Li. "A Compressive Sensing Approach for Expression-Invariant Face Recognition". *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1518-1525, Jun. 2009.

[6] H. Jiang, W. Deng, and Z. Shen, "Surveillance video processing using compressive sensing," *Inverse Problems and Imaging*, vol. 6, no. 2, pp. 201–214, 2012.

[7] V. Majidzadeh, L. Jacques, A. Schmid, P. Vandergheynst and Y. Leblebici. "A (256x256) Pixel 76.7mW CMOS Imager/Compressor Based on Real-Time In-Pixel Compressive Sensing". Circuits and Systems (ISCAS), *Proc. IEEE International Symposium*, pp. 2956-2959, May 2010.

[8] Y. Oike, A. El Gamal. "CMOS Image Sensor With Per-Column ΣΔ ADC and Programmable Compressed Sensing". *IEEE Journal of Solid-State Circuits*, Vol. 48, No. 1, pp. 318-328, Jan. 2013

[9] S. Becker, J. Bobin, E. Candès. "NESTA: a Fast and Accurate First-Order Method for Sparse Recovery". *SIAM J. Imaging Sci.*, Vol. 4, No. 1, pp. 1–39. Jan 2011.