



REAL-TIME AUTOFOCUS SYSTEM FOR AIRBORNE APPLICATIONS USING 2D-FFT

Md. Shahid, Krishna V.Vamshi & Mahule Vibhor

Aeronautical Development Establishment, DRDO, Bangalore, India

ABSTRACT

This paper proposes a system that achieves automatic focusing of a target for aerial surveillance applications. In airborne environment, image gets defocused due to continuous variation of slant range and operation of zoom. Thus there is need to minimize the defocusing of the image during the mission. The proposed methodology utilizes novel frequency domain approach using 2-D Fast Fourier Transform [FFT] of video frames to obtain the varying focus parameter. The 2-D FFT module is highly optimized and customized to execute on the TMS320F2812 DSP based video platform, in real time. It exploits redundancy of FFT implementation in accordance with core resources of processor.

FFT provides the extent of frequency content present in the image, from which the focus parameter is calculated. The focus parameter is given by the sum of the energy of all the frequency components excluding the DC component. Higher the value of the focus parameter simply means the image is properly focused. Else the motor control drive lens focus is adjusted in accordance with a pre-defined motor control algorithm to obtain best focus of the target image.

KEYWORDS: Fast Fourier Transform, DC component, Unmanned Aerial Vehicles, line of sight

INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are increasingly being used for surveillance and reconnaissance. The camera is mounted on a turret which is steered from Ground control Station via RF command up-link. In airborne surveillance the operator has option to steer camera line of sight [LOS] to acquire the target of interest. In addition, operator has option to change image magnification to perform various levels of target acquisition. The camera LOS movement causes variation in slant range continuously, which may cause defocusing of the image. This defocusing shall be corrected with the help of computing focus parameter continuously. In object recognition or tracking systems it is necessary for the input images to be properly focused for further applications viz. target acquisition and tracking.

Manual operation of 'focus' from GCS suffers from positional hunting of the lens 'focus' drive due to involvement of 'up' and 'down' links. Additionally, ground based auto focussing suffers with reduced video quality due to bandwidth constraints with video RF link. These limitations necessitate the utilisation of 'Auto Focus' feature within the imaging system. Onboard system involves challenges in terms of processing power, memory, airworthy resources, real time etc.

Certain image and video processing applications such as noise removal, filtering, compression and pattern recognition utilize the properties that are not directly visible in the time or spatial domain. As such, conversion to a frequency domain representation is essential for obtaining accurate output. It

also results in a more efficient implementation. An application where this is found to be useful is for the purpose of auto focus. Unrolling techniques for FFT implementation is carried out for 1 dimensional data [1] & 2 dimensional data [2] respectively. Efficient pruning the output samples of the 2-D DFT, based on the radix-(2x2) 2-D DIT FFT is developed by Saad [3]. Split vector radix type 2 FFT is proposed by Soo [4]. Generalized method for pruning an FFT type of transform is discussed at [5] - [6]. The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression etc.

This paper proposes a frequency domain approach using 2D FFT that is applied to an auto focus system. Key contributions of this paper involve efficient implementation by employing FFT redundancy and exploiting resources of DSP processor. The paper is outlined as follows: Section II gives a brief overview about the proposed method to Autofocus the target using 2D FFT. Section III gives a detailed description about implementation of 2D FFT followed by methods used to optimize 2D FFT. Section IV & V describes the implementation of Auto focus system & involved control logic. Section VI discusses experimental results and section VII concludes with discussion.

AUTOFOCUS SYSTEM USING 2D FFT

The Auto focus operation is based on a feedback system where the controlling parameter is the high frequency content in the video. A simplified version of the imaging system is shown in Fig. 1. The position of the lens determines the

degree of focus in the production of the digital image. The lens will give a sharp image at a certain ideal position. The position of the lens is controlled by a motor mechanism. A comparator in turn controls this with one of its inputs from the auto focus block. While tracking targets, it the auto focus system intelligently adjusts the lens ‘focus drive’ to achieve focused image of the target under the dynamic situation as stated above.

A frame is first captured by the imaging system and then passed to the Auto focus block. 2D FFT is applied on center portion of image for frequency domain representation. The high frequency content is obtained from certain coefficients of the FFT. The comparator determines whether the desired focus has been obtained by comparing the output parameter and an adaptive pattern. Based on the difference between these two values the motor mechanism is made to move the lens to a position where the image is expected to be sharper.

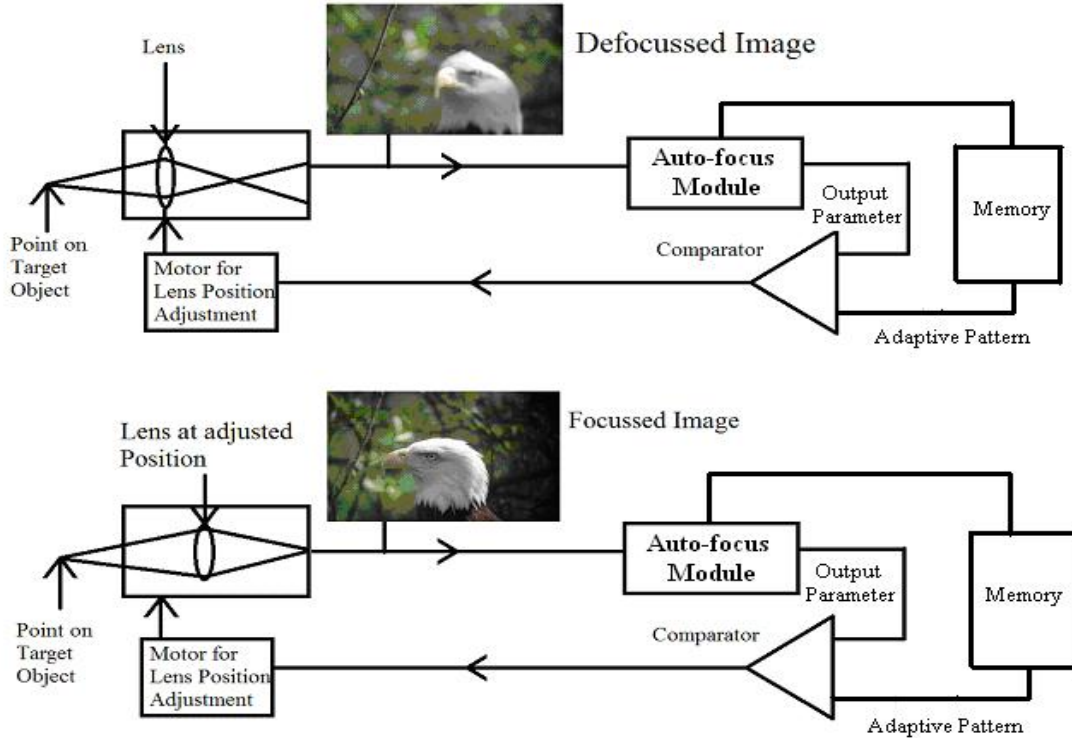


Fig 1 Block Diagram of the Autofocus System using 2D FFT

IMPLEMENTATION OF 2D-FFT

The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image. [3]- [4]

The DFT (Discrete Fourier Transform) is the sampled Fourier Transform and therefore does not contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial domain image. The number of digital frequencies corresponds to the number of pixels in the spatial domain image, i.e. the image in the spatial and Fourier domain is of the same size.

For a square image of size N×N, the two-dimensional DFT is

$$F(k,l) = \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} f(a,b) e^{-i2\pi f \left(\frac{ka}{N} + \frac{lb}{N} \right)} \dots(1)$$

where f(a,b) is the image in the spatial domain and the exponential term is the basis function corresponding to each point F(k,l) in the Fourier space. The equation can be interpreted as: the value of each point F(k,l) is obtained by multiplying the spatial image with the corresponding base function and summing the result.

The basis functions are sine and cosine waves with increasing frequencies, i.e. F(0,0) represents the DC-component of the image which corresponds to the average brightness and F(N,N) in a similar way, the Fourier image can be synthesized to the spatial domain. The inverse Fourier transform is given by

$$f(a,b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l) e^{i2f\left(\frac{ka}{N} + \frac{lb}{N}\right)} \dots (2)$$

The term $\{1/(N^2)\}$ is for normalization in the inverse transformation. This normalization is sometimes applied to the forward transform instead of the inverse transform, but it should not be used for both [2].

Separable Property

To obtain the result for the above equations, a double sum has to be calculated for each image point. However, because the Fourier Transform is separable, it can be written as

$$F(k,l) = \frac{1}{N} \sum_{b=0}^{N-1} P(k,b) e^{-i2f \frac{lb}{N}} \dots(3)$$

Where

$$P(k,b) = \frac{1}{N} \sum_{a=0}^{N-1} f(a,b) e^{-i2f \frac{ka}{N}} \dots(4)$$

Using separable properties, the spatial domain image is first transformed into an intermediate image using N one-dimensional Fourier Transforms. This intermediate image is then transformed into the final image, again using N one-dimensional Fourier Transforms. Expressing the two-dimensional Fourier Transform in terms of a series of 2N one-dimensional transforms decreases the number of required computations.

Methods used to optimize the 2-D FFT Computation

FFT is an optimized technique to implement DFT for real time applications. It reduces order of computation from N^2 to $\log_2 N$.

a. Effective memory management

Cache misses or excessive data movement between registers and memory can greatly slow down an FFT computation. An In-place algorithm has been implemented to reuse the data memory throughout the transform, which can reduce cache misses for longer lengths [3].

b. Special butterflies

The butterflies involving $W_0^N=1$ twiddle factor, can be implemented without additional operations, or with fewer real operations than a general complex multiply. In our implementation these butterflies are computed with simple add and subtract operations.

c. Fast bit-reversal

Bit-reversing the input or output data can consume several percent of the total run-time of an FFT program if coded inefficiently. In our implementation, we have used a method where the values of a given array (which is a row/column in the image) are exchanged with the values of the next row/column. The exchange process [1] takes place in bit-reversed manner as shown in Fig. 2 and Fig. 3.

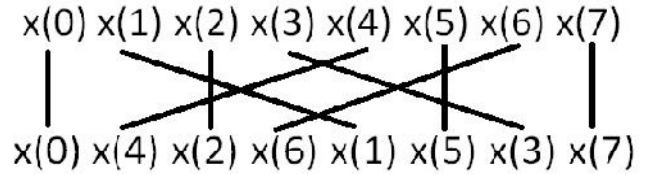


Fig.2 Bit reversal using separate

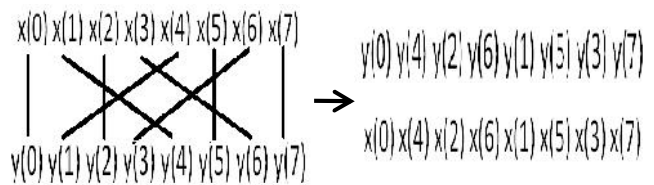


Fig.3 Bit reversal using next row as temporary array

i. Two rows FFTs for the price of one row

For real sequences of two rows, $x(n)$ and $y(n)$ we compute both FFTs simultaneously by considering a composite sequence $z(n)$ as given in Eqn. (5). Eqn. (6) represents frequency decomposition of input sequence $z(n)$.

$$z(n) = x(n) + j*y(n) \dots (5)$$

$$Z(k) = \text{FFT}[z(n)] \dots (6)$$

$$X(k) = (Z(k) + Z^*(N-k))/2 \dots (7)$$

$$Y(k) = -j.(Z(k)-Z^*(N-k))/2 \dots (8)$$

Eqns. (7) and (8) can be used to recover $X(k)$ and $Y(k)$ from $Z(k)$.

ii. Overall Symmetry of 2D DFT

The final 2D DFT contains a lot of redundant information. We can see from X_r and X_i in Fig. 4, the areas surrounded by rectangles contains duplicate information and need not be stored. The real and imaginary parts of the final result have been combined into a single array in our program as discussed later.

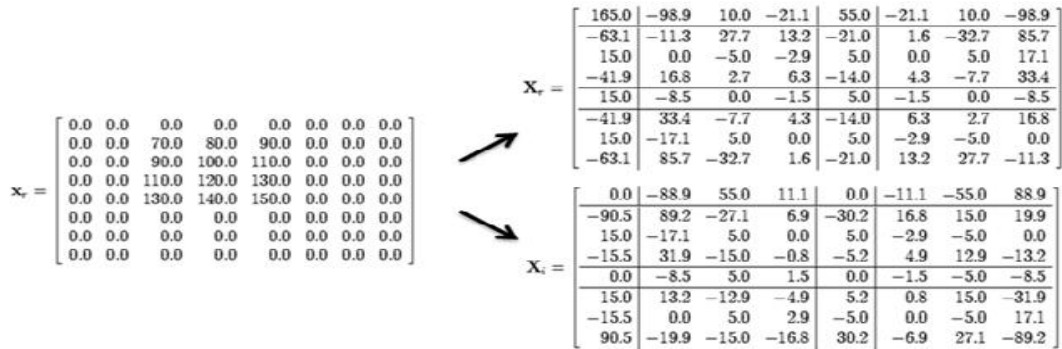


Fig.4 Redundant information in 2D FFT

IMPLEMENTATION

The video processing platform consists of on video decoder, encoder, TMS320F2812 DSP and Xilinx FPGA chip. The 2D- FFT is implemented on this video processing platform. The continuous image frames from the camera are fed as input to the video processor platform. The images by the camera are captured in PAL format at the rate of 25 frames per second (i.e., the processor gets an image at an interval of every 40ms). The detailed block diagram is shown in Fig. 5. Video platform is followed by a buffer. This buffer isolates video platform from power amplifier. Power amplifier drives focus lens motor of zoom lens. Camera output to video platform, acts as a feedback in this system. Efficient utilization of core

resources of processor in assembly language and ignoring the coefficient’s calculation relevant to inverse FFT, helped to achieve desired real time performance (40 ms). Table-I shows a break-up of involved computation time under ‘C’ language, Assembly language and Assembly language with optimal implementation (DMAC, PMAC, SQRT etc.). Efficient utilisation of core resources provided single cycle performance with minimum overheads. Processor consists of single 32 bit multiplier, but our data is 16 bits. DMAC instruction provided two 16 bit multiplications in effectively half cycle duration. Similarly, PMAC provides copying one row/column from data memory to program memory without overheads.

Table-I Processing time comparisons

‘C’ language Implementation	‘ASM’ language Implementation	‘ASM’ language with optimisation
80 mSec	50 mSec	32 mSec

Implementation has advantage since reconstruction is not involved therefore require only magnitude component has to be stored. Phase component is ignored during column FFT operation. This reduces computation power & memory requirement. The 2-D FFT module is highly optimized and

customized to execute on the TMS320F2812 DSP based video platform, in real time. The focus parameter is given by the sum of the energy of all the frequency components excluding the DC component. This sum has been found to vary linearly with the perceived degree of focus of an image

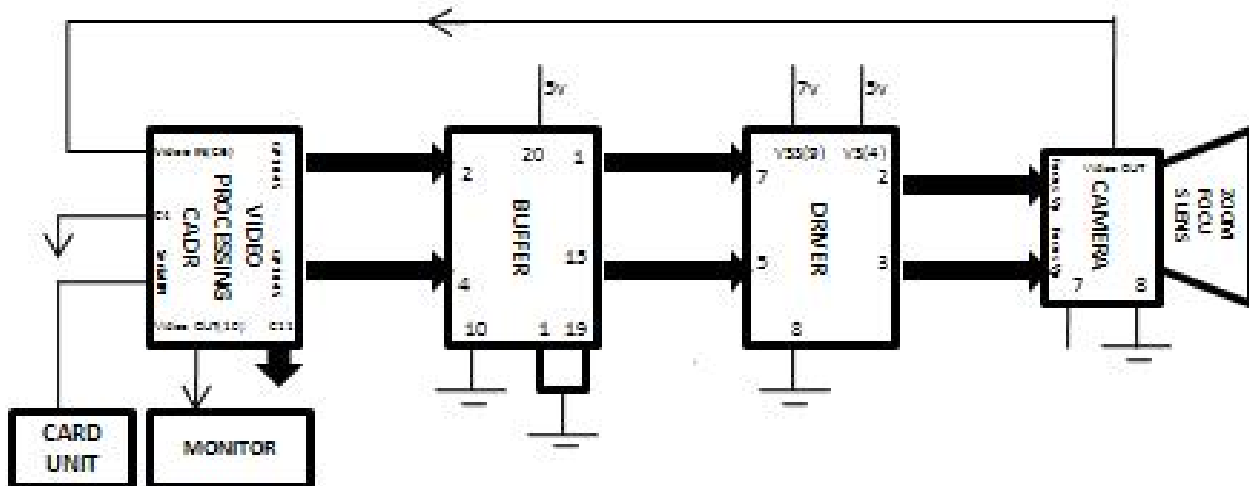


Fig.5 Detailed Block Diagram of Auto-focus System

The DSP processor is also loaded with the CONTROL ALGORITHM, which controls the direction of rotation of the motor controlled focusing lens. The lens rotates in clockwise direction with logic 1 (3.3v) and logic 0 (0v) are on the pins GPIOA5 and GPIOA6 respectively. It rotates in anti-clockwise direction when the processor outputs voltages of 0v and 3.3v on pins GPIOA5 and GPIOA6 respectively. DSP processor GPIO5 and GPIO6 pins output 3.3V (LVTTTL logic) whereas power amplifier requires 5V (TTL logic). Buffer is required to isolate video processor platform from driver in addition to level conversion (3.3V to 5V).

CONTROL LOGIC

The control unit is connected via GPIO pins of the TMS320F2812 board. The autofocus algorithm begins when the switch is closed. The operator in the base station can choose to run the autofocus algorithm, by sending a command that closes the DIP switch on board. This provides more control over the focusing operation. Upon closing of the switch, the following algorithm is executed

STATE 1: The focusing lens is pushed to the nearest end of the entire focus position range (say anti-clock wise)

STATE 2: The lens is moved in the opposite directions (clockwise) up to the other end of the focus position range. During its retrace, the following tasks are performed. 2D-FFT of the images acquired is calculated and the focus parameter is calculated. The focus parameter $\{H[f]\}$ is calculated by summing the energy of all the frequency components excluding the DC component. The focus parameters are stored in a memory. The parameter stored at an instant 'n' is compared with the parameter calculated at "n+1" instant. The greater of the two is stored in the memory location withinstant. This process continues till the end of the focus range is reached. At the end of State 2, the maximum value of the focus parameter $\{H[f]_{max}\}$ is available (stored) in the memory.

STATE 3: In this state, lens is rotated in anti-clockwise direction. The frequency parameter $\{H[f]\}$ is again calculated and is compared to the maximum value of the focus parameter $\{H[f]_{max}\}$. When a value $\{H[f]\} = 0.98\{H[f]_{max}\}$ is obtained the motor is stopped and best focus is attained. The focus parameter maxima are used in a motor control algorithm, which positions the focusing lens to achieve best focus of the target. During clockwise & anti-clock wise scanning (4-5 seconds), tracker in loop ensures that target remains in the centre of instantaneous video frames.

EXPERIMENTAL RESULTS

The Fujinon zoom lens is integrated with Pulnix765i CCD camera. It provides video in CCIR format at 25fps. Each pixel in the image has a depth of 8 bits. The frames are stored in a buffer with 16 bits per pixel. A windowed operation (128 pixels x 64 lines) extracts center portion of the instantaneous frame of video. This windowing operation is implemented in FPGA and contents are read by DSP processor. Windowed portion of the centre of the image is sent to the internal SRAM

These output voltages cannot drive the lens directly as there are chances the camera might drive more current from the board. This leads to LOADING of the DSP processor and in turn may result in damage of the apparatus. Therefore, buffer output is fed to the power amplifier. The power amplifier produces a maximum of 4A current. The output from the power amplifier is obtained at pins 2 and 3. The voltage level at these pins vary alternatively from Logic '0' to Logic '1', and are fed to the focus control input pins (5 and 6) of the lens. Instantaneous video output of camera is used as feedback for closed loop operation.

of DSP processor. The 2D FFT of this input is calculated as discussed above. The Focus parameter is calculated by summation of the energy of all coefficients except the DC components.

While tracking the variation of the Focus Parameter we store the position corresponding to its highest value. The Focusing lens is brought back to the position which gives a similar or greater value of the Focus parameter of instantaneous video frame.

CASE 1

The given input sweep consists of 115 frames (4-5 seconds).

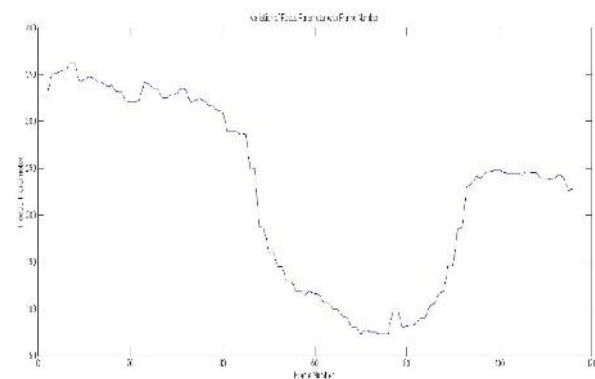


Fig.6 Variation of Focus Parameter with Frame Number for Case 1

Fig. 6 shows the variation of the calculated Focus Parameter with the incoming frames. Once the motor has finished its scanning sweep it estimates the maximum value of the Focus Parameter to be around 350 (10th frame of instantaneous video). It starts the reverse sweep to approach a value approximately equal to or greater than 98% of the maximum value.

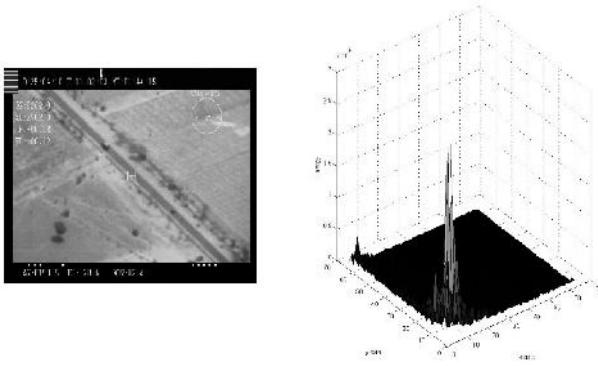


Fig. 6a Best Focus Position Image - Original video frame number 10 (left) and spectrum derived from 2D FFT of 128x64 windowed portion (right)

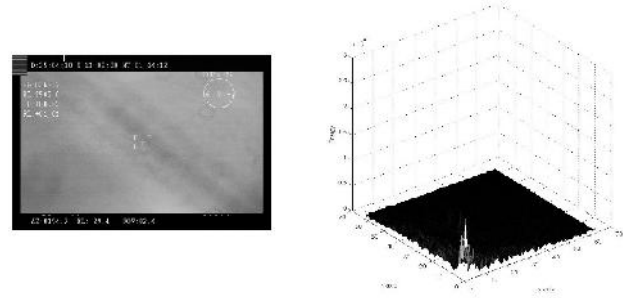


Fig.6b Worst Focus Position Image - Original video frame number 70 (left) and spectrum derived from 2D FFT of 64x128 windowed portion (right). Best and worst focus position with original video frame and spectrum are shown in Fig. 6a and Fig. 6b respectively.

CASE 2

Peak value of focus parameter for another video sequence is achieved at later side of scanning mechanism as shown in Fig.7. The visual representation of frames for best and worst frame positions are shown in Fig. 7a and Fig. 7b.

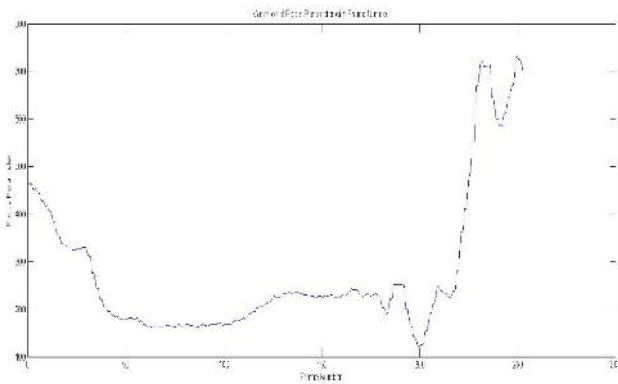


Fig.7 Variation of Focus Parameter with Frame Number for Case 2

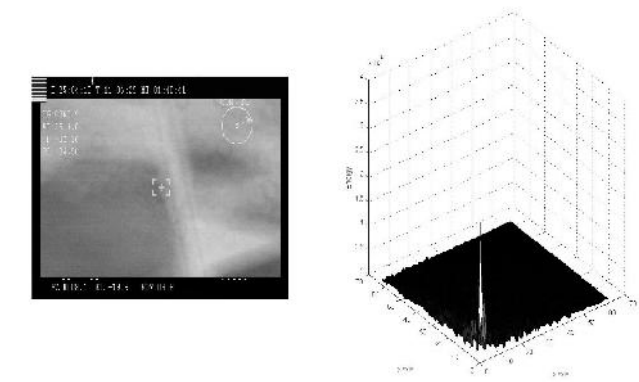


Fig.7b Worst Focus Position Image - Original video frame number 200 (left) and spectrum derived from 2D FFT of 64x128 windowed portion (right).

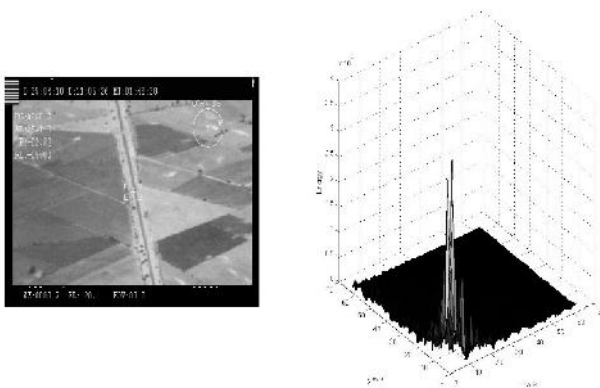


Fig.7a Best Focus Position Image - Original video frame number 230 (left) and spectrum derived from 2D FFT of 64x128 windowed portion (right).

CONCLUSION AND DISCUSSIONS

This paper successfully uses the merits of analyzing and processing an image in the frequency domain. One of the major applications of this concept is in Auto Focusing of Images acquired from the UAV's for real time airborne applications. This 2D Discrete in Time (DIT) -FFT module is implemented on the TMS320F2812 DSP platform for real time applications. This paper minimizes loss of video quality and delay effects associated with RF link. Frequency domain approach yields accurate output consuming less time. This method is effectively implemented to overcome the loopholes of the existing techniques (manual, lookup table etc.), and hence apt for real-time airborne applications. Moreover, the performance achieved with this low end control specific DSP processor TMS320F2812 (150 MIPS) is on par with high end video processors.

Scanning lens across the entire patch takes 4-5 seconds. This is the case at the power on for first time. Initially, starting point (focused/defocused) is not certain. During subsequent auto focusing arose due to slant range variation and zooming operation, is indicate that best focus position is moved out to

nearby positions. Therefore complete scanning is not required until unless large variation in slant range and zooming carried out. Target is assumed to be in center portion of video which is achieved by tracker in loop. More processing power platform can work for entire frame instead of a portion of instantaneous video frame. In this work, Focus parameter is summation over entire spectral range, instead a weighted pattern (bell shaped curve) may be used.

This 2D FFT module can be extended to phase correlation algorithm, which requires frequency domain representation of template & search images, along with inverse FFT. This system is developed to cater for PAL standard video(40 mSec frame update rate). However fast moving platform involves higher frame rate cameras requiring more sophisticated airworthy processing platform.

ACKNOWLEDGEMENT

The authors thank Director, ADE and also convey sincere thanks to Dr. RK Mishra, Sc-F(retd.) for valuable guidance.

REFERENCES

- [1] H. Kee, N. Petersen, J. Kornerup, and S. S. Bhattacharyya, "Systematic generation of FPGA-based FFT implementations," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, Nevada, March 2008, pp. 1413–1416.
- [2] "A Resource-efficient Acceleration of 2-Dimensional Fast Fourier Transform Computation on FPGAs", by Hojin Kee and Shuvra S. Bhattacharyya, Newton Petersen and Jacob Kornerup. IEEE 2009.
- [3] Saad Bouguezel, M. Omair Ahmad, M.N.S. Swamy "Efficient Output-Pruning of the 2-D FFT Algorithm" ISCAS conference 2004, vol. III, pp. 285-288.
- [4] Soo-Chang Pei, and Ja-Lin Wu, "Split Vector Radix 2D FFT", ICASSP 1987, pp. 1987-1990 .
- [5] S. R. Rangarajan and S. Srinivasan, "Generalized method for pruning an FFT type of transform", IEEE Proc. on Vision, Image and Signal Processing, 1997, Vol. 144, No. 4, pp. 189-192.
- [6] ackledge, Digital Image Processing. Horwood Publishing, 2005.