SAND2019-15046R

TFA Performance Improvement

Jamie J. Wingo

December 9, 2019

Acknowledgment

The author would like to thank Jaclynn Stubbs and Gabriel Birch for their advice and support. The research in this presentation was conducted under contract with the U.S Department of Homeland Security (DHS) Science and Technology Directorate (S&T), contract #HSHQABCD-00-X-00000. The opinions contained herein are those of the contractors and do not necessarily reflect those of DHS S&T.
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

1 Abstract

The objective of this project was to increase the rate at which video data is processed using temporal frequency analysis. A common solution to increasing the speed of data processing is to increase the computing power of the system however size, weight and power (SWAP) constraints require computing power to be limited. This project focused on increasing the processing speed by reducing the expense of computing the Fourier Transform (FT).

2 Introduction

Sandia National Labs (SNL) has been developing a method to detect and assess unmanned aerial systems UAS that uses temporal frequency analysis (TFA) [1]. TFA examines the changes in pixels over time and has shown promise in successfully detecting and assessing UASs. The image in figure 1 has been processed and classified using TFA. One area in which TFA can be improved is the rate at which frames can be processed. The data must be processed in real time to realize the necessary overall system performance. In this project, we investigated if a Sparse Fourier Transform method, a Short Time Fourier

Transform (STFT) method and if using a larger temporal window size resulted in a speed increase without lowering the accuracy of the program.

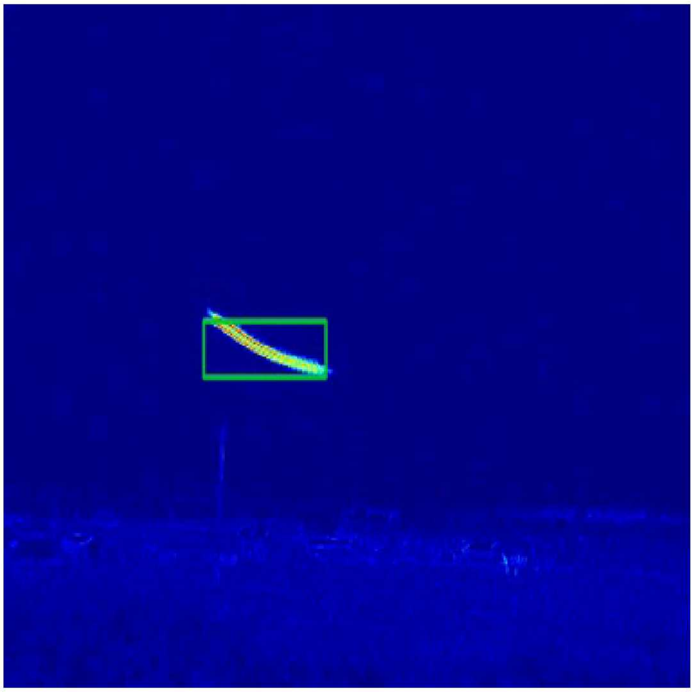

Figure 1: Drone in Flight after TFA Processing [1]

3 Experimental Methods

3.1 Data and Processing

Real video was used as a background in which to insert a synthetic drone in flight to enable accuracy testing. Labeling the real video was outside the scope of this project however the results of this study will be verified once the data set has been labeled. The background video has motion in the background and added noise to imitate real video as closely as possible. The drone flight starts at a random point within the video and continues for either 300 or 320 frames, dependent on the temporal window size being used to process that video. Twenty videos were created in each of four square dimensions: 512, 256, 128 and 64 pixels. These videos were processed using TFA and the run time was recorded. The accuracy of the classification was computed by comparing the coordinates of the bounding boxes returned by the algorithm with the coordinates used to insert the drone into each video. Since the image stack shows change in pixels over time, the drone might be found multiple times within a stacked frame or just once and still be considered a correct classification. When TFA returned a bounding box that should not have contained a drone or did not find the drone when it was present, this was counted as an incorrect classification. The number of correct classifications out of the approximately 300 known instances is shown for each image size in figure 2. The blue bars correspond to a 32 frame temporal window processed with FFT, the purple to a 30 frame temporal window processed with FFT and the green bars to a 30 frame temporal window processed with STFT. The number of frames per second for each size is shown in figure 3 with the same color correspondence.

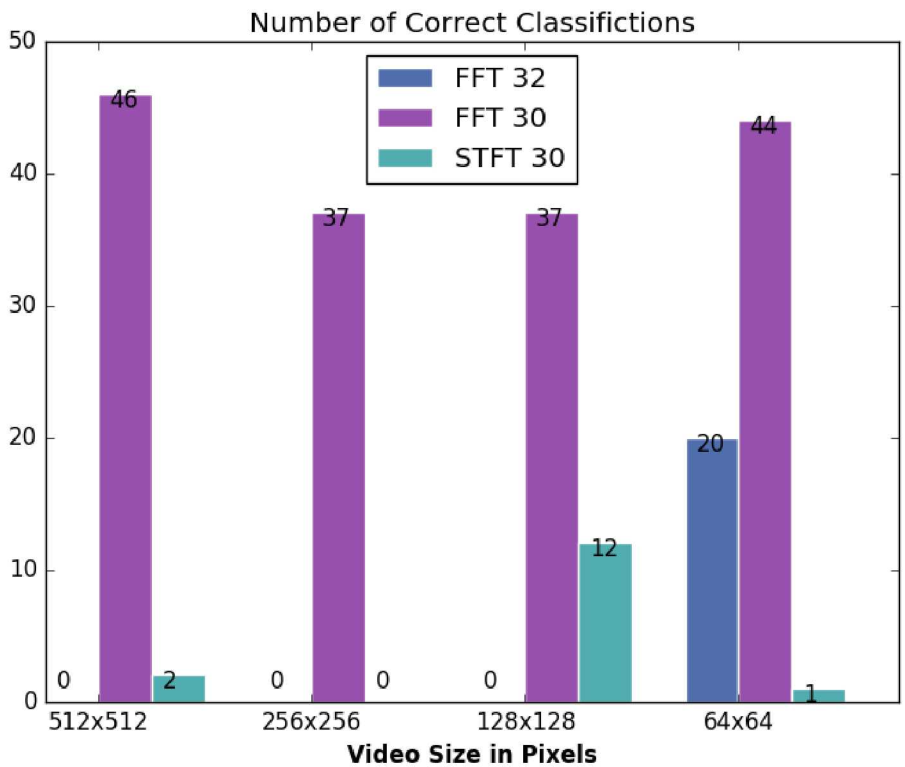

Figure 2: Correct Classifications by Video Size

4 The Current Implementation

The program accepts streaming video data with no restrictions of the video dimension. Frames are gathered in stacks of 30 and the Fast Fourier Transform is performed on each pixel fluctuation vector [1]. Background subtraction and filtering is employed to further isolate the drone. Each image stack is then classified using a retrained version of You Only Look Once (YOLO)[2]. The

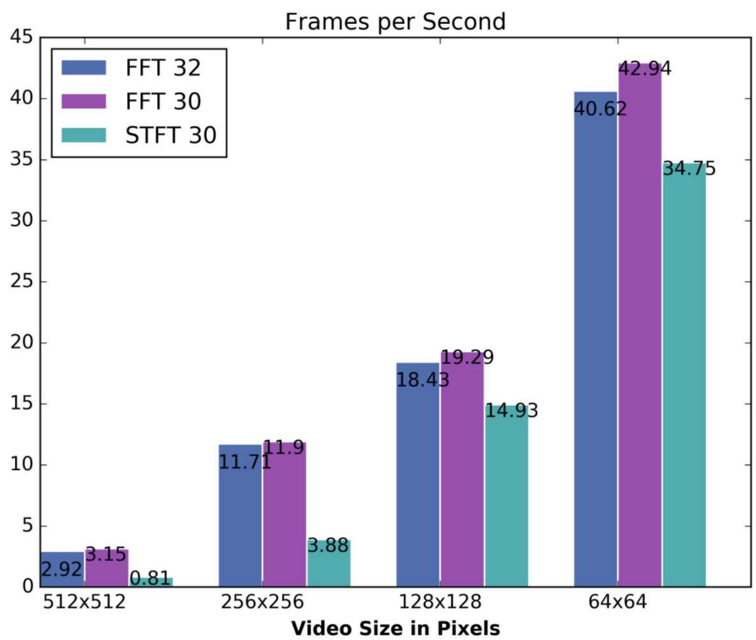

Figure 3: Frames per Second by Video size

frame rate for each size tested is shown in table 1. The original TFA correctly found the drone more times that any other version of TFA as shown in figure 2.

Image Size	Frames per Second
512X512	3.15
256X256	11.9
128X128	19.29
64X64	42.94

Table 1: FFT Frames per Second for 30 Frame Stack

5 Alternative Windowing

One simple way to increase the frames per second processed by TFA is to increase the size of the frame stack. If more frames are processed each iteration, then there will be less overall iterations however if the stack is too large information will not be presented to the user in real time. The benchmarks for FFTW, the FFT library used by TFA for computation, show that it is faster at computing the FFT of power of two sizes[3]. When the temporal window size was set to 32 rather than 30, a slight decrease in frames per second was seen for each video size. The frame rate for each of the sizes is shown in table 2. The accuracy of this method is lower than the accuracy of the original TFA algorithm as can be seen in figure 2. More analysis is needed to determine why the accuracy dropped when the temporal window size changed.

Image Size	Frames per Second
512X512	2.92
256X256	11.71
128X128	18.43
64X64	40.62

Table 2: FFT Frames per Second for 32 Frame Stack

6 Alternative Fourier Transforms

Reducing the computational expense of the FT step in TFA is another way to decrease the run time for each iteration. Two techniques for performing the FT were tested: the Sparse Fourier Transform (SFT) and the Short Time Fourier Transform (STFT).

6.0.1 Sparse Fourier Transform

The SFT is a method of performing the FT on data that will be sparse after the transform has been performed. When many of the Fourier Coefficients are zero, it is possible to compute the FT in sub-linear time with respect to N, where N is the number of elements[4]. The claim is made the the SFT method is faster the FFT for all transforms with sizes in powers of two. The results of testing the ST on 32 frame data stacks did not support that claim, with the frames per second being lower than images processed using the FFT. None of the data for this experiment was recoverable when a computer malfunctioned and had to be rebuilt. Due to time constraints and the unfavorable frame per second rates, the testing of the SFT has not been redone. More recent literature on SFT methods for distributed data warrant a second look at this prospective FT method[5].

6.0.2 Short Time Fourier Transform

The STFT is a signal processing method that works well for finding changes over time. It has been used to analyze video to track honey bees [6] and measure heart rates [7]. The STFT method of computing the FT was tested on both 30 frame stacks and 32 frame stacks, however the timing and accuracy data for the 32 frame stacks was corrupted and needs to be re-created. The frame per second rate for the 30 frame stacks is shown in table 3. The processing rate for the STFT is much lower than for the FFT. The accuracy, as shown in figure

3 is also lower than when using FFT. The lower accuracy could be due to the filtering of the image after the STFT is performed. The FFT has a filter that was evolved using a genetic algorithm while the STFT implementation keeps the mid-range frequencies.

Image Size	Frames per Second
512X512	0.81
256X256	3.88
128X256	14.93
64X64	34.75

Table 3: STFT Frames per Second for 30 Frame Stack

7 Conclusion

The alternative Fourier Transform methods were both slower and less accurate than when using FFT so they are not viable alternatives to using FFT. The change in the temporal window size did not result in a better frame rate as expected and the loss of accuracy needs to be addressed if it is to be implemented in the final product. To determine the cause, the optimal filter for the 32 temporal window should be developed and the tests re-run. The testing also needs to be done with real video that has been labeled rather than only on the simulated video. This project did not succeed in improving TFA performance however further work will be done in exploring the options for improvement.

References

[1] B. L. Woo, G. C. Birch, J. J. Stubbs, and C. Kouhestani, "Unmanned aerial system detection and assessment through temporal frequency analysis," in *2017 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–5, IEEE, 2017.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[3] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[4] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Nearly optimal sparse fourier transform," *arXiv preprint arXiv:1201.2501*, 2012.

[5] A. C. Gilbert, P. Indyk, M. Iwen, and L. Schmidt, "Recent developments in the sparse fourier transform: A compressed fourier transform for big data," *IEEE Signal Processing Magazine*, vol. 31, pp. 91–100, Sep. 2014.

[6] K. Shimasaki, M. Jiang, T. Takaki, I. Ishii, and K. Yamamoto, "Hfr-video-based honeybee activity sensing using pixel-level short-time fourier transform," in *2018 IEEE SENSORS*, pp. 1–4, Oct 2018.

[7] Y. Yu, B. Kwan, C. Lim, S. Wong, and P. Raveendran, "Video-based heart rate measurement using short-time fourier transform," in *2013 International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 704–707, Nov 2013.