UNSYS digital

Journal of Unmanned System Technology

Development of Raw Data Processing System for CP-SAR On-board JX-2 UAV using Mobile Heterogeneous Computing

Bambang Setiadi[†‡], Good Fried Panggabean[‡], Josaphat Tetuko Sri Sumantyo[‡], Koo Voon Chet[§]

[†]Lembaga Ilmu Pengetahuan Indonesia, Indonesia

[‡]Center for Environmental Remote Sensing Chiba University, Chiba, Japan

[§]Faculty of Engineering and Technology, Multimedia University, Malaka, Malaysia

Abstract—This paper presents the development of unmanned aerial vehicle (UAV) -based Circularly Polarized Synthetic Aperture Radar (CP-SAR) raw data processing system for the experimental JX-2 UAV. First, we present the design and outline of CP-SAR system, followed by derivation of resolution and raw data rate value from axial ratio and geometry. Following the signal processing requirements, we propose to use mobile heterogeneous platform for on-board raw SAR data processing system. Then we describe the raw data processing steps based on Range-Doppler Algorithm for stripmap mode UAV SAR. Experiment results using a middle-end CPU-GPU mobile heterogeneous hardware development kit with CUDA programming interface and raw data input from actual UAV mission demonstrate the capability of the proposed mobile heterogeneous platform for SAR processing on-board JX-2 UAV.

Keywords—Synthetic aperture radar, SAR processing, Range-Doppler, Mobile Heterogeneous Computing, CPU, Graphic Processing Units, GPU, Common Unified Device Architecture, CUDA.

DOI: 10.21535/just.v4i2.914

I. Introduction

SYNTHETIC Aperture Radar (SAR) is a renowned technology in remote sensing offering the ability to observe earth during day and night under all weather conditions [1]. It offers high resolutions images for various remote sensing applications such as geosciences, climate change, environmental monitoring, two three and four dimensional mapping, change detection, disaster mitigation up to planetary explorations [2].

Currently available SAR sensor mostly utilized linearly polarized wave which produces horizontal and vertical polarization combinations namely HH, HV, VH and VV in full polarization mode. Different with this trend, a SAR system that uses elliptical polarized wave is currently under development by the Center for Environmental Remote Sensing in Chiba University. The usage of elliptical polarized wave in a system called Circularly Polarized SAR (CP-SAR) is expected to open new possibilities in remote sensing applications offered by this scattering phenomena. **Table 1** lists the various experiments and studies in remote sensing proposed by CP-SAR polarimetry [3].

Corresponding author: Bambang Setiadi (e-mail: bambang@gmail.com).

This paper was submitted on June 28, 2016; revised on August 3, 2016; and accepted on August 14, 2016.

Presently we are developing the CP-SAR sensors for various electromagnetic wave bands (L, C and X bands). Initially, the sensors will be deployed on Unmanned Aerial Vehicle (UAV) before launched on microsatellite. UAV was selected due to its cost-effective development, lower risk, and agile operations, compared to conventional aircrafts and satellites [4]. However, small sized and lightweight UAV has major limitations related to the sensor payload: size, weight and power (SWAP), which dictates the further development of the sensor [5][6].

Table 1 CP-SAR Objectives

Field	Item	Details
Basic SAR Experiments	Scattering mechanism of CP microwave	• Studying scattering mechanism from vegetation, cryosphere, soil and rocks, desert, etc.
	Interferometry	• Linear versus circular polarization interferometric SAR • DEM extraction using CP wave
	Axial ratio image	• Vegetation, geologic, cryosphere mapping using axial ratio image
SAR Applications	Land cover mapping	• Forest/non-forest area classification • Paddy field/wet land extraction • Mangrove area mapping • Snow - iceberg detection
	Disaster monitoring	• Earthquake • Volcano eruption • Flood, forest fire, landslide, etc.
	Cryosphere monitoring	• Iceberg, glacier, • Arctic routing
	Ocean monitoring	• Oil spill • Ocean wave monitoring

A small UAV named Josaphat laboratory experimental UAV (JX-2) becomes the platform for CP-SAR sensor under development. This platform will have ground speed of approximately 30-40 m/s with operating altitude of 1-4 km and flight endurance up to 4 hours. The UAV has wing span of 6 m, body length of 4.75 m, and total weight of 146 kg including 25 kg payload, i.e. the SAR sensor, ground positioning system (GPS), inertial measurement unit (IMU) and data communication link. **Figure 1** shows JX-1 UAV, the predecessor

of JX-2 UAV which has similar specification and payload capacity.

Besides the antennas, signal generator, and the radio frequency, the raw data processor subsystem inside a Synthetic Aperture Radar (SAR) system is another important part that determines the final SAR image quality. Its main objective is to convert raw data produced by the sensor into two dimensional image ready for various remote sensing applications. This image formation tasks are usually done digitally using high performance computing (HPC) system due to the complex computations and large amount of data involved. Desktop computers with CPU and GPU, clusters of networked computers and super computers comprise the vast majority of hardware platforms used for processing raw SAR data [7]-[10]. In current work, we developed raw data processing system to convert raw data from CP-SAR sensor on-board JX-2 UAV into two dimensional image. The processing makes use of the Range Doppler image formation algorithm for stripmap mode SAR which is one of the most popular and mature frequency domain processing algorithm [11].

Depending on its system specification, SAR sensors generally produce large amount of raw data during operations. Due to the complex computation of the image formation algorithm, processing large amount of raw data into images requires high performance computing which is usually done on the ground station instead of on-board the platform.

Although rapid digital computers are currently already available, highly desirable feature such as real-time processing F-SAR image without using super computer is still rapidly evolving under development.

Limitations on size and weight of payload have made it difficult to use conventional HPC for SAR processing onboard UAV. Consequently, high performance embedded computing (HPEC) such as special purpose digital signal processors (DSPs), application-specific integrated circuits (ASIC), field programmable gate array (FPGA) and lightweight single board personal computers become a suitable solution candidates for this problem [12]-[14].

Miniaturization of SAR sensor system is an answer to the challenge of SWAP on lightweight SAR platforms. Currently there are efforts to develop smaller SAR sensor subsystems including smaller antennas, radio frequency (RF) subsystem, signal generators and raw data processing. Onboard raw SAR sensor data processing (or onboard SAR processing) as one of key part of SAR sensor system is also significantly influenced by this challenge [5][15]-[19].

The mobile heterogeneous computing platform which integrates CPU and application-specific accelerators such as GPU and DSP, offers opportunity to accelerate compute-intensive application using low powered processor with light-weight form factor [20]. As part of the CP-SAR development effort, we are exploring the potential of using CPU-GPU computing available on mobile heterogeneous computing platform to develop on-board SAR raw data processing systems for JX-2 UAV.

Despite the facts that such heterogeneous computing platforms with CPU and GPU as those available on desktops and servers are widely available and have been used for accelerating SAR raw data processing, minor approach has been done to use the mobile heterogeneous computing platform for on-board SAR processing. Mobile heterogeneous computing platforms which integrates energy efficient CPU and GPU, provides opportunity to accelerate SAR processing onboard lightweight UAV platforms with SWAP limitations.

The current work focused on the feasibility of using mobile heterogeneous computing platform for processing raw SAR data. We report on the development process of implementing the SAR processor and its runtime performance compared to the speed of raw data generation.

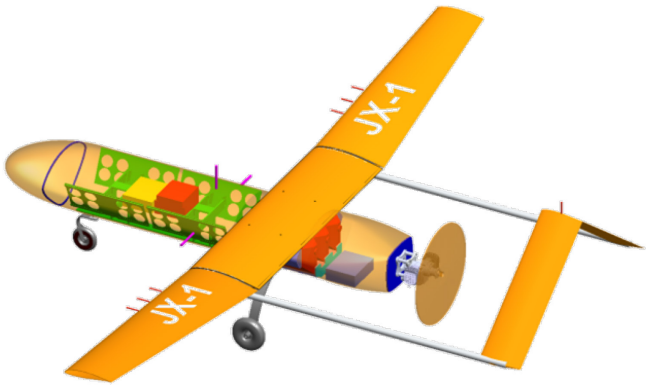


Figure 1 Illustration of JX-1 UAV with SAR Payload

The remainder of this paper is organized as follows. Starting with background discussion on the CP-SAR development status, stripmap SAR, Range Doppler algorithm, and a brief introduction to the mobile heterogeneous CPU/GPU platform architecture and Compute Unified Device Architecture (CUDA) programming paradigm, then we present the algorithm design and implementation strategies. This is followed by discussion of the implementation of the SAR processor on mobile heterogeneous CPU/GPU platform. Finally, we use actual SAR data from CASIE 2009 [21] and [22] to examine the performance of our implementation and present the resulting image, followed by discussion and conclusion.

II. Development of CP-SAR System for UAV

This section presents a brief introduction to the CP-SAR system outline, axial ratio and geometry, resolutions, raw data generation rate, and signal processing requirements.

A. System Outline

In terms of system requirement, the JX-2 CP-SAR is a frequency-modulated pulse radar working in L-band with a center frequency of 1.27 GHz. It will be operated at flight altitudes of 1 to 4 km, operating in the stripmap SAR acquisition mode with off-nadir angles between 40° to 60°. The system is

designed to be able to produce SAR image with 1 to 10 m ground resolutions. The pulse duration is designed to be 3.9 to 23.87 ms with PRF of 1000 Hz with interleaving RHCP and LHCP transmit polarization in the full circular polarimetric mode.

On-board the JX-2 UAV platform, the SAR sensor's transmitter and receiver antennas are positioned on the right side of the UAV. Each antenna consists of right-handed CP (RHCP) and left-handed CP (LHCP) subunits forming a four panel with the total size of 1.5 m in elevation and 0.4 m in azimuth direction [3][23]. A block diagram of the system is shown in **Figure 2**. The CP-SAR sensor system is composed of a chirp signal generator, RF Module, CP antenna, analog-to-digital converter/data recorder, SAR processor, motion sensing module and data downlink module. The CP antenna is connected to the RF module, while the motion sensing module contains an inertial measurement unit (IMU) that receives position signals from GPS antenna. In order to communicate with the ground station, a data link antenna is connected to the data downlink module.

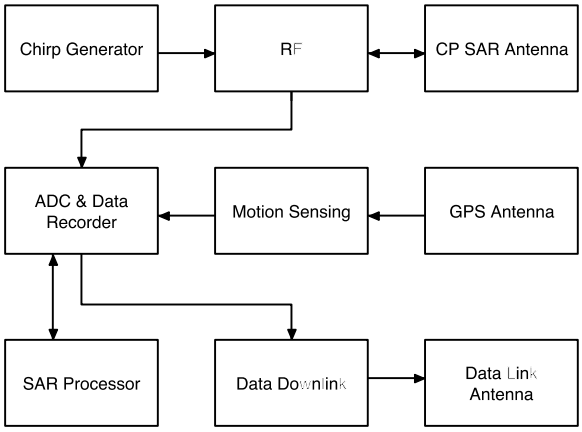


Figure 2 Block diagram of CP-SAR System on JX-2

TABLE 2 CP-SAR SYSTEM SPECIFICATION

Parameter	Symbol	Value
Operating altitude	H	1 – 4 km
Platform speed	v	100 km/h
Center frequency	F	1.27 GHz
Wavelength	λ	0.2362 m
Off nadir angle	θ_0	40° – 60°
Antenna Width	W	0.75 m
Antenna Height	L	0.4 m
Azimuth beam width	$\theta_{az\text{-}cp}$	7.94°
Range beam width	$\theta_{el\text{-}cp}$	29.78°
Ground Resolution (Rg)	δ_{rg}	1 m
Ground Resolution (Az)	δ_{az}	1 m
Swath width	W_{gr}	1000 m
Observation length	L_O	5 km
PRF	PRF	1000 Hz

The RF subsystem consists of a transmitter, receiver, local oscillator, and low noise amplifier, weighing approximately 10 kg. The transmitter mixes the 150 MHz bandwidth chirp up to 1.27 GHz for transmission. The receiver and local oscillator are used to mix the RF radar return from the antenna to an offset baseband, and amplify it so that it can be sampled by the ADC subsystem.

The signal generator subsystem generates chirp signal as the input for RF subsystem. It is implemented using a field programmable gate array (FPGA) to enable rapid modification to the generated chirp signal. The chirp signal generator must be able to generate chirp signal with various configurations depending on the mission requirement such as flight height and platform speed. The parameters of generated chirp signal that can be modified are: chirp bandwidth, pulse duration and pulse repetition frequency (PRF). Direct digital synthesizer (DDS) technique is used for the chirp signal generation process because it has many advantages compared to its analog counterpart. **Figure 3** shows an example of real and imaginary components of the chirp signal created by the signal generator subsystem [24].

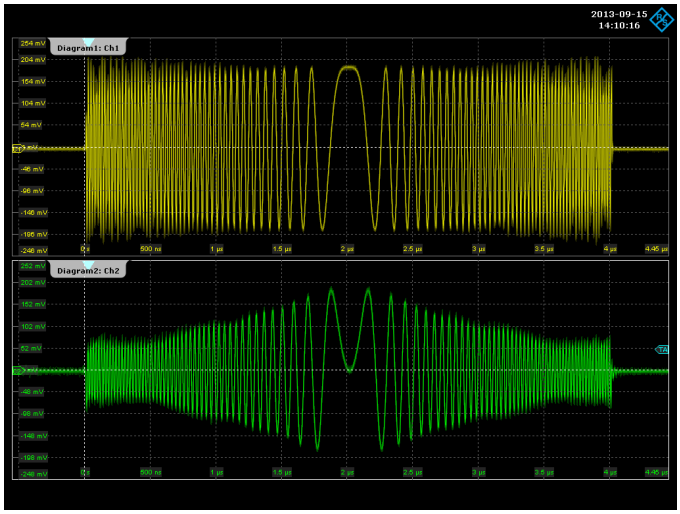


Figure 3 Chirp Signal Waveform in Real and Imaginary Form

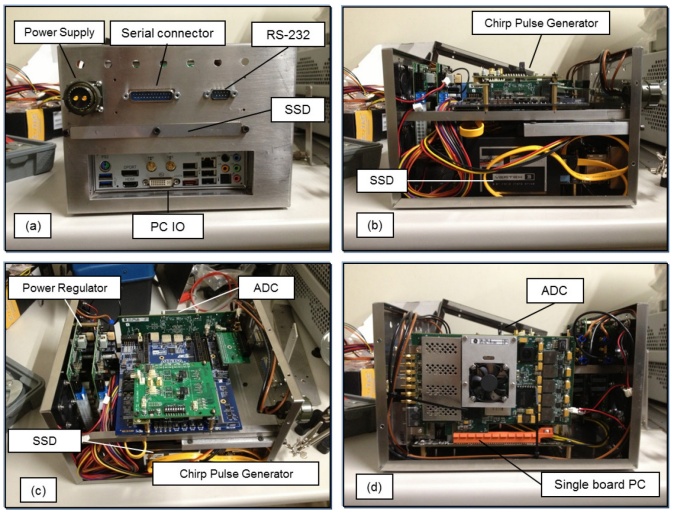


Figure 4 Block diagram of CP-SAR System on JX-2

The analog-to-digital converter (ADC) and data recorder subsystem handles the sampling and down-converting of signal received by the CP antenna. After down-converting to the baseband, the data are stored in a solid-state drive (SSD) storage. These data are known as raw SAR data or phase histories, and will be processed further using signal processing technique in order to produce a SAR image. **Figure 4** shows the photographs of chirp generator subsystem, analog-to-digital converter/data recording subsystem, and the SAR processor on a compact single PC board.

B. Axial Ratio and Geometry

An active microwave sensor can be operated in one of the three polarization modes of linear, elliptical or circular polarization. An essential parameter that differentiates circularly polarized wave and its linear counterpart is the axial ratio (AR) value, which is defined as the absolute ratio between the major and minor axis length of the electric field amplitude, as shown in **Figure 5** [25].

$$|R| = \frac{\text{major axis length}}{\text{minor axis length}} = \frac{OA}{OB} \geq 1. \quad (1)$$

$$R(\text{dB}) = 20\log|R| \quad (2)$$

$$\varepsilon = \text{arccot}(-R) \ -45 \leq \varepsilon \leq 45 \quad (3)$$

In **Figure 5**, ε is the ellipticity angle and τ is the tilt angle of the wave. The value of ε is from -45° to 45° and that of τ is from 0° to 180°. The value of τ is related to the orientation of plane polarization vector of the microwave. When ε is equal to 0, the elliptical shape will degenerate into a straight line, or called as linear polarization (LP). This LP wave is horizontal polarization (HP) when τ is equal to 0 and vertical polarization (VP) when τ is tilted to 90°. According to (1), the value of R is 1 for circular polarization, infinite for linear polarization, and between one and infinite for elliptical polarization. The value of R is usually represented in decibel as in (2).

With respect to the SAR geometry, the difference between CP-SAR geometry and LP is in the definition of beamwidth of the antenna. In the conventional LP-SAR, the 3-dB half power antenna beamwidth is defined as an angular range in which the value of antenna gain is no less than -3 dB from the peak value. The 3-dB half power beamwidth can be estimated as

$$\theta_{el-LP} = 0.886\frac{\lambda}{W} \quad (4)$$

$$\theta_{az-LP} = 0.886\frac{\lambda}{L} \quad (5)$$

where and θ_{az-LP} are the half power beamwidth in the elevation direction and azimuth direction, respectively; W is the antenna width, L is antenna length and λ is the radar wavelength.

Two important parameters of the SAR system are related with the value of half power beamwidth, namely, the value of θ_{el-HP} determines the ground swath width, while the value of θ_{az-HP} does the synthetic aperture length. This latter relationship can be written as

$$L_{sa-LP} = R_0 \cdot \theta_{az-LP} \quad (6)$$

$$\delta_{az-LP} = \frac{R_0 \cdot \lambda}{2L_{sa-LP}} \approx \frac{L}{2} \quad (7)$$

where R_0 represents the nearest distance to the target in azimuth plane view [11].

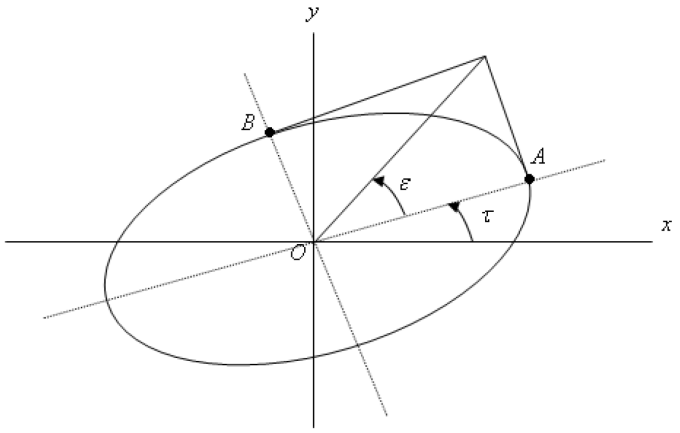


Figure 5 Polarization ellipse showing the major and minor axes with the definitions of angles ε and τ

Figure 6 shows the geometry of CP-SAR sensor onboard an airborne platform. Here, $\theta_{\text{el-CP}}$ is the 3-dB AR beam-width in the range direction. Since the beam-width of CP antenna is determined by the AR value, the CP-SAR geometry is also dictated by this value. The main difference of CP-SAR geometry from that of LP-SAR is that the calculation is using the circular version the beam-width.

The relationship between the circular and linear polarization beam-width can be expressed as

$$\theta_{el-CP} \leq \theta_{el-LP} \quad (8)$$

$$\theta_{az-CP} \leq \theta_{az-LP} \quad (9)$$

where θ_{el-CP} is the elevation AR beamwidth, and θ_{az-CP} is the azimuth AR beamwidth. In our implementation, we define circular polarized antenna as an antenna with the value of AR less than or equal to 3 dB. This is due to the difficulty of implementing antenna with AR value of zero which represent a completely circular polarization [23].

A. Resolutions and Raw Data Generation Rate

On the basis of the definition of CP-SAR geometry, the CP synthetic aperture length (LSA-CP) and the azimuth resolution (θ_{az-CP}) can be defined as

$$L_{sa-CP} = R_0 \cdot \theta_{az-CP} \quad (10)$$

$$\theta_{az-CP} = \frac{R_0 \cdot \lambda}{2L_{sa-CP}} = \frac{R_0 \cdot \lambda}{2L_{sa-CP}} \quad (11)$$

and the ground range resolution can be defined as

$$\delta_{gr} = \frac{2}{2B\sin\theta_i} \quad (12)$$

where c is the velocity of light, B is the radar bandwidth, and θ_i is the incidence angle. The ground swath width can be given approximately as

$$W_{gr-CP} \approx \frac{\theta_{el-CP} R_m}{\cos\theta_i} \tag{13}$$

This indicates that the ground swath width is determined by the elevation angle (θ_{el-CP}), center slant range (R_m) and incidence angle (θ_i) [26].

The amount of data received during data acquisition is dependent on the minimum flight distance, L_a, which related to the synthetic aperture length, $L_{sa\text{-}CP}$, from (10). **Figure 7** shows that the minimum flight distance, L_a, is equal to the desired image length, L_0, added with two times the half of synthetic aperture length:

$$L_a = L_0 + L_{sa-CP} \tag{14}$$

Using this value of minimum flight distance and the platform velocity (v), the data recording time, t_D, can be expressed as

$$t_D = \frac{L}{v} \tag{15}$$

which shows the time duration in which the SAR data are acquired.

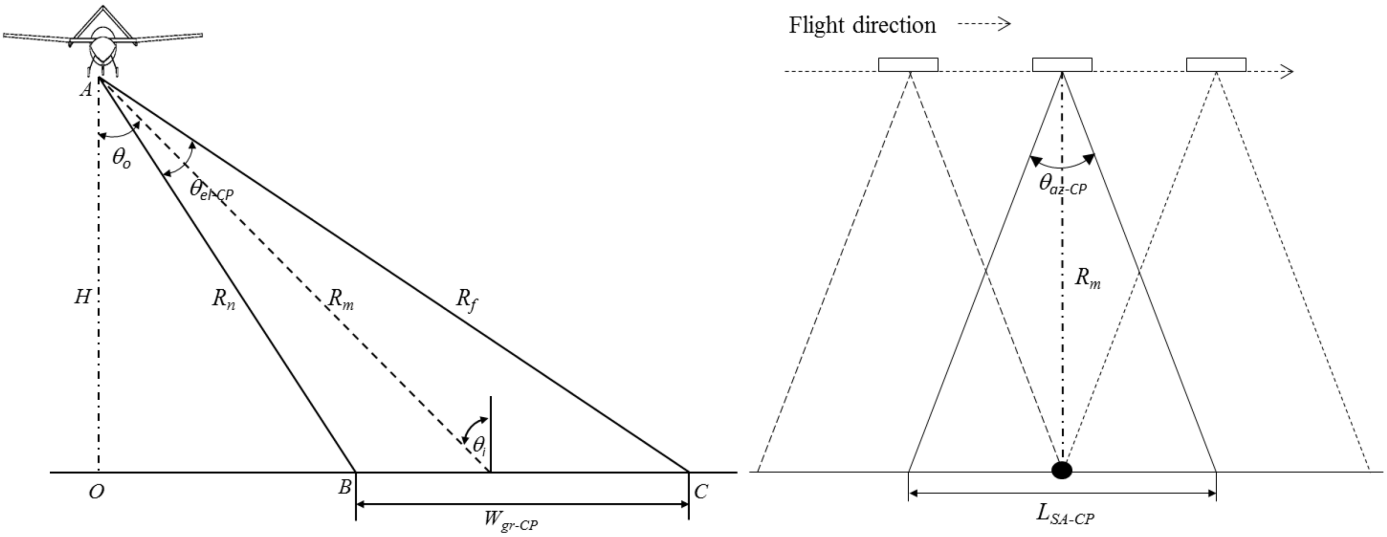


Figure 6 CP SAR Geometry: (a) range direction (b) azimuth direction

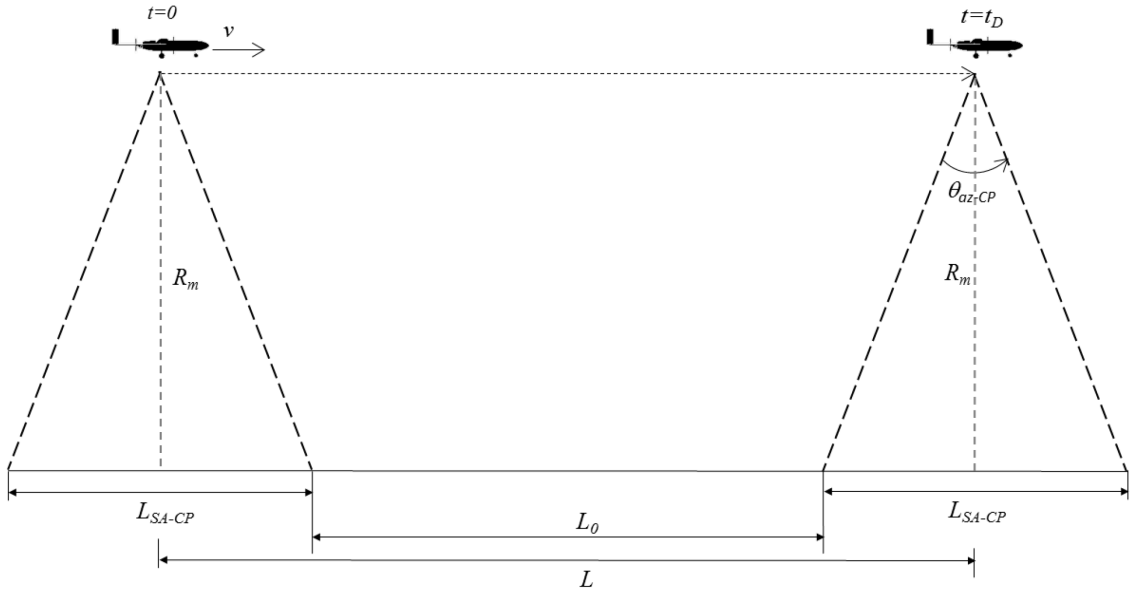


Figure 7 CP-SAR flight distance and time

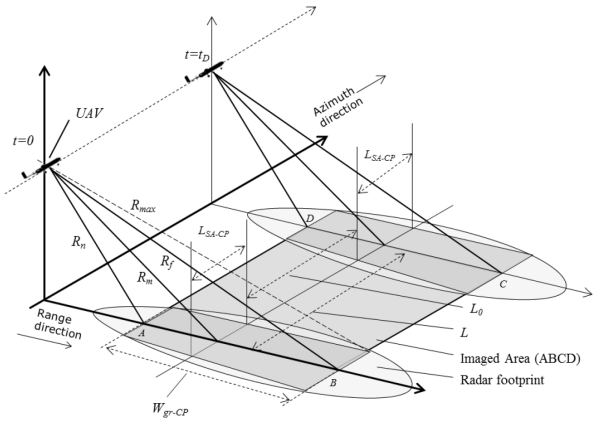


Figure 8 CP-SAR flight distance and time

Figure 8 illustrates the concept of the total imaged area on the ground when the UAV moves from time $t = 0$ to $t = t_D$. The observed area is denoted as *ABCD*, which causes the backscattering signal from ground area illuminated by the air-borne radar from time $t = 0$ to $t = t_D$. The area of imaged plane can be expressed by the product of the flight distance and desired swath width.

$$A_{\text{image}} = W_{gr-CP} \cdot L_a \tag{16}$$

C. Sampling Requirements

One of the major requirements of JX-2 CP-SAR is the ground resolution in range and azimuth direction, which is 1 m by 1 m. These values will dictate the values of other parameters, especially those related to the signal acquisition: pulse bandwidth, pulse length, and pulse repetition frequency (PRF).

In the RF subsystem, the sampling window start time (SWST) is calculated using the minimum and maximum values of the range between sensor and target. As depicted in **Figure 9**, the minimum range is the near range, R_n, and the maximum range is $R_{\max}$, which is defined as

$$\sqrt{R_f^2 + \left(\frac{L_{sa-CP}}{2}\right)^2} \tag{17}$$

The minimum and maximum range is used to derive the value of start and stop sample time given as

$$t_{start} = \frac{2R_n}{C} \tag{18}$$

$$t_{stop} = \frac{2R_{max}}{C} + \tau_p \tag{19}$$

The sampling window stop time is determined by $R_{\max}$ and the pulse length τ_p. The value of t_{start} must be larger than the pulse length to avoid collision with the end of the pulse. The time to switch the antenna operation from transmitting to receiving mode is known as the switch time τ_{sw}, and this imposes the limitation of the system [26] so the start time of receiving must satisfy

$$t_{start} = \tau_p + \tau_{sw} \tag{20}$$

In our current design, τ_{sw} is 3 μs. Another limitation on the pulse length is that the product between pulse length and bandwidth (time-bandwidth product) must be bigger than 100 [27].

$$\tau_p \cdot B > 100 \tag{21}$$

Finally, we have the sampling interval as the difference between sampling stop time value and the sampling start time

$$t_{si} = t_{stop} - t_{start} \tag{22}$$

The number of samples in range (n) and azimuth (m) directions determine the amount of raw data produced by the data recorder and how much data produced per second or data rate. This data rate, in turn, determines the minimum processing speed that the SAR processor must provide in order to be able to process the data in real time.

The number of samples in range direction depends on the sampling frequency used in the system. The CP-SAR uses in-phase and quadrature (IQ) modulation where the ratio between sampling frequency and pulse bandwidth is typically equal to 1.2 [11]:

$$\alpha = \frac{f_s}{B} = 1.2 \tag{23}$$

The sampling frequency is inversely proportional to the sample spacing of the pulse:

$$\Delta t = \frac{1}{f_s} \tag{24}$$

The number of samples in the range direction can be calculated as the proportion between the sampling interval (21) and the sample spacing:

$$n = \frac{t_{si}}{\Delta t} \tag{25}$$

Using the same principle, the number samples in the azimuth direction can be calculated by first calculating the sample spacing in azimuth direction, Δs_{az}, which is equal to the effective platform velocity divided by the PRF:

$$\Delta s_{az} = \frac{v}{PRF} \tag{26}$$

The down-sampling rate, K_{p}, is estimated using the value of PRF and Doppler bandwidth, BD [28]:

$$K_p \leq \frac{PRF}{1.2\, B_{Dop}} \tag{27}$$

$$B_{Dop} = \frac{2v \cdot \theta_{az-CP}}{\lambda} \tag{28}$$

Thus, the Doppler bandwidth directs the sampling requirements that define the lower limit of the PRF. The azimuth sample spacing after pre-summer filter value is equal to the value of azimuth sample spacing in (26) multiplied by the down-sampling rate. In CP-SAR design, the maximum value of K_p is used to obtain the azimuth sample spacing after presuming.

$$du = K_p \cdot \Delta S_{az} \tag{29}$$

with the number of sample in azimuth direction with the length of L can be obtained as [28]

$$m = \frac{L_a}{du} \tag{30}$$

The number of samples in the range and azimuth directions is related directly with the size of raw data, D_{size},

$$D_{\text{size}} = m \cdot n \cdot s_{\text{size}} \tag{31}$$

Here m is the number of sample in azimuth, n is the number of samples in range and s_{size} is the size of one sample point, which is in our implementation equal to two bytes for a complex sample point.

The data acquisition time required for an area with the azimuth length of L can be calculated as the ratio between the flight length La, and the effective platform velocity v:

$$t_D = \frac{L_a}{v} \tag{32}$$

where L_a is defined in **(14)**. The value of t_{D} is then used to calculate the estimated data rate per analog to digital sampling channel as

$$D_{\text{rate}} = \frac{D_{\text{size}}}{t_D} \tag{33}$$

By combining (24), (32), and (33), the data rate can be estimated as

$$D_{\text{rate}} = \frac{1.2v\theta_{az-CP}n}{\lambda} = \frac{1.2v\theta_{az-CP}t_{si}f_s}{\lambda} \tag{34}$$

This shows that the data rate is proportional to the platform velocity, azimuth beam-width, sampling interval, and sampling frequency, and inversely proportional to the radar wavelength.

III. CP-SAR Raw Data Processor

This section presents the development of CP-SAR processing system for the JX-2 UAV. First, we present the algorithm called the Range Doppler algorithm (RDA) that is used for the SAR image formation, followed by an overview of the mobile heterogeneous platform and the implementation of the CP-SAR processing system on this platform.

A. Range-Doppler Algorithm

The Range-Doppler algorithm (RDA) is one of the oldest and the most commonly used algorithm for processing SAR data. The first successful application of this algorithm is for processing SEASAT-the first satellite-borne SAR mission-data in 1976. RDA is computationally efficient and, for typical spaceborne imaging geometries and it is an accurate approximation to the exact SAR transfer function. Thus, the algorithm is phase-preserving and Single Look Complex (SLC) images formed with RDA can be used for advanced applications such as interferometry [29].

To achieve block processing efficiency, RDA uses frequency domain operations in range and azimuth directions, while maintaining the simplicity of one-dimensional operations. It takes advantage of the approximate separability of processing range and azimuth directions, allowed large difference in time scales of the range and azimuth data, and by the use of range migration correction (RCMC) between the two one-dimensional operations in order to compensate the migration effect.

The RCMC operation is done in the range time and azimuth frequency domain to also gain block processing efficiency. This domain is known as "range Doppler" domain, because azimuth frequency is the same with Doppler frequency. The algorithm is called range Doppler algorithm because the RCMC, as the unique feature of this algorithm, is executed in this domain.

A key feature of this algorithm lies in the fact that the processing of energy from point targets, at the same range but different azimuth, is transformed to the same location in azimuth frequency domain. Therefore, correction of one target trajectory in this domain resulted in an effectively corrected family of target trajectories that have the same slant range of closest approach.

To achieve efficiency in implementation, the matched filter convolution process in range and azimuth directions are implemented as multiplication in frequency domain. RDA able to accommodate range variant matched filtering and RCMC operation with relative ease, and all operations are performed in one dimensional data arrays to achieve simplicity without losing efficiency.

Currently there are several variants of RDA available. For example, in order to handle data with a moderate amount of squint, in 1984, NASA added a secondary range compression (SRC) step before RCMC is done. Using SRC, the algorithm is able to compensate the range and azimuth coupling of target's phase history and helps to remove phase distortions in case of SAR with squinted or large aperture datasets.

Figure 9 gives a block diagram of the basic RD algorithm which suited for processing data from relatively small squint angle and aperture lengths antenna.

The following are the general processing steps of RD algorithm:

1. Range compression is performed with fast convolution when the data are in the azimuth time domain. In other words, a range FFT is performed followed by a range matched filter multiply, and finally a range IFFT, to complete the range compression.
2. An azimuth FFT transform the data into range Doppler domain, where Doppler centroid estimation and most of the subsequent operations are performed.

3. RCMC, which is range time and azimuth frequency dependent, is performed in the range Doppler Domain, where a family of target trajectories at the same range are transformed into one single trajectories so that they now run parallel to the azimuth frequency axis.
4. Azimuth matched filtering can be conveniently performed as a frequency domain matched filter multiply at each range gate.
5. The final step in azimuth IFFT to transform the data back to the time domain, resulting in a compressed image. Detection and look summation can be done at this stage, if desired.

Data received from the radar system are referred to as signal data or raw data. The raw data are first demodulated to baseband, so that the nominal center frequency is zero. The demodulated radar signal $s_0(\tau,\eta)$, received from a point target can be modelled as (35).

$$s_0(\tau,\eta) = A_0 \cdot \omega_r\left(\gamma - \frac{2R(\eta)}{c}\right) \cdot \omega_\sigma \cdot (\eta - \eta_c) \cdot \exp\left(-\frac{j4\pi f_0 R(\eta)}{c}\right) \cdot \exp\left(-j4\pi K_r\left(\gamma - \frac{2R(\eta)}{c}\right)^2\right) \quad (35)$$

where A_o, γ, η_c, η, $\omega_\sigma(\tau)$, $\omega_\sigma(\eta)$, f_o, Kγ, R(η) are an arbitrary complex constant, range time, azimuth time reference to closest approach, beam center offset time, range envelope (a rectangular function), azimuth envelope (a sinc-squared function), radar center frequency, range chirp FM rate, and instantaneous slant range, respectively.

The two omega terms model the magnitudes of the range and azimuth signals, and are often neglected in the signal analysis. The instantaneous slant range $R(\eta)$, is given by:

$$R(\eta) = \sqrt{R_0^2 + V_0^2\eta^2} \quad (36)$$

where R_0 is the slant range closest approach.

For this target, the azimuth time η is referenced to zero Doppler. When multiple targets are considered, a common absolute time η, is needed, such as $\eta = 0$ at the start of data acquisition.

In range compression, range FFT is performed followed by a range matched filter multiply. Range IFFT is then performed to the data after the range matched filter multiply, where the range matched filter phase $G(f_\tau)$ is defined as

$$\left.\begin{aligned} s_1(\tau,\eta) &= IFFT_\tau\{FFT\{s_0(\tau,\eta)G(f_\tau)\} \\ G(f_\tau) &= \exp\{-j\pi K_\tau \tau^2\} \end{aligned}\right\} \quad (37)$$

Second step of RDA is azimuth FFT. In this stage, azimuth FFT is performed on the data, which transforms the data into range Doppler domain (τ, f_η).

$$s_2(\tau, f_\eta) = FFT\{s_1(\tau,\eta)\} \quad (38)$$

Third step of RDA is RCMC which performed by a range interpolation operation in the range Doppler domain. In RCMC, an interpolation is implemented based on the sinc function. The data are corrected by the amount given by RCM, which is defined in

$$s_3(\tau, f_\eta) = RCMC\{s_2(\tau, f_\eta)\} \quad (39)$$

$$\Delta R(f_\eta) = \frac{\lambda^2 R_0 f_\eta^2}{8V_t^2} \quad (40)$$

Azimuth Compression is the fourth step of RDA. In azimuth compression, azimuth matched filtering is performed on the data at each range gate in range Doppler domain. Azimuth matched filter phase $H_{az}(f_\eta)$ is as defined in

$$s_4(\tau, f_\eta) = s_3(\tau, f_\eta) \cdot H_{az}(f_\eta) \quad (41)$$

$$H_{az}(f_\eta) = exp\left(-j \cdot \pi \cdot \frac{f_\eta^2}{K_a}\right) \quad (42)$$

$$K_a \approx \frac{2V_\tau^2}{\lambda \cdot R_0} \quad (43)$$

$$s_5(\tau,\eta) = FFT\{s_4(\tau, f_\eta)\} \quad (44)$$

Compare to other algorithms, RDA is relatively simple and can be adapted to most SAR processing tasks with the best trade-offs between accuracy and complexity. Many find it the easiest to understand and implement although its accuracy is not the highest. RDA can also be easily implemented in an efficient pipeline architecture where all operations are done in one dimension at a time. However, in terms of beam-width and squint angle, RDA has restrictive limitations. RDA also has high computation load as the interpolator used for RCMC is a time consuming operation [11].

B. RDA for LFM-CW SAR

Actual raw data from the JX-2 CP-SAR is not yet available because the system is currently under development. In order to test the functionality of the raw data processing system, sample raw data, processed images, and Matlab scripts from microASAR mission courtesy of David G. Long at Brigham Young University is used [21][22]. The microASAR uses linear frequency modulated continuous wave (LFM-CW), which has different signal characteristics compared to our pulse-based SAR, leading to differences in RDA implementations.

There are two major differences between conventional RDA and the LFM-CW version: range compression and range cell migration correction. In LFM-CW radar, range compression can be done by zero padded FFT of each chirp, which is simpler than the convolution method in pulsed radar in (37) of the previous section. In current implementation, due to the low flight geometry, the processing can be simplified by omitting the RCMC steps completely.

In range Doppler processing, to optimize the low altitude data, a hyperbolic azimuth chirp is used instead of parabolic as

the azimuth reference function. The conventional parabolic reference function is defined as:

$$Az_{par}(m) = exp\left\{-j \cdot \left(\frac{f_c \cdot m^2 \cdot v^2}{c \cdot r_a \cdot PRF^2} - \frac{2f_c \cdot v^2 \cdot m}{c^2 \cdot PRF}\right)\right\} \quad (45)$$

where $m = [-N_p/2 \ldots N_p/2]$ is the pulse index, N_p is the number of pulses, fc is the carrier frequency, c is the speed of light, PRF is the pulse repetition frequency, and v is the along track velocity.

The hyperbolic version of this reference function can be expressed as:

$$Az_{hyp}(m) = exp\{-j \cdot (2\pi \cdot f_c \cdot \tau(m) + K_r \cdot \tau^2(m))\} \quad (46)$$

where K_r is the frequency chirp rate and $\tau(m)$is

$$\tau(m) = 2\sqrt{r_a^2 + \left(\frac{mv}{PRF}\right)^2} \quad (47)$$

and mv/PRF is the azimuth displacement.

Figure 9 shows the range processing flow of CASIE SAR data. Before range compression, the raw data must be preprocessed to interpolate gaps between individual transmitted signal in order to create a uniform sampled data. Preprocessing also involved the process of zeroing the first 30 samples to avoid artifacts induced by transient RF signals caused by the switching of the transmitter and antenna side lobe suppression using Taylor window [22]. After zero padding of each received signal to the value of N=4096, with N is the sample number, the range compression is carried out by FFT operation of each sample. The result must be truncated in half to remove the negative frequency portion from the range compressed data. The range compressed data then ready for further processing in azimuth direction depicted in **Figure 10**.

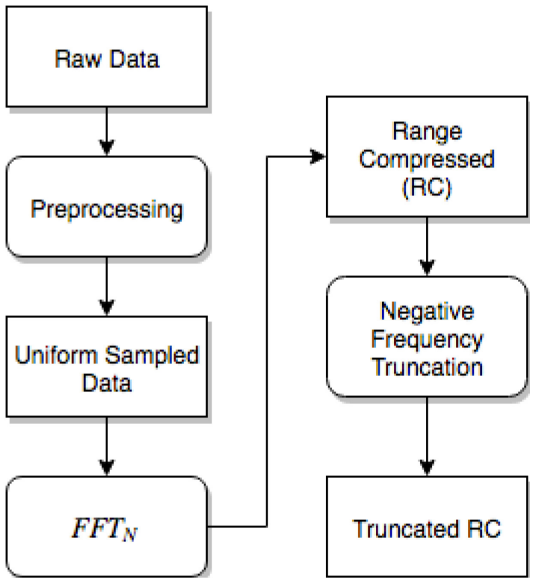


Figure 9 Range Processing Steps

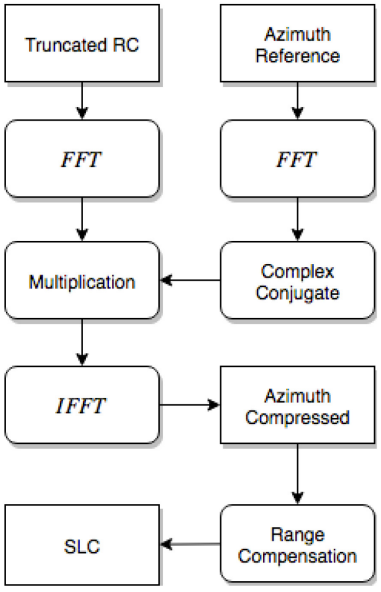


Figure 10 Azimuth Processing Steps

The processing in azimuth direction consists of azimuth compression implemented using multiplication in frequency domain between each azimuth samples with range dependent azimuth reference function in **(46)**. An additional range compensation step by scaling the pixel value with the cubic of range distance is added to approximately compensate for range roll-off effect in the image.

C. Mobile Heterogeneous Computing Architecture

Heterogeneous computing refers to the use of different computing processing cores to maximize performance [32]. Several previous researches focused on the advantages of heterogeneous multiprocessors concerning processing throughput, power consumptions, and Amdahl's law mitigation over its homogeneous counterpart [30]-[33]. Nowadays, the heterogeneous computing paradigm has also been applied to mobile computing areas where size and energy consumption become a key factor [34]. Mobile devices such as laptops, smartphones or tablet is equipped with integrated multicore CPU and application-specific accelerators such as GPU and DSP, and became ubiquitously available.

In this research, we explored the feasibility of implementing on-board SAR processor for JX-2 UAV using NVIDIA Jetson TK-1 mobile heterogeneous computing platform hardware. Jetson TK-1 is powered by NVIDIA Tegra K1 mobile processor with quad core 4 plus 1 ARM Cortex processor, 2 GB of RAM, 16 GB of on-board storage and various peripherals and IO ports (**Figure 11**). It has the 5"-by-5" dimension, weighs 120 gr (excluding the power supply unit) and is equipped with USB 3.0, Gigabit Ethernet and PCIe ×1 port.

The platform runs Linux for Tegra (L4T) operating system derived from Ubuntu 14.04 distribution, including Common Unified Device Architecture (CUDA) 6.5 Toolkit for software development, OpenGL 4.4 drivers and NVIDIA integrated development software on Windows and Linux/Unix Platform.

The GPU architecture of Tegra K1 GPU (codenamed Kepler) is similar with the desktop/server Kepler GPU architecture with additional optimization for mobile usage. The Jetson-TK1 platform with Tegra K1 SoC includes a 192 cores GPU with theoretical performance of 300 GFLOPS and 15 Watts maximum power consumption [35].

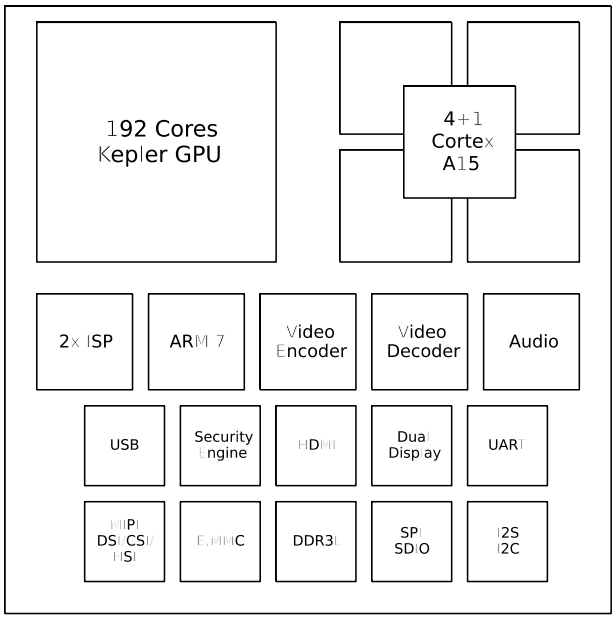


Figure 11 Jetson-TK1 Architecture

Compared to other embedded solution such as FPGA, Custom ASICS and DSP processor, Jetson TK1 offers easier programming model using standard C/C++ with additional CUDA extension. It comes with free tool suite including editor, debugger, and profiler to support the embedded software development. Until recently, this platform has been used for various embedded applications such as robotics, augmented reality, computational photography, human-computer interface, and advance driver assistance systems [35].

D. Software Implementations

We have implemented two version of raw data processing using RDA to run on the mobile heterogeneous platform (**TABLE 3**). The first implementation called RDA-1, was written using GNU Octave scripts. The second implementation RDA-2, was written in C++ programming language and ArrayFire library with NVIDIA CUDA SDK. This implementation uses the GPU to accelerate processing.

The processing code for the data set that we use in this work was created using Matlab programming language; we used a numerical software called GNU Octave to replace Matlab as the reference implementation platform. GNU Octave is selected because it provides mathematical and signal processing function required by our SAR processing routine.

Another major consideration is because it provides extensive graphical plotting capabilities for data visualization and manipulation. GNU Octave is also free and open source, making it a low-cost solution for Matlab replacement. Not to mention that Octave supports most Matlab programming language constructs, making most Matlab program portable with Octave. Due to this high compatibility between Octave with Matlab instructions, only minor modifications of existing Matlab code is required during this process.

For the GPU software development process, we utilize the open source ArrayFire programming library for CUDA. ArrayFire is a software library that enables a software developer to create data-parallel programs in various programming language. It provides high-level matrix abstraction to allow rapid development of software on top of low-level GPU APIs such as CUDA, OpenCL, and OpenGL [36].

TABLE 3 JX-2 AND MICROASAR COMPARISON

Parameter	JX-2 SAR	MicroAsar
Frequency	1270 MHz	5428.76 MHz
Velocity	100 km/h	36 – 540 km/h
Altitude	1 – 4 km	1.5 – 3 km
Bandwidth	150 MHz	80 – 200 MHz
PRF	1 kHz	7 – 14 kHz
Swath width	1000 m	300-2500 m

The actual raw data that we use is from microASAR UAV sample dataset [21][22], which has been chosen to test the RDA implementation due to its relatively similar flight geometry with JX-2 SAR.

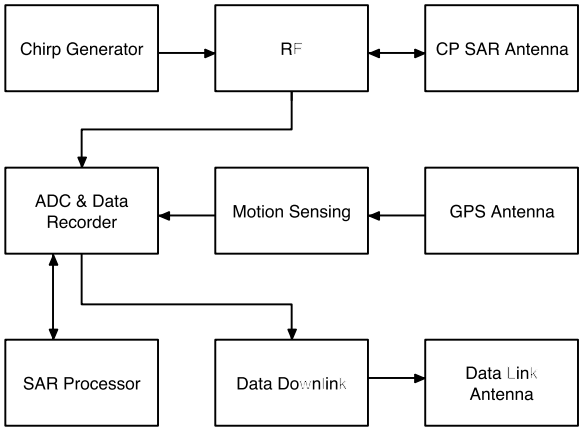


FIGURE 2 Block diagram of CP-SAR System on JX-2

TABLE 2 lists the comparison between of CASIE and JX-2 SAR parameters.

The original raw data is formatted in 64 bit double precision and stored in Matlab's .mat formatted file. The phase histories data consist of 3884 range lines, with 1702 samples for each line representing 13 seconds of SAR data acquisition. To reduce memory usage during processing and maximize the

number operation per second, we have implemented the software using 32 bit single precision floating point number. This due to the capability of Tegra K1 GPU deliver performance up to 300 GFLOPS for single precision and 13 GFLOPS for 64 bit double precision.

The process of reading input signal data and writing output image is handled using MatIO library. This library simplifies the loading and saving matrix, array and numbers into Matlab compatible .mat file. Using this library enables us to interchange data between Matlab/Octave and C++ programs, and allows us to visualize processing result rapidly using Matlab/Octave graphics plotting functions.

IV. Experimental Results and Discussion

Table 4 shows the specifications of the experimental setup. To test the software implementations, we use Jetson TK1 mobile heterogeneous development platform, a small (5×5 in^2) development board for embedded and mobile applications (**Figure 12**). It is powered by the Tegra K1, a mobile processor featuring a CUDA-capable GPU. It has a GK20A Kepler GPU with 192 cores and a quad-core ARM Cortex-A15 32bit CPU. It has 2GB of system RAM and runs Ubuntu Linux version 14.4.

Table 4 Software Implementations

	RDA-1	RDA-2
Language	Octave	C++
CPU Core	1	1
Use GPU	No	Yes
Compiler	-	GNU G++
Matio library	No	Yes
ArrayFire	No	Yes
CUDA SDK	No	Yes
Data size	Float (32 bit)	Float (32 bit)

Table 5 Evaluation Hardware Specification

Parameter	Value
Name	Jetson TK-1
CPU	4+1 Cores
CPU Architecture	ARM Cortex-15
GPU	NVIDIA Tegra K1
GPU Architecture	Kepler
No. of GPU Cores	192
GPU memory size	2 GB
Compute capability	3.2
Ubuntu Linux version	14.04 (32 bit)
Kernel version	3.10.40-gdacac96

Figure 12 shows the two dimensional view of the raw data.

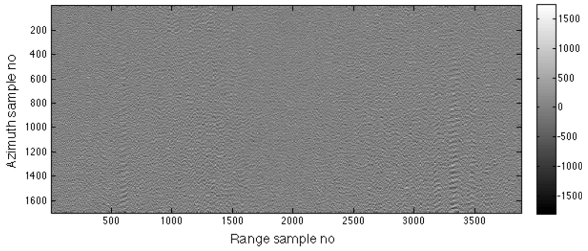
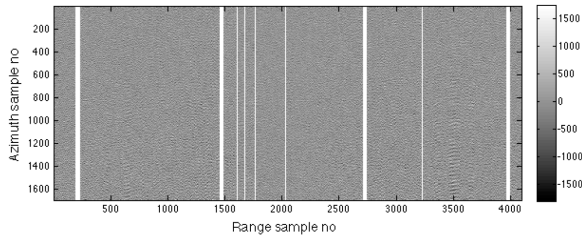
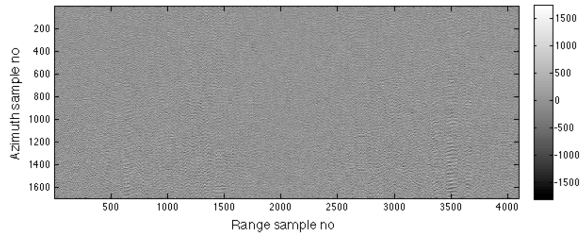


Figure 12 Raw SAR Data

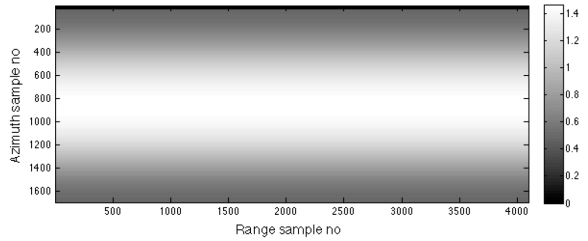
As described in **Figure 9**, before range compression using zero padded FFT, a preprocessing step is required to make the data uniformly sampled, removing RF noise, and suppressing antenna side lobe using Taylor window. The raw data with removed noise and suppressed side lobe is depicted in **Figure 13**.



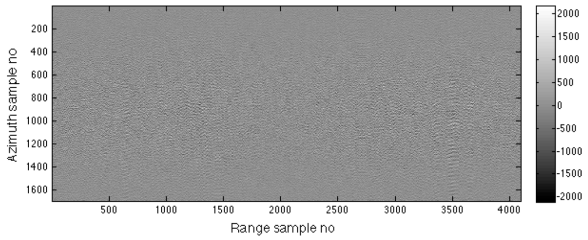
(a) Missing data identification, white color indicating the missing range sample value



(b) After interpolation



(c) Noise removal and side lobe suppression window (Taylor window)



(d) After RF noise removal and side lobe suppression

Figure 13 Preprocessed Data

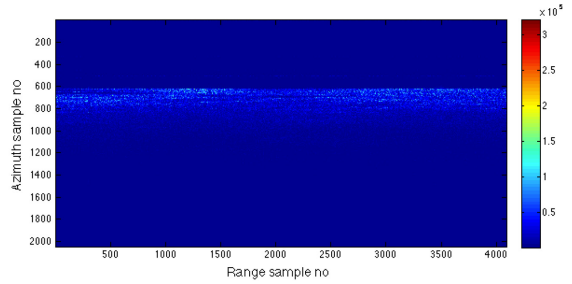


Figure 14 Range compressed data after zero padded FFT and Truncation (absolute value)

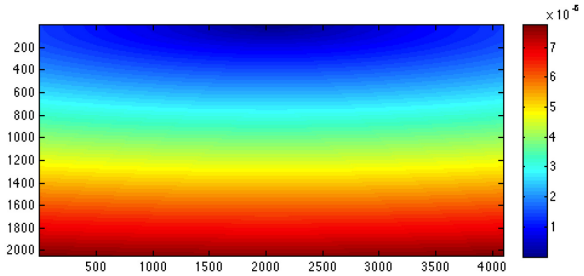


Figure 15 The azimuth time value (τ) calculated from range and azimuth displacement (Equation 47)

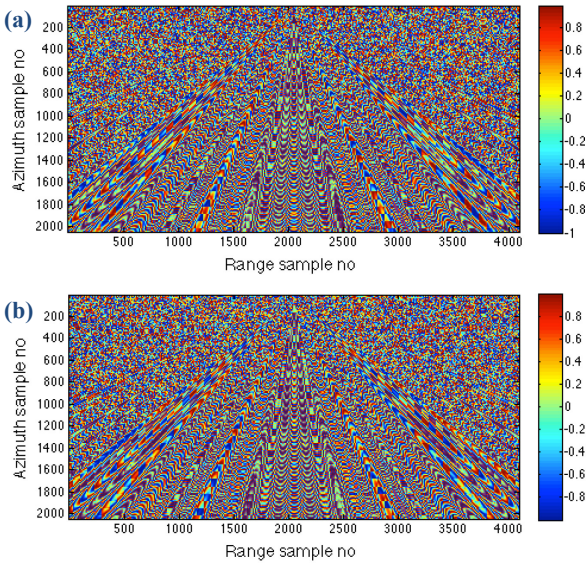


Figure 16 (a) Azimuth chirp filter (real), and (b) Azimuth chirp filter (imaginary used for azimuth compression

After zero padded FFT in the range direction with the number of sample equal to 4096, we have the range compressed data in complex value. After zero-padding each chirp to 4096 values and computing the FFT, which resulting a 4096-by-4096 elements matrix, the redundant negative frequency half of the range compressed data can be discarded by truncation operation creating a 2048×4096 complex single valued matrix (**Figure 14**).

Azimuth processing in RDA is carried out by creating an azimuth filter from parabolic azimuth chirp according to **(46)** with azimuth slow time value from **(47)**. After convolution between the range compressed data (**Figure 14**) with azimuth filter (**Figure 15**), the azimuth compressed data is obtained (**Figure 17**). The convolution operation is carried out by multiplication between range compressed data and complex conjugate of azimuth filter in frequency domain involving pair of FFT and IFFT operation.

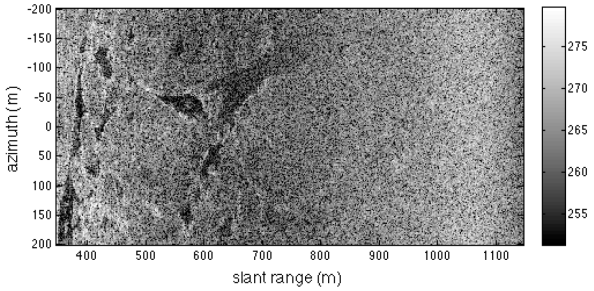


Figure 17 Magnitude value of Single Look Complex (SLC) image from sample dataset

TABLE 6 shows the runtime execution result of two implementation of RDA, RDA-1 the Octave version and RDA-2 the C++ version. In general, RDA-2 shows shorter execution latency except for the following steps: Load Data, Interpolate Missing Data and Calculate Azimuth Chirp. Our initial analysis indicates that there is a possibility of performance degradation of RDA-2 for processing that involves disk read-and-write operation. We also found the possibility of performance improvement in the future by doing detailed runtime profiling of each steps to pinpoint the processing bottleneck.

Although the RDA-2 implementation shows significant speed-up compared to RDA-1, using currently available data, we could not make direct comparison between the two implementation. This is due to the major difference in the programming implementation between Octave, which is an interpreted language, and C++, which is compiled programming language. In order to make fair comparison and to show valid speed up value, we propose further RDA implementation using the same C++ programming language running on CPU and the one running on the CPU and GPU.

TABLE 6 RUNTIME EXECUTION RESULTS

	RDA-1 (sec)	RDA-2 (sec)
Load Data	0.453367	5.324360
Compute Geometry	0.004375	0.012857
Interpolate Missing Data	5.491740	0.299552
Range Compression	1.155760	0.043489
Calculate Azimuth Chirp	12.42320	6.076900
Azimuth Compression	8.867780	1.184680
Total	28.396222	12.941838

Based on runtime execution measurement result, RDA-2 version is able to process raw SAR data approximately in 12.94 seconds. This total processing time value is slightly under the SAR sensor data production time which is 13 seconds.

Although it showed the potential of this platform for near real-time processing of SAR data on-board the UAV platform, further investigation is required in order to optimize the processing time especially those related to data loading and the calculation of azimuth chirp.

V. Conclusion

This work presents the development of raw data processing system for airborne Circularly Polarized Synthetic Aperture Radar on-board JX-2 Unmanned Aerial Vehicle using mobile heterogeneous computing platform. It employs Range Doppler algorithm and parallel processing enabled by GPU on the mobile computing platform to accelerate the raw data processing.

The CP-SAR specific parameter for determining resolutions and data generations has been introduced. Based on CP SAR geometry and axial ratio we derived the image resolution and raw data generation rate, followed by the calculation of sampling requirements which play important role in SAR processing.

The Range Doppler algorithm is implemented rapidly on mobile heterogeneous computing platform with CPU-GPU processor using existing programming language, tools and libraries. Experimental result showed its lightweight, small and compact form factor is highly potential for SAR processing on-board JX-2 UAV. The raw data processing on mobile heterogeneous computing platform reliably generate single look complex image (SLC) from actual UAV SAR mission and feasible for future JX-2 CP-SAR raw data processing.

References

[1] M. Skolnik, *Radar Handbook, Third Edition*, Third. McGraw-Hill Professional, 2008.

[2] A. Moreira, P. Prats-iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 1, pp. 6–43, 2013. https://doi.org/10.1109/MGRS.2013.2248301

[3] J. T. Sri Sumantyo, "Circularly Polarized Synthetic Aperture Radar Onboard Unmanned Aerial Vehicle (CP-SAR UAV)," in *Autonomous Control Systems and Vehicles*, K. Nonami, M. Kartidjo, K.-J. Yoon, and A. Budiyono, Eds. Springer Japan, 2013, pp. 175–192.

[4] V. C. Koo, Y. K. Chan, V. Gobi, M. Y. Chua, C. H. Lim, C.-S. Lim, C. C. Thum, T. S. Lim, Z. bin Ahmad, K. A. Mahmood, M. H. Bin Shahid, C. Y. Ang, W. Q. Tan, P. N. Tan, K. S. Yee, W. G. Cheaw, H. S. Boey, A. L. Choo, and B. C. Sew, "A New Unmanned Aerial Vehicle Synthetic Aperture Radar For Environmental Monitoring," *Prog. Electromagn. Res.*, vol. 122, pp. 245–268, 2012. https://doi.org/10.2528/PIER11092604

[5] M. Edrich, "Ultra-lightweight synthetic aperture radar based on a 35 GHz FMCW sensor concept and online raw data transmission," *IEE Proc. - Radar, Sonar Navig.*, vol. 153, no. 2, p. 129, 2006. https://doi.org/10.1049/ip-rsn:20045080

[6] K. Y. Chan and V. C. Koo, "AN INTRODUCTION TO SYNTHETIC APERTURE RADAR (SAR)," *Prog. Electromagn. Res. B*, vol. 2, pp. 27–60, 2008. https://doi.org/10.2528/PIERB07110101

[7] P. G. Meisl, M. R. Ito, and I. G. Cumming, "Parallel processors for synthetic aperture radar imaging," in *Proceedings of the 1996 ICPP Workshop on Challenges for Parallel Processing*, 1996, pp. 124–131. https://doi.org/10.1109/ICPP.1996.537391

[8] I. G. Cumming and J. R. Bennett, "Digital processing of Seasat SAR data," *Acoust. Speech, Signal Process. IEEE Int. Conf. ICASSP '79.*, pp. 710–718, 1979. https://doi.org/10.1109/icassp.1979.1170630

[9] C. Clemente and J. J. Soraghan, "Range Doppler and chirp scaling processing of synthetic aperture radar data using the fractional Fourier transform," *IET Signal Process.*, vol. 6, no. 5, p. 503, 2012. https://doi.org/10.1049/iet-spr.2011.0354

[10] M. di Bisceglie, M. Di Santo, C. Galdi, R. Lanari, and N. Ranaldo, "Synthetic Aperture Radar Processing with GPGPU," *IEEE Signal Process. Mag.*, vol. 27, no. 2, pp. 69–78, Mar. 2010. https://doi.org/10.1109/MSP.2009.935383

[11] I. G. Cumming and F. H. Wong, *Digital Signal Processing of Synthetic Aperture Radar Data: Algorithms and Implementation (Artech House Remote Sensing Library)*. Artech House, 2004.

[12] Y. C. Lee, V. C. Koo, and Y. K. Chan, "FPGA-based Pre-processing Unit for Real-time Synthetic Aperture Radar (SAR) Imaging," in *Progress In Electromagnetics Research Symposium*, 2012, pp. 1087–1091.

[13] G. F. Panggabean, B. Setiadi, and J. T. S. Sumantyo, "A Single-on-Chip for Onboard SAR Imaging Processor Based on the LEON3," in *The 11th International Conference on Intelligent Unmanned Systems*, 2015.

[14] A. W. Doerry and D. F. Dubbert, "Digital signal processing applications in high-performance synthetic aperture radar processing," in *Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on*, 2003, vol. 1, pp. 947–949 Vol.1. https://doi.org/10.1109/acssc.2003.1292055

[15] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 369–392, 2014.

[16] P. Li, X. Wang, D. Zhang, and C. Deng, "The Design of Miniature UHF SAR Antenna," in *2006 CIE International Conference on Radar*, 2006, pp. 1–2. https://doi.org/10.1109/ICR.2006.343294

[17] Z. Fang and J. Xia, "A miniature implementation of air-born SAR real-time processing," in *2009 2nd Asian-Pacific Conference on Synthetic Aperture Radar*, 2009, pp. 939–942. https://doi.org/10.1109/apsar.2009.5374308

[18] S. I. Tsunoda, F. Pace, J. Stence, M. Woodring, W. H. Hensley, A. W. Doerry, and B. C. Walker, "Lynx: a high-resolution synthetic aperture radar," in *2000 IEEE Aerospace Conference. Proceedings (Cat. No.00TH8484)*, 2000, vol. 5, pp. 51–58.

[19] A. W. Doerry and D. F. Dubbert, "Digital signal processing applications in high-performance synthetic aperture radar processing," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, 2003, vol. 1, pp. 947–949. https://doi.org/10.1109/acssc.2003.1292055

[20] Y.-C. Wang and K.-T. Cheng, "Energy and Performance Characterization of Mobile Heterogeneous Computing," in *2012 IEEE Workshop on Signal Processing Systems*, 2012, pp. 312–317. https://doi.org/10.1109/SiPS.2012.61

[21] M. Edwards, D. Madsen, C. Stringham, A. Margulis, B. Wicks, and D. G. Long, "microASAR: A Small, Robust LFM-CW SAR for Operation on UAVs and Small Aircraft," in *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, 2008, vol. 5, pp. V – 514–V – 517. https://doi.org/10.1109/igarss.2008.4780142

[22] C. Stringham and D. G. Long, "Improved processing of the casie SAR data," in *2011 IEEE International Geoscience and Remote*

Sensing Symposium, 2011, pp. 1389–1392. https://doi.org/10.1109/IGARSS.2011.6049325

[23] Y. Yohandri, V. Wissan, I. Firmansyah, J. . Sri Sumantyo, H. Kuze, P. Rizki Akbar, and J. T. Sri Sumantyo, "Development of Circularly Polarized Array Antenna for Synthetic Aperture Radar Sensor Installed on UAV," *Prog. Electromagn. Res. C*, vol. 19, no. January, pp. 119–133, 2011. https://doi.org/10.2528/PIERC10121708

[24] K. Suto, J. T. Sri Sumantyo, C. W. Guey, and K. V. Chet, "FPGA Based Multiple Preset Chirp Pulse Generator For Synthetic Aperture Radar Onboard Unmanned Aerial Vehicle System," in *The 20th CEReS International Symposium, Symposium on Microsatellite for Remote Sensing 2013*, 2013, pp. 2–3.

[25] W. L. Stutzman, *Polarization in electromagnetic systems*. Artech House, 1993.

[26] P. R. R. Akbar, J. T. T. Sri Sumantyo, V. C. C. Koo, and H. Kuze, "Estimation of Data Memory Capacity for Circularly Polarized Synthetic Aperture Radar Onboard Unmanned Aerial Vehicle (CPSAR-UAV)," *J. Remote Sens. Earth Sci.*, vol. 7, pp. 24–35, 2011.

[27] F. M. Henderson, *Principles and Applications of Imaging Radar (Manual of Remote Sensing, Volume 2)*. Wiley, 1998.

[28] P. Rizki Akbar, J. Tetuko S. S., and H. Kuze, "A novel circularly polarized synthetic aperture radar (CP-SAR) system onboard a spaceborne platform," *Int. J. Remote Sens.*, vol. 31, no. 4, pp. 1053–1060, Apr. 2010. https://doi.org/10.1080/01431160903156528

[29] J. Bennett and I. Cumming, "A Digital Processor for the Production of Seasat Synthetic Aperture Radar Imagery," *LARS Symp.*, 1979.

[30] R. Kumar, D. M. Tullsen, N. P. Jouppi, and P. Ranganathan, "Heterogeneous chip multiprocessors," *Computer (Long. Beach. Calif).*, vol. 38, no. 11, pp. 32–38, Nov. 2005. https://doi.org/10.1109/mc.2005.379

[31] V. Petrucci, O. Loques, D. Mossé, R. Melhem, N. A. Gazala, and S. Gobriel, "Energy-Efficient Thread Assignment Optimization for Heterogeneous Multicore Systems," *ACM Trans. Embed. Comput. Syst.*, vol. 14, no. 1, pp. 1–26, 2015. https://doi.org/10.1145/2566618

[32] M. A. Watkins and D. H. Albonesi, "ReMAP: A reconfigurable heterogeneous multicore architecture," in *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*, 2010, pp. 497–508. https://doi.org/10.1109/micro.2010.15

[33] Z. Zhong, V. Rychkov, and A. Lastovetsky, "Data Partitioning on Heterogeneous Multicore Platforms," *2011 IEEE Int. Conf. Clust. Comput.*, pp. 580–584, 2011. https://doi.org/10.1109/cluster.2011.64

[34] Y.-C. Wang and K.-T. (Tim) Cheng, "Energy and Performance Characterization of Mobile Heterogeneous Computing," *2012 IEEE Work. Signal Process. Syst.*, pp. 312–317, 2012.

[35] NVIDIA, "Data Sheet NVIDIA Tegra K1 Series Processors with Kepler Mobile GPU for Embedded Applications," 2015.

[36] J. Malcolm, P. Yalamanchili, C. McClanahan, V. Venugopalakrishnan, K. Patel, and J. Melonakos, "ArrayFire: a GPU acceleration platform," in *Proc. SPIE 8403, Modeling and Simulation for Defense Systems and Applications VII*, 2012, p. 84030A–84030A–8.