Evaluation of Compressive Sensing encoding on AR Drone

Karan Shetti, Asha Vijayakumar
Airbus Group Innovations, Singapore
Email: karan-rajendra.shetti@airbus.com

***Abstract*—Micro-flying robots such as quadcopters or drones are being used extensively in many civilian applications. They are generally integrated with different sensors and are designed to perform tasks both autonomously as well as with manual feedback. These drones typically transmit data periodically either to a base station or to each other if deployed in a swarm. As the number of sensors on-board increases, communication bandwidth becomes a critical aspect for these drones. While there are multiple approaches to improve bandwidth, they typically involve modification of the communication infrastructure. In this paper, we propose a unique method to reduce the data from a typical sensor like an on-board camera using Compressive Sensing (CS) technique. Our method does not require any changes to the communication infrastructure used (WLAN 802.11a) and can be possibly extended to other communication links. We have implemented the CS based encoder on-board the A.R. Drone and conducted various experiments to measure execution time of the processing and quality of the data transmitted to validate our approach.**

I. Introduction

Design and development of micro-flying robots, also termed drones or UAVs, is an active topic of research. Different applications of such autonomous control for detection and tracking of objects [1], as stations for creating multi-hop wireless networks [2] are some of the topics being investigated. Some commercial applications such as package delivery and infrastructure inspection are also being developed [3]. One of the crucial aspect for the success of these applications on drones would be the communication link. Drone swarms as well as single autonomous drones need to continuously send and receive data either to each other or communicate with a base station.

Most drones typically use commercial off-the-shelf wireless devices such as WLAN operating in 802.11a mode for communication through an ad-hoc network. This method has gained prominence as it is economical and well supported in the embedded systems domain. However, such drone networks are challenged by various factors which traditional networking approaches have not been designed for. While they perform well in an indoor environment, they are affected by changes in signal propagation, link quality, PHY rate selection and antenna orientation in 3D space [4] in the outdoor environment. Asadpour et al. [4] also show that automatic rate adaptation of standard 802.11 chip-sets cannot cope with high mobility of drones and needs to be adapted for the same.

To tackle these problems, research to improve antenna and protocol designs is being undertaken to increase throughput and reduce delays. Sharawi et al. [5] propose using 3D printed array of antennas to improve throughput. Their approach does not change the aerodynamics of the UAV as the antennas are embedded on the wings. While this is a good approach, it requires additional hardware and may not be suitable for use in cheaper quadcopter based drones.

In this paper, we propose an alternate solution that supports effective transmission of the data through the available bandwidth. The focus is on reducing the data that needs to be transmitted from the drone using compressive sensing techniques. Such an implementation allows the use of existing infrastructure and protocols without comprising on the quality of the data. We illustrate how CS based encoding of images can be implemented on an embedded platform like the drone's on-board computer (OBC) by sampling them below the Nyquist rate.

A. Compressive Sensing

The theory of CS enables the acquisition of a few random measurements of the signal thereby acquiring the signal in the compressed form combining the stages of signal acquisition and compression[6]. A prerequisite for the successful reconstruction of signals from few non-adaptive random measurements is that the signal is sparse in some basis ψ [6], [7],

$$\mathbf{x} = \psi \mathbf{s} \tag{1}$$

where ψ is a $N \times N$ basis matrix and $\mathbf{s}$ is a sparse vector. Most of the real world signals are known to be sparse as in (1). A random $M \times N$ projection matrix, ϕ, is used to capture the input vector measurements (y), where $\mathbf{y} = \phi\psi\mathbf{s}$. The number of measurements M to be captured depends on the sparsity K of the signal. A CS based systems takes the load off the encoder by passing it on to the decoder which is ideal for applications on drones. The two important design tools for a CS system are the sensing matrices ϕ to randomly sample the measurement and the optimization algorithm to reconstruct the sparse signals [6], [7]. The sensing matrices are basically projection matrices designed to conform to two important properties. Firstly, the matrix ϕ should preserve the metric when the vectors are projected from a higher dimensional space to a lower dimensional space. This property is referred to as restricted isometry property (RIP). The second property is that the sensing matrix should be incoherent to the basis in which the signal is sparse. The sparse signal can be recovered using convex optimization like, $\hat{\mathbf{s}} = argmin\|\mathbf{s}\|_1$ such that $\chi\mathbf{s} = \mathbf{y}$ where $\chi = \phi\psi$.

II. Implementation

A. System Design

The system design consists of an encoder and a decoder. The on-board encoder receives data from the HD video camera and converts the data to the compressed sensed domain using a structured random matrix[8]. The data captured is transmitted as UDP packets to the ground station where the decoder reconstructs the data using an optimization algorithm. There are two major challenges for the design of a CS based system for real time applications (1) Design and implementation of a sensing matrix and (2) A fast reconstruction algorithm. In this work we consider a sparse reconstruction method by separable approximation (SpaRSA) [9] for reconstruction of compressed sensed images obtained from the drones. In [9] the authors have shown that SpaRSA is much faster and is suitable for solving large scale optimization problems. The compressed sensed data on drones is transmitted through the wireless network in two different ways - frame wise transmission and selective frame transmission as explained in the subsequent sections.

1) Frame wise transmission: The Parrot AR.Drone 2.0 is based on a classic quadrotor design. It consists of a Parrot P6 processor (32bits ARM9-core, running at 468 MHz) and a front camera capable of HD recording and VGA resolution transmission (640×480) at a frame-rate of 15 frames per second [10]. A Linux based real-time operating system with Busybox is used and all processing is done on-board by the propriety software installed. To maximize the quality of the view, the front camera uses a 93 degrees wide-angle diagonal lens. The AR Drone camera produces a raw HD image of size 1280×720, these frames are then down-sampled to 256×256 image to fit the data into a single UDP packet for transmission to the ground station. The total number of samples N in this case is 65536 and only M samples of each frame are transmitted using the UDP protocol. Each of the transmitted frames are reconstructed at the decoder using SpaRSA.

2) Selective frame transmission: In many field missions the changes between successive frames, in the captured video, is not always significant. In [11], the authors have shown that it is possible to do signal detection and estimation from the few random measurements that are captured without having to reconstruct the signal. Such methods facilitate quick on board decisions. In this work we adopt a simple method to mark a sudden change in the successive frames in the compressed sensed domain. A difference of each incoming frame with the previous frame is taken. The variance of the compressed sensed differential frame is measured against an empirically derived threshold. If the variance of the compressed sensed differential frame is above the marked threshold then the compressed sensed samples of that corresponding frame is transmitted. A selective frame transmission not only helps in reducing the data as shown in Section III-C but also helps us to evaluate the processing capability on board the Unmanned Aerial vehicle (UAV).

B. Embedded Implementation of the sensing matrix

The primary challenge of the design of a CS based system is to design a sensing matrix that supports fast computation, optimal performance and is also hardware friendly. In [8] authors have shown that a structured random matrix (SRM) satisfies the required characteristics of a sensing matrix making it a preferred choice for real time compressive sensing applications. In the SRM implemented in this paper we pre-randomizes the input by flipping the sample signs and then transform it using a Fast Fourier Transform (FFT) before picking up the M random measurements. The selection vector to pick up the random measurements is stored as a lookup table in files. The structured random matrix was implemented on an embedded platform. The Cooley-Tukey FFT algorithm was used in developing the hand written code for FFT with twiddle factors stored as look-up table to improve the execution time.

C. Implementation on AR Drone 2.0

The Parrot AR Drone provides only the basic accessibility to the drone hardware and cameras. While the software running on-board is not accessible, Parrot provides an SDK to communicate with drone, however this does not allow any processing on-board. Since our objective is to illustrate the on-board embedded implementation of a CS encoder and evaluate the performance of processing in the compressed sensed domain, we completely halt the program running the drone firmware to avoid overheads during our experiments. While this disabled the flight control and navigation, it allowed us access to all the devices as any Linux embedded system. The front camera was setup to tap the raw HD data using the Video4Linux library thereby resulting in more flexibility to implement the encoder onboard the drone hardware. In order to achieve a higher frame rate, we ported the FFTW library [12], [13] on the ARM processor. FFTW is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data. This library had to be recompiled to work in the ARM environment and by enabling the NEON instructions, which are similar to Intel's SSE, there was an significant improvement in the execution time as shown in Section III.

III. Results

A. Experimental Setup

The implementation of the encoder was done on the ARM processors as mentioned in Section II. Various experiments were conducted which measured the execution time and the signal to noise ratio (SNR) for different sampling ratios M/N. The decoder was implemented on a Intel Core 2 Duo PC as a Matlab application. The encoder used the SpaRSA library to reconstruct the transmitted image. The plots contains the average value of the results tabulated over 10 iterations for both execution time and SNR experiments.

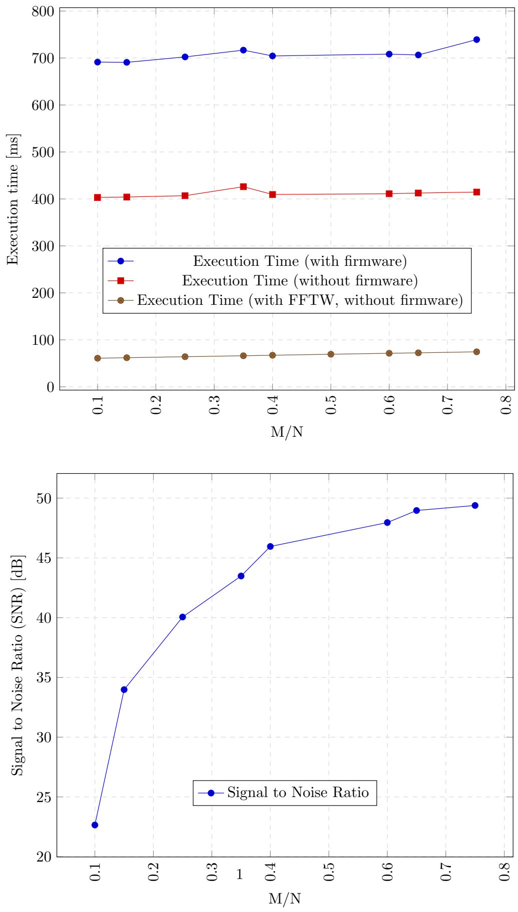


Fig. 1. Execution time on AR Drone and Reconstruction SNR

B. *Frame wise transmission*

Figure 1 shows the execution time of frame wise transmission implemented under three different cases namely, with propriety firmware, without firmware and finally using the FFTW library. Case 1 and 2 simulate the performance of the application on-board a fully operational drone and a generic ARM processor respectively. The drone firmware uses the CPU quite extensively and shows a significant overhead ($\approx 300ms$) across all sampling ratios. On average without the firmware overhead, we achieve $\approx 400ms$ execution time using the hand written Cooley-Tukey algorithm for FFT. However, this is still not optimal as we can only achieve 2 frames per sec using this method. As shown in Figure 1 the FFTW library provides a significant improvement over the former method as it also uses the Neon instructions to further accelerate the program. It can also be observed from Fig. 1 that the overall execution time does not change significantly for different sampling ratios allowing the choice of the sampling ratio to be made based on the SNR requirements.

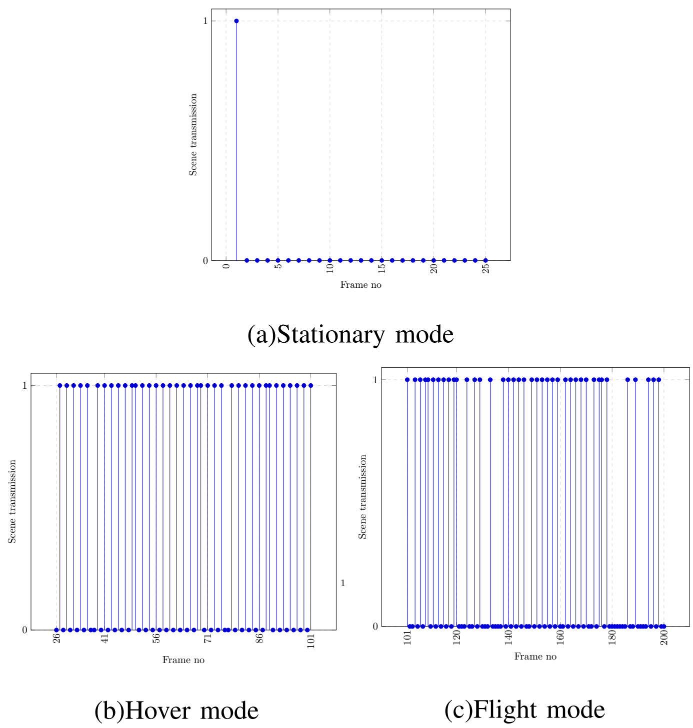
The SNR is defined as $20\,log_{10}(\|\mathbf{x}\|_2/\|\mathbf{x}-\widehat{\mathbf{x}}\|_2)$ where the original image, $\mathbf{x}$ is from the drone and $\widehat{\mathbf{x}}$ is the reconstructed image. When the image was reconstructed using only 10% of the data, we can observe that the SNR was poor, whereas the SNR improves with an increase in the sampling ratio reaching a peak of 49.5 dB when 75% of the samples were used. The variation of the SNR over different sampling ratios is shown in Figure 1.

C. *Selective frame transmission*

Three scenarios were chosen which were used for testing the effectiveness of selective frame transmission. These are namely

1) Frame [0-25]: Stationary drone with no movement
2) Frame [26-100]: Slight movement (hover mode)
3) Frame [101-200]: Rapid movement (flight mode)

As shown in Fig. 2, the binary result of scene(sampling ratio = 0.5) transmission under different modes is plotted against the different frames used. We define keyframe as the frame where significant changes are observed. The difference of each frame with its previous frame is measured and the variance of the compressed sensed differential frame is measured against a threshold. In the first section, we can observe that no new information was detected and hence only one keyframe was transmitted. The number of keyframes increase as there is more movement in the hover mode and flight mode as the changes are more prominent. Overall when this method is used, a total of 79 frames ($1/25$ is stationary, $38/75$ in hover and $40/100$ in flight mode) out of 200 are sent, reducing the data by 61%.

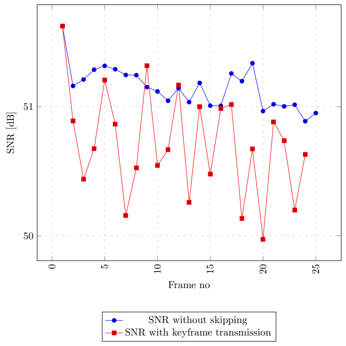


(a)Stationary mode

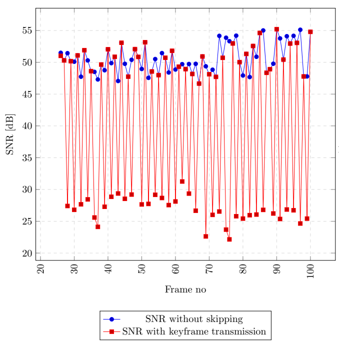
(b)Hover mode (c)Flight mode

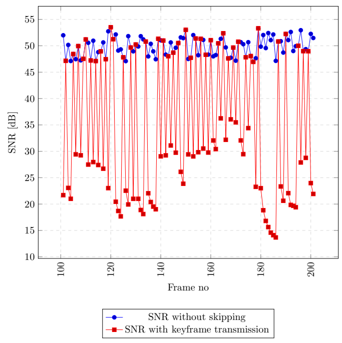
Fig. 2. Image transmission for different modes

Fig. 3 shows the comparison of SNR analysis of image frames with and without using keyframes. Images that were not transmitted were compared with their corresponding original images to judge the accuracy of this approach. In this graph, the blue stem shows the variation of SNR for the case when the skipping of frames was not done and the red stem shows the case of variation of SNR when only the keyframes are used. While there is a drop in the SNR as expected, the absolute value of the SNR is still quite high. However, in the case of rapid motion, the selective transmission shows poor performance especially between the frame numbers $180-200$. From Fig.3 it can be observed that though there is rapid motion frames are not updated frequently. A possible solution could be to derive more robust thresholding scheme that determines the frame transmission.



(a)Stationary mode





(b)Hover mode (c)Flight mode

Fig. 3. SNR comparison for selective transmission

D. CPU utilization and power consumption

As mentioned before, the firmware onboard the AR Drone is processor heavy and on average consumes $\approx 47\%$ of the CPU when the drone is not in flight. In terms of power it consumes $0.26A$ which corresponds to 3.13 W (operating voltage of $12V$). At 10 fps, our proposed method (without firmware) utilizes $\approx 44\%$ of the CPU for $M/N = 0.5$. The power consumed in this case is $2.88W$ $(0.24A)$. We attribute this to efficient use of NEON instructions applied by the FFTW library that provide superior performance without increasing the clock frequency.

IV. Conclusion

We have shown that it is possible to effectively reduce the image data transmitted using compressive sensing while still using current communication infrastructure. The random sensing of the image data was implemented on-board an embedded processor like ARM by carefully partitioning while reconstructing the data at the decoder(ground station). We have also shown the feasibility of using compressed sensed data statistics on-board UAV to decide on a drastic scene change.

While 802.11n used here as an example, does provide enough bandwidth to transmit images even without CS based encoding we believe our method is relevant as it can easily be adapted to other communication protocols that provide lower bandwidth as well as sensors that produce more data. We further look to refine the method of selective frame transmission as helps to track a scene change on board thereby facilitating a significant reduction in the data that needs to be transmitted.

References

[1] I Sa and P Corke, “Vertical infrastructure inspection using a quadcopter and shared autonomy control,” *Field and Service Robotics*, pp. 219–232, 2014.

[2] D. Henkel and T. Brown, “On controlled node mobility in delay-tolerant networks of unmanned aerial vehicles,” 2006.

[3] Md R Haque, M Muhammad, D Swarnaker, and M Arifuzzaman, “Autonomous quadcopter for product home delivery,” in *2014 International Conference on Electrical Engineering and Information & Communication Technology*. IEEE, 2014, pp. 1–5.

[4] D. Giustiniano M. Asadpour and K.A. Hummel, “From ground to aerial communication: Dissecting wlan 802.11n for the drones,” in *Proceedings of the 8th ACM international workshop on Wireless network testbeds*. ACM, 2013, pp. 25–32.

[5] M.S. Sharawi, D.N. Aloi, and O.A. Rawashdeh, “Design and implementation of embedded printed antenna arrays in small uav wing structures,” *Antennas and Propagation, IEEE Transactions on*, vol. 58, pp. 2531–2538, Aug. 2010.

[6] D. Donoho, “Compressed sensing,” *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

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

[8] Thong T. Do, Lu Gan, Nam H. Nguyen, and Trac Tran, “Fast and efficient compressive sensing using structurally random matrices,” *IEEE Trans. on Signal Prcoessing*, vol. 60, no. 2, pp. 139–154, Jan. 2012.

[9] S. J. Wright, R. Nowak, and M. A. T. Figueiredo, “Sparse reconstruction by separable approximation,” *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, July 2009.

[10] PJ Bristeau, F. Callou, D. Vissiere, and N. Petit, “The navigation and control technology inside the ar. drone micro uav,” in *18th IFAC World Congress*, 2011, vol. 18, pp. 1477–1484.

[11] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, “Signal processing with compressive measurements,” *IEEE J. Sel. Topics on Signal Prcoessing*, vol. 4, no. 2, pp. 445–460, April 2010.

[12] Matteo F. and Steven G. J., “The design and implementation of FFTW3,” *Proceedings of the IEEE*, vol. 93, pp. 216–231, 2005, Special issue on “Program Generation, Optimization, and Platform Adaptation”.

[13] Vesperix Corporation, “Building fftw on arm,” 2011.