





**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ**

**YAPAY AÇIKLIKLI RADAR GÖRÜNTÜLEME ALGORİTMALARININ  
İTERPOLASYON KULLANMAKSIZIN  
GPU ÜZERİNDE GERÇEKLENMESİ**

**YÜKSEK LİSANS TEZİ**

**Özgür ALTUN**

**İletişim Sistemleri Anabilim Dalı**

**Uydu Haberleşmesi ve Uzaktan Algılama Programı**

**HAZİRAN 2013**



**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ**

**YAPAY AÇIKLIKLI RADAR GÖRÜNTÜLEME ALGORİTMALARININ  
İTERPOLASYON KULLANMAKSIZIN  
GPU ÜZERİNDE GERÇEKLENMESİ**

**YÜKSEK LİSANS TEZİ**

**Özgür ALTUN  
(705101007)**

**İletişim Sistemleri Anabilim Dalı**

**Uydu Haberleşmesi ve Uzaktan Algılama Programı**

**Tez Danışmanı: Prof. Dr. Selçuk PAKER**

**HAZİRAN 2013**



İTÜ, Bilişim Enstitüsü'nün 705101007 numaralı Yüksek Lisans Öğrencisi **Özgür ALTUN**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **“YAPAY AÇIKLIKLI RADAR GÖRÜNTÜLEME ALGORİTMALARININ İNTERPOLASYON KULLANMAKSIZIN GPU ÜZERİNDE GERÇEKLENMESİ”** başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

**Tez Danışmanı :**      **Prof. Dr. Selçuk PAKER** .....  
İstanbul Teknik Üniversitesi

**Jüri Üyeleri :**        **Prof. Dr. Şebnem BAYDERE** .....  
Yeditepe Üniversitesi

**Doc. Dr. Mesut KARTAL** .....  
İstanbul Teknik Üniversitesi

.....

**Teslim Tarihi :**      **3 Mayıs 2013**  
**Savunma Tarihi :**    **6 Haziran 2013**





*Aileme,*



## ÖNSÖZ

Yüksek lisans tezi kapsamında yapmış olduğum çalışma ve uygulama süresi boyunca bana desteğini esirgemeyen, yönlendirmeleri ile tezimin gerçekleşmesinde bana yardımcı olan değerli hocam Prof. Dr. Selçuk PAKER'e teşekkür etmeyi bir borç bilirim. Aynı zamanda tezimin gerçekleşmesi esnasında fikirleri ve yardımlarıyla beni yönlendiren değerli hocam Doc. Dr. Mesut Kartal'a, Yeditepe Üniversitesi'nde birlikte çalıştığım ve bana araştırma kültürü aşılayan, araştırma yapmam konusunda beni yönlendiren değerli hocam Prof. Dr. Şebnem Baydere'ye ayrı ayrı teşekkür etmeyi bir borç bilirim.

Çalışmalarım boyunca beni manevi olarak destekleyen, hep yanımda olan, yoğun çalışma tempomda bana güç veren aileme ve arkadaşlarıma sonsuz teşekkürlerimi iletirim.

Haziran 2013

Özgür ALTUN  
Mühendis



## İÇİNDEKİLER

### Sayfa

ÖNSÖZ .....	vii
İÇİNDEKİLER .....	ix
KISALTMALAR.....	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET .....	xvii
SUMMARY .....	xix
<b>1. GİRİŞ.....</b>	<b>1</b>
1.1 Tezin Amacı ve Kapsamı.....	2
1.2 Literatür Araştırması .....	4
<b>2. SAR SİSTEMİ VE GÖRÜNTÜLEME ALGORİTMALARI .....</b>	<b>7</b>
2.1 SAR Teorisi .....	7
2.2 SAR Görüntüsünün Çözünürlüğüne Etki Eden Faktörler .....	9
2.3 SAR Çalışma Modları .....	12
2.3.1 Şerit taramalı (Stripmap) SAR .....	12
2.3.2 Spot aydınlatmalı (Spotlight) SAR.....	12
2.3.3 Ters SAR (Inverse SAR – ISAR) .....	13
2.4 SAR Görüntüleme Algoritmaları .....	14
2.4.1 Menzil-Doppler algoritması .....	15
2.4.2 Bilgisayar destekli tomografi yöntemi .....	17
<b>3. SAR ALGORİTMALARINDA GPU MİMARİSİNİN KULLANILMASI ..</b>	<b>21</b>
3.1 GPU İşlem Mimarisi.....	22
3.2 GPU Programlama Mimarisi.....	23
<b>4. UYGULAMA .....</b>	<b>27</b>
4.1 En Yakın $k$ -Komşu İnterpolasyonu.....	27
4.2 Paralel Ayrık Fourier Dönüşümü.....	29
4.3 İnterpolasyon Kullanmaksızın RDA Gerçekleme .....	31
4.4 İnterpolasyon Kullanmaksızın CAT Gerçekleme .....	34
4.5 Uygulama Sonuçları .....	37
<b>5. SONUÇ VE ÖNERİLER .....</b>	<b>41</b>
<b>KAYNAKLAR.....</b>	<b>43</b>
<b>ÖZGEÇMİŞ .....</b>	<b>45</b>



## **KISALTMALAR**

<b>SAR</b>	: Synthetic Aperture Radar - Yapay Açıklıklı Radar
<b>ISAR</b>	: Inverse Synthetic Aperture Radar - Ters Yapay Açıklıklı Radar
<b>DSP</b>	: Digital Signal Processor - Dijital Sinyal İşleyicileri
<b>FPGA</b>	: Field Programmable Gate Array - Sahada Programlanabilir Kapı Dizileri
<b>RAR</b>	: Real Aperture Radar - Gerçek Açıklıklı Radar
<b>CPU</b>	: Central Processing Unit - Merkezi İşlem Birimi
<b>GPU</b>	: Graphical Processing Unit - Grafik İşlem Birimi
<b>DFT</b>	: Discrete Fourier Transform - Ayrık Fourier Dönüşümü
<b>FFT</b>	: Fast Fourier Transform - Hızlı Fourier Dönüşümü
<b>IFFT</b>	: Inverse Fast Fourier Transform - Ters Hızlı Fourier Dönüşümü
<b>P-DFT</b>	: Parallel Discrete Fourier Transform
<b>RADAR</b>	: Radio Detection and Ranging
<b>RDA</b>	: Range-Doppler Algoritma
<b>CAT</b>	: Computer Aided Tomography
<b>BCA</b>	: Bluestein Chirp Algorithm
<b>RCMC</b>	: Range Cell Migration Correction
<b>RCM</b>	: Range Cell Migration
<b>SNR</b>	: Signal to Noise Ratio
<b>PRF</b>	: Pulse Repetition Frequency
<b>OpenMP</b>	: Open Mutli-Processing





## ÇİZELGE LİSTESİ

	<u>Sayfa</u>
<b>Çizelge 4.1:</b> Test ortamı.....	37
<b>Çizelge 4.2:</b> RCMC üzerinde interpolasyon ve P-DFT performansı.....	38
<b>Çizelge 4.3:</b> CAT üzerinde interpolasyon ve P-DFT performansı.....	38



## ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 1.1 : SAR görüntüleme geometrisi. ....	2
Şekil 2.1 : Uzaktan algılama sistemleri.....	8
Şekil 2.2 : Range doğrultusu çözünürlük geometrisi. ....	10
Şekil 2.3 : Azimuth doğrultusu çözünürlük geometrisi. ....	12
Şekil 2.4 : Şerit taramalı (Stripmap) SAR modu. ....	13
Şekil 2.5 : Spot aydınlatmalı (Spotlight)SAR modu.....	13
Şekil 2.6 : Ters SAR (ISAR) modu.....	14
Şekil 2.7 : SAR ile hedef üzerinden verinin toplanması.....	15
Şekil 2.8 : Menzil-Doppler algoritması (RDA). ....	16
Şekil 2.9 : Tomografik görüntüleme geometrisi. ....	18
Şekil 2.10 : Bilgisayar destekli tomografi (CAT) yöntemi. ....	19
Şekil 3.1 : CPU ve GPU çekirdek sayıları. ....	22
Şekil 3.2 : Ayrık eş işlemcili hesaplama modeli. ....	23
Şekil 3.3 : Kafes, blok ve kanal yapıları. ....	24
Şekil 3.4 : Çekirdek fonksiyonun sunucu tarafında uyarılması. ....	24
Şekil 3.5 : NVCC derleme süreci.....	25
Şekil 3.6 : GPU üzerinde çalışması için tasarlanmış bir çekirdek fonksiyonu. ...	26
Şekil 4.1 : En yakın $k$ -komşu interpolasyonu. ....	29
Şekil 4.2 : Paralel ayrık fourier dönüşümü. ....	31
Şekil 4.3 : RCMC üzerinde P-DFT gerçekleştirme. ....	32
Şekil 4.4 : Paralel Menzil-Doppler algoritması. ....	34
Şekil 4.5 : CAT üzerinde P-DFT gerçekleştirme. ....	35
Şekil 4.6 : RCMC üzerinde interpolasyon ve P-DFT performansı. ....	38
Şekil 4.7 : CAT üzerinde interpolasyon ve P-DFT performansı. ....	39
Şekil 4.8 : RDA üzerinde elde edilen sonuçlar. ....	39
Şekil 4.9 : CAT algoritması üzerinde elde edilen sonuçlar. ....	39



# YAPAY AÇIKLIKLI RADAR GÖRÜNTÜLEME ALGORİTMALARININ İTERPOLASYON KULLANMAKSIZIN GPU ÜZERİNDE GERÇEKLENMESİ

## ÖZET

Yapay Açıklıklı Radar (Synthetic Aperture Radar – SAR) sistemleri, görüntülenmek istenen bir hedef veya bölge üzerine farklı açılardan geniş bantlı ve yüksek frekanslı darbeler gönderip, elde edilen yansımaların sentezlenmesi ile yüksek çözünürlüklü görüntü elde edilmesi için dizayn edilirler. SAR sistemlerinde görüntünün elde edilebilmesi için gönderilen geniş bantlı darbelerin toplanarak çözümlenmesi ve dijital olarak işlenmeye hazır ham verinin elde edilmesi gerekmektedir. Elde edilen ham veri içerisinde genlik ve faz değerleri bulunan iki boyutlu bir matris olarak ifade edilmektedir. İki boyutla ifade edilen ham verinin boyutlarından biri görüntüleme işlemi sırasında radarın hareket doğrultusu olan çapraz-menzil (azimuth) doğrultusunu, diğeri ise platformun hareket doğrultusuna dik olan menzil (range) doğrultusunu temsil etmektedir. Azimuth ve range doğrultularında elde edilen veriler, elde edilecek görüntünün çözünürlüğünün artırılması hususuna doğrudan etki eden parametrelerdir.

SAR görüntüleme sistemlerinde, radar tarafından toplanan ve içerisinde hedefe gönderilip geri alınan işaretlerin genlik ve faz bilgilerini içeren iki boyutlu ham veri dijital olarak işlenerek görüntüye çevrilirler. Bu amaç doğrultusunda geliştirilen SAR görüntüleme algoritmaları SAR sistemi tarafından sunulan ham görüntü verilerini hem range hem de azimuth ekseninde işleyerek sonuç görüntüsünün oluşmasını sağlarlar.

SAR görüntüleme algoritmaları Fourier dönüşümleri, bir boyutlu veya iki boyutlu filtreleme operasyonları ve interpolasyon adımlarından oluşmaktadır. Interpolasyon sürecinin uygulanması bilinmeyen bir noktanın tahminine dayalı olduğundan uygulama sonucunda belirli bir oranda hata görülmektedir. Buna göre algoritma içerisinde kullanılan interpolasyonun çeşidine göre radar görüntüsünün kalitesi de değişmektedir. SAR görüntüleme algoritmalarında interpolasyon işlemine ihtiyaç duyulmasının iki temel sebebi bulunmaktadır. Bunlardan birincisi kartezyen olmayan noktalarda elde edilen verilerin, kartezyen noktalar üzerine taşınmasını sağlamaktır. Diğeri ise, azimuth ekseninde yapılacak olan işlemlerin range ekseninde yapılan işlemlere bağımlı olmasından dolayı ayrıştırılamaması ve interpolasyon gibi bir dönüşüm sürecine ihtiyaç duyulmasıdır.

Yapılan tez çalışması kapsamında, SAR görüntüleme algoritmalarının uygulanmasından sonra elde edilen görüntünün üzerindeki hata miktarının minimum seviyeye indirilmesi ve doğruluğun artırılması için algoritma akışında yer alan interpolasyon sürecinin kaldırılarak yerine eş değer bir operasyon olarak düzenlenen Paralel Ayrık Fourier Dönüşümü (Parallel Discrete Fourier Transform – P-DFT) operasyonunun uygulanması sağlanmıştır. Ayrık Fourier Dönüşümü'nün paralel olarak işletilmesi Grafik İşlem Birimi (Graphical Processing Unit – GPU) üzerinde gerçekleştirilmiş ve sonuçların Merkezi İşlem Birimi (Central Processing Unit – CPU) üzerinde yapılan paralelleştirme ile karşılaştırılması sağlanmıştır.

Yapılan tez çalışması Range Doppler Algoritması (Range Doppler Algorithm – RDA) içersinde bulunan menzil hücre göçü üzeltmesi adımı üzerinde ve Bilgisayar Destekli Tomografi (Computer Aided Tomography – CAT) algoritmasının işletimi sırasında kutupsal koordinat sisteminden kartezyen sisteme dönüşümün sağlandığı adım üzerinde uygulanmıştır. Yapılan test çalışmalarında RDA ve CAT algoritmaları hem interpolasyon hem de P-DFT kullanılarak gerçekleştirilmiştir. RDA ve CAT algoritmalarının interpolasyon ve P-DFT kullanılarak gerçekleştirilmesi işlemi de hem CPU hem de GPU üzerinde ayrı ayrı olarak gerçekleştirilmiştir. Bu işlemler sonrasında paralel olmayan çözüme göre zaman yönünden avantaj sağlamıştır. Bunun yanısıra interpolasyon uygulanmasının sonucunda elde edilen görüntü üzerine etkiyen bozulmalar açıkça görülmektedir. Bu kapsamda P-DFT'nin hem CPU hem de GPU üzerinde işletilmesi paralel interpolasyon uygulaması sonuçlarına göre hem zaman hem de doğruluk yönünden avantaj sağlamıştır. Ayrıca görüntüde bulunan veri sayısının artmasıyla oluşturulan uygulamanın daha hızlı sonuç üretilmesi bağlamında GPU üzerinde P-DFT uygulaması zaman yönünden büyük avantaj sağlamaktadır.

# **REALIZATION OF INTERPOLATION-FREE SYNTHETIC APERTURE RADAR IMAGING ALGORITHMS BY USING GPU**

## **SUMMARY**

Synthetic Aperture Radar (SAR) is a kind of high-resolution radar which can be applied widely in remote sensing and to acquire information about the Earth's surface from radar signals gathered by a moving platform such as an aircraft. In order to produce high-resolution image in a SAR system, obtained signals must be modulated and raw data should be generated which will be used to produce high-resolution image. The obtained raw data is expressed as a two dimensional matrix which keeps the amplitude and phase values of the signal. One of the dimensions of this matrix is azimuth direction which is parallel direction to the radar flight path while scanning the area or object. The other dimension of this matrix is range direction which is perpendicular to azimuth direction. Both azimuth and range direction are the key parameters that directly affect to increase the resolution of the radar image. Besides of this, this raw data includes complex valued

In the SAR imaging system, the resultant image is acquired by applying digital processing techniques to the SAR raw data. In order to acquire an image, many SAR imaging algorithm are developed by the researchers. These algorithms process the raw data digitally on both azimuth and range direction and produce the resultant data which is interpretable as an image by using special programs. But, because of the complex form of the radar signal and intensive floating-point operations, producing high-resolution requires more computations times.

General SAR imaging algorithm includes Fourier transformation, one and two dimensional filtering operations and interpolation step. Almost all interpolation process results with computation errors because of the pointwise convergence. Besides of this, the 2-D complex valued interpolation also needs extra processing because of ambiguous search operation. For this reason, in order to increase the quality of the resultant image we should use an interpolation which produces minimum computation error while processing the data. There are two main reasons that require applying an interpolation process in the SAR imaging algorithm. One of these reasons is mapping the non- Cartesian coordinated point onto Cartesian coordinated point. The data, which are obtained by radar, don't have equal distance according to each other as both linearly and radially. Therefore, Fast Fourier Transform (FFT) operation cannot be applied to the data in order to perform processing quickly. In order to provide applying FFT operation to the data while processing SAR image algorithms, the data must be expressed in Cartesian coordinated system. For this reason, the data which are expressed on non-Cartesian coordinate system must be mapped to the Cartesian coordinate system. The other one is the operations which are realized on the azimuth axis are depends on the operations which are realized on the range axis. Therefore, these operations cannot be separated and requires a transformation process like an interpolation.

In order to realize the interpolation process for our system, we use nearest n neighbor and weighted nearest n neighbor interpolation method to regulate the position of the points in the radar image. Nearest n neighbor interpolation process is used to determine desired number of nearest points according to selected reference point in the whole data. To do this, all data should be scanned and corresponding points must be determined. After that, average of the point values must be calculated to update the reference point with this calculation result. Similar process must be performed for all chosen data respectively. In weighted interpolation method, weights of the chosen neighbors are determined according to the distance to the reference point and apply to the data in order to update the reference point value with the result of the multiplication of the neighbor's value and its determined weight value. Therefore, mentioned methods takes very long time. On the other hand, the interpolation operation results with calculation errors because of the pointwise convergence.

In this thesis study, in order to minimize the error and increase the accuracy of the resultant image data which is obtained after applying SAR imaging algorithms, we have applied an equivalent operation, which is named Parallel Discrete Fourier Transform (P-DFT), instead of interpolation process. While processing the SAR image algorithm, the Cartesian points places can be calculated analytically and the values on this Cartesian points can be calculated by using Discrete Fourier Transform (DFT) operation. In our work, DFT and Inverse Fast Fourier Transform (IFFT) operation have been applied respectively to get rid of the interpolation process. Actually, Discrete Fourier Transform (DFT) operation is a costly operation and has been applied at the expense of the additional computation cost. Therefore, GPU have been used to parallelize the DFT operation to perform this operation as fast as possible.

The main aim in the SAR processing concept is performing all operation as fast as possible. Generally, real time processing of SAR images can be performed on Digital Signal Processor (DSP) and Field Programmable Gate Array (FPGA) systems. Due to difficulties of programming and debugging processes on DSP and FPGA other programming environment options should be considered. For this reason, applications of SAR processing have been performed on the other processing environments such as CPU and GPU platforms. At this point, parallelization of process provides us very big advantage to accelerate the processing of the algorithm. One of novel technique for SAR image processing is using Graphical Processing Unit (GPU) to obtain the high computation rate.

GPU is the one of the technology which has widely usage area in processing intensive data. In the GPU computing, CPU and GPU are used together in a heterogeneous co-processing computing model. In general, sequential operations are processed on CPU and parallel processing of data and acceleration are processed on GPU. To develop an application on the GPU several programming language have been produced. In our work, The Compute Unified Device Architecture (CUDA) programming interface produced by NVIDIA are used for developing application to perform our proposed algorithm on the GPU. CUDA programming interface is integrated into C programming language and called as CUDA C.

Our study have been applied on Range Cell Migration Correction (RCMC) step of Range Doppler Algorithm (RDA) and the step which is provide transformation from polar coordinate system to Cartesian coordinate system while processing Computer Aided Tomography (CAT) algorithm. In our work, to show the performance of our



system, RDA and CAT have been applied by using interpolation and P-DFT method in parallel. The parallelization of DFT has been performed both on GPU and CPU. Besides of this, we also performed the interpolation operation in parallel both on GPU and CPU. Although the interpolation operation could be realized by using parallel processing, the errors caused by applying interpolation would still remain on the results. Obtained results show us, performing P-DFT on GPU is faster than the interpolation. On the other hand, the obtained results provided by P-DFT are more accurate than interpolation. The results also show us processing on GPU improves the performance of the used algorithm. We have tested used algorithms with different data size. Our results show us, the processing time is growing considerably while the data size is increasing. We can clearly determine that performing P-DFT algorithm on GPU is more suitable in order to process SAR data fast and get more accurate results.



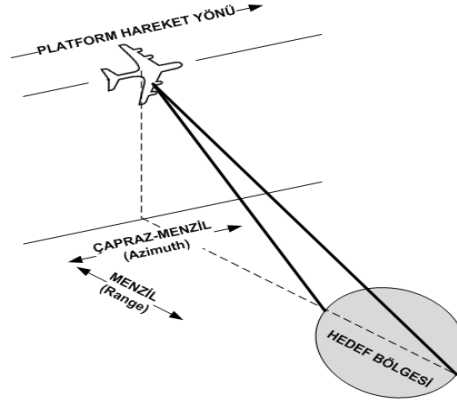
## 1. GİRİŞ

Yapay Açıklıklı Radar (Synthetic Aperture Radar – SAR) 20. yüzyılın ikinci yarısından itibaren kullanılmaya başlanan, yeryüzü hakkında bilgi edinmeyi sağlayan ve uzaktan algılama uygulamalarında da sıklıkla kullanılan yüksek çözünürlüklü görüntü üreten bir radar çeşididir. Temel çalışma prensibi, hareketli bir platform üzerine yerleştirilen radarın görüntülenmek istenen hedef veya alan üzerine gönderdiği geniş bandlı ve yüksek frekanslı darbelerin yansımalarının geri toplanması ilkesine dayanır. Hemen hemen tüm SAR sistemleri hareketli platformdan elde edilen verilerin sentezlenmesiyle yüksek çözünürlüklü görüntü oluşturulması için tasarlanmıştır. SAR sistemleri hem askeri hem de sivil görüntüleme uygulamalarında gözlem, keşif ve izleme amaçlı kullanılmaktadır.

Temel SAR sistemlerinde hareketli platform üzerine yerleştirilen verici ve alıcı, platformun sağladığı hareket ile birlikte yapay açıklık oluşturarak geniş bir bölgenin görüntüsünün farklı bakış açılarından veri toplayarak tarama yoluyla elde edilmesini sağlar. SAR görüntüleme radarları hem durağan bir bölgenin ya da cismin görüntüsünün hem de hareketli bir cismin görüntüsünün elde edilmesi için kullanılabilir. Hareketli bir cismin görüntüsünün elde edilmesi isteniyorsa Ters Yapay Açıklıklı Radar (Inverse Synthetic Aperture Radar – ISAR) kullanımı ile görüntü elde edilebilir. ISAR, bir cisim görüntülemek üzere SAR'da olduğu gibi radarı hareket ettirmek yerine özellikle hedefin iç açısal veya yer değiştirme hareketini kullanır. Bu özellik bize hareketli cisimleri sabit veya hareket eden radarlar ile görüntülemeyi sağlar [1].

SAR sistemlerinde görüntünün elde edilebilmesi için gönderilen geniş bandlı darbelerin toplanarak çözümlenmesi ve dijital olarak işlenmeye hazır ham verinin elde edilmesi gerekmektedir. Elde edilen ham veri içerisinde genlik ve faz değerleri bulunan iki boyutlu bir matris olarak ifade edilmektedir. İki boyutla ifade edilen ham verinin boyutlarından biri görüntüleme işlemi sırasında radarın hareket doğrultusu olan çapraz-menzil (azimuth) doğrultusunu, diğeri ise platformun hareket doğrultusuna dik

olan menzil (range) doğrultusunu temsil etmektedir. Azimuth ve range doğrultularında elde edilen veriler, elde edilecek görüntünün çözünürlüğünün artırılması hususuna doğrudan etki eden parametrelerdir.



**Şekil 1.1:** SAR görüntüleme geometrisi.

Radar görüntüleme algoritmaları, SAR sistemi tarafından sunulan ham görüntü verilerini hem range hem de azimuth doğrultularında işleyerek sonuç görüntüsünün oluşmasını sağlarlar. Radar görüntüleme sistemlerinin gelişmesiyle birlikte elde edilen ham veriden görüntü oluşumunu sağlamak için bir çok algoritma oluşturulmuştur. Bu kapsamda öne çıkan ilk algoritmalarından biri olan 1978 yılında MacDonald Dettwiler ve Jet Propulsion Lab tarafından geliştirilen Menzil-Doppler Algoritması (Range-Doppler Algorithm - RDA) günümüzde de halen kullanılan popüler bir algoritmadır. RDA öne sürüldüğü tarihten itibaren, birçok araştırmacı tarafından geliştirilmiş ve yeni algoritmaların ortaya çıkmasında öncü olmuştur.

### 1.1 Tezin Amacı ve Kapsamı

SAR sistemleri kendi içinde birçok daldan oluşan geniş bir araştırma alanıdır. SAR sistemi araştırma alanlarına örnek olarak SAR sensörlerinin geliştirilmesi veya SAR ile toplanan verilerin işlenmesi verilebilir. Günümüzde birçok araştırmacı radar vasıtasıyla görüntülenen hedef bölgesi üzerinden toplanan verilerin işlenerek ilgili bölgeye ait yüksek çözünürlüklü görüntü oluşturulmasını sağlayan algoritmaların geliştirilmesi üzerine çalışmalar yapmaktadır. SAR algoritmaları konusunda yapılan çalışmalarda üzerinde durulan iki temel kavram bulunmaktadır. Bunlardan biri verilerin gerçek zamanlı olarak işlenmesini, diğeri ise elde edilen verilerden yüksek çözünürlüklü görüntü oluşturmak için yeni algoritmalar geliştirilmesi veya mevcut algoritmaların performanslarının iyileştirilmesi için yapılan çalışmaları kapsamaktadır.

Genellikle, SAR verilerinin gerçek zamanlı işlenmesi Dijital Sinyal İşleyicileri (Digital Signal Processor – DSP), Sahada Programlanabilir Kapı Dizileri (Field Programmable Gate Array – FPGA) vasıtasıyla gerçekleştirilmektedir. Ancak hem DSP hem de FPGA'in programlama ve program üzerindeki hataların tespit edilip düzeltilmesi işlemlerindeki zorluklar SAR verisi işleyen araştırmacıları farklı çalışma ortamları aramaya yöneltmiştir. Bu nedenle son yıllarda günümüz bilgisayarlar üzerinde bulunan Merkezi İşlem Birimi (Central Processing Unit - CPU) ve Grafik İşlem Birimi (Graphical Processing Unit - GPU) üzerinde çalıştırılabilir uygulamalar geliştirilmeye başlanmıştır. Bununla birlikte, SAR verilerinin işlenmesi için CPU ve GPU üzerinde geliştirilen uygulamaların paralel metotlarla desteklenmesiyle yapılan işlemlerin performansı artırılmış ve gerçek zamanlı bir sistem performansına ulaşılması sağlanmıştır.

SAR görüntüleme algoritmaları Fourier dönüşümleri, bir boyutlu (1-B) veya iki boyutlu (2-B) filtreleme operasyonları ve interpolasyon adımlarından oluşmaktadır. Bu tez kapsamında SAR görüntü algoritmalarının içerdiği interpolasyon süreci yerine Paralel Ayrık Fourier Dönüşümü (Parallel Discrete Fourier Transform – P-DFT) yöntemi uygulanmıştır. Yapılan çalışma RDA ve Bilgisayar Destekli Tomografi (Computer Aided Tomography – CAT) algoritmaları üzerinde test edilmiş ve hem hata oranı minimuma indirilmesi hem de hızlı sonuç elde edilmesi sağlanmıştır. SAR görüntüleme algoritmalarında interpolasyon işlemine ihtiyaç duyulmasının iki temel sebebi bulunmaktadır. Bunlardan birincisi kartezyen olmayan noktalarda elde edilen verilerin, kartezyen noktalar üzerine taşınmasını sağlamaktır. Diğeri ise, azimuth ekseninde yapılacak olan işlemlerin range ekseninde yapılan işlemlere bağımlı olmasından dolayı ayrıştırılamaması ve interpolasyon gibi bir dönüşüm sürecine ihtiyaç duyulmasıdır.

SAR görüntüleme algoritmalarının içerisinde bulunan Fourier ve filtre işlemleri paralelleşmeye uygundur. Ancak interpolasyon algoritmalarının hem paralelleştirilmesinin zor olması, hem bir çoğunun bünyesinde belirsiz bir arama süreci içermesi hem de radar ile toplanan verilerin kompleks ve hassasiyeti yüksek ondalıklı sayılardan oluşmasından dolayı görüntü oluşturmak için yapılacak işlemler çok yüksek hesaplama yükü gerektirmektedir. Bunların yanında, interpolasyon algoritmalarının uygulanması bilinmeyen bir noktanın tahmini üzerine dayalı olduğundan, hemen hemen tüm

interpolasyon süreçleri belirli bir oranda hata üretmektedir. Kullanılan interpolasyon çeşidine göre radar görüntüsünün kalitesi de artmaktadır. SAR görüntüsünü oluşturan verinin destek bölgesi bilindiği için bu bölge içerisinde yer alan noktaların değerlerini bulmak üzere Ayrık Fourier Dönüşümü (Discrete Fourier Transform – DFT) yöntemi uygulanabilmektedir. DFT operasyonu, bir noktanın değerinin hesaplanması sırasında, veri içerisinde bulunan diğer tüm noktaların değerlerinin etkilerinin de dikkate alarak hesaplama yapar. Bu durum da hesap edilen noktanın hata değerinin minimuma indirilmesini sağlar.

Bu tez kapsamında, SAR görüntüleme algoritmalarının uygulanmasından sonra elde edilen görüntü üzerindeki hata miktarının minimuma indirilmesi ve doğruluğun artırılması için DFT operasyonunun kullanılan görüntü oluşturma algoritmasına getireceği ek işlem yükü önemsizleştirilmiştir. Bu şekilde algoritma üzerine yüklenen ek işlem yükü ise GPU üzerinde paralelleştirme yaparak telafi edilmiştir. İnterpolasyon süreçlerinin GPU üzerinde paralelleştirilerek uygulanması DFT'ye bir alternatif olarak görünse de, sonuç görüntüsü üzerine etkileyecek interpolasyon hataları devam edecektir. DFT operasyonunun uygulanmasına geçilen Fourier bölgesinden geri dönüşü sağlamak için Ters Hızlı Fourier Dönüşümü (Inverse Fast Fourier Transform – IFFT) kullanılmaktadır. Parallelleştirme işlemlerinin GPU üzerinde yapılmasının sebebi, GPU'nun CPU'ya göre hem daha çok miktarda işlemci birimine sahip olması, hemde matematiksel işlem hızının çok daha yüksek olmasıdır. Bu nedenlerden dolayı GPU paralelleştirme işlemlerinde CPU'ya göre çok ciddi performans artışı sağlamaktadır.

Yapılan tez çalışmasına, destek bölgesi dışında kalan veri değerlerinin tahmini konusu yani ekstrapolasyon işlemi dahil değildir. SAR ve ISAR görüntüleme algoritmalarında kullanılan ekstrapolasyon işlemi başlı başına farklı bir çalışma konusudur ve birçok araştırmacı bu konu üzerinde çalışmalarını devam ettirmektedir.

## **1.2 Literatür Araştırması**

Radar görüntüleme algoritmaları ile sonuç görüntüsü elde edilmesi işlemlerinin hızlandırılması ve aynı zamanda elde edilen sonuç görüntüsünün doğruluğunun artırılması için birçok çalışma yapılmaktadır.

Munson ve diğeri Spot Aydınlatmalı SAR kullanılarak toplanan veri üzerine CAT yönteminin uygulanarak aydınlatılan hedef bölgesinin görüntüsünün oluşturulması üzerine çalışmalar yapmıştır. Yapılan çalışmada genellikle medikal uygulamalarda kullanılan tomografi tekniği ile elde edilen veri ile Spot Aydınlatmalı SAR ile toplanan verinin benzerlikleri gösterilmiş ve bu benzerliklere dayanarak CAT yönteminin Spot Aydınlatmalı SAR verisi üzerinde uygulanması sağlanmıştır. Munson ve diğeri CAT yönteminin SAR verisi üzerine uygulanması için temel olarak izdüşüm-dilim teoremi kullanmıştır. Bu teoreme göre görüntülenmek istenen cismin farklı açılardan elde edilen tüm izdüşümlerinin 1-B'lu Fourier dönüşümleri birleştirilerek cismin 2-B Fourier dönüşümü elde edilmiş olur. Daha sonra izdüşüm-dilim teoremi uygulanarak elde edilen 2-B Fourier dönüşümüne interpolasyon ve IFFT uygulanmasıyla görüntü elde edilir [2].

Wayne Lawton CAT yönteminin Spot Aydınlatmalı SAR üzerinde gerçekleşmesi sırasında uygulanan izdüşüm-dilim teoreminin sonucunda elde edilen ve kutupsal koordinatlarda olan cismin Fourier dönüşümünün noktalar arası uzaklıkları sabit kartezyen koordinatlar üzerine taşımak için Bluestein Çırpı Algoritması (Bluestein Chirp Algorithm – BCA) kullanılmasını sağlamıştır. Genel yöntemler kutupsal koordinatlardan kartezyen koordinatlara dönüş için interpolasyon kullanmaktadır. Wayne Lawton çalışmasında, interpolasyonun yerine BCA kullanarak verilerin kartezyen noktalar üzerindeki gerçek yerlerine yerleştirilmesini sağlar. Uygulanan BCA algoritması  $O(K \log(K))$  işlem karmaşıklığına sahip olup  $K$  ise kartezyen nokta sayısını ifade etmektedir [3].

Lui ve diğeri SAR için GPU tabanlı efektif bir sinyal işleyicisi geliştirilmesi üzerine çalışmışlardır. Yapılan çalışmada RDA algoritmasının tüm adımlarının GPU üzerinde CUDA programlama dili kullanarak paralel olarak gerçekleştirilmesi sağlanarak CPU tabanlı bir SAR işlemcisine göre on sekiz daha hızlı sonuç üretimi sağlamışlardır [4].

Ning ve diğeri çoklu genel amaçlı GPU'lar ile SAR verilerini işleyen bir işlemci dizaynı gerçekleştirilerek RDA algoritmasının tüm adımlarının GPU üzerinde CUDA programlama dili kullanarak işlenmesini sağlamışlardır. Bu çalışmada kullanılan sekiz adet GPU yardımıyla çok yüksek ölçekli verilerin hızlı bir şekilde işlenmesine olanak sağlanmıştır. Gelen yüksek miktarda veri, önce GPU'lar üzerinde işlenmesi için paylaştırılmış, her bir GPU'da kendi üzerine düşen işlem miktarını paralel olarak

yürütüp sonucunu çıkarmıştır. Birden fazla GPU üzerinde oluşan sonuçlar, ortak sistem tarafından toplanarak birleştirilmiştir. Bu sayede sonuçlar dört çekirdekli iki adet CPU üzerinde gerçekleştirilen işlemlere göre 500 kata varan hız arttırımı gerçekleştirilmiştir [5].

Liu ve diğerleri GPU üzerinde doku giydirme metodu kullanarak yeni bir RCMC tekniği üzerinde çalışmış ve 2-B görüntülerin üç boyutlu (3-B) görüntülere dönüştürülmesini sağlamışlardır. Oluşturulan yeni RCMC tekniği üzerinde çalışacak olan interpolasyon operasyonunun getireceği yüksek işlem gücü kullanılan GPU yardımıyla telafi edilmiştir ve hızlı gerçekleştirilen interpolasyon işlemi sayesinde 2-B doku görüntüsü, 3-B bir model üzerine giydirilerek, 3-B görüntünün elde edilmesi sağlanmıştır. Yapılan gerçekleştirme sonucunda GPU ile elde edilen sonucun CPU'ya göre 20 kat daha hızlı olduğu görülmüştür [6].

Altun ve diğerleri Menzil Doppler Algoritmasının (Range-Doppler Algorithm – RDA) içerisinde interpolasyon işlemi barındıran Menzil Hücre Göçü Düzeltme (Range Cell Migration Correction – RCMC) adımının interpolasyon işlemi kaldırılarak Parallel-DFT (P-DFT) ile gerçekleştirilmesi sağlamıştır. Yapılan çalışmayla birlikte, hem interpolasyonun sonuç görüntüsü üzererine getirdiği hataların minimuma indirilmesi sağlanmış hem de sonucun CPU'da gerçekleştirilen interpolasyon işlemine göre yaklaşık yirmi kat daha hızlı sonuç alınmıştır [7].



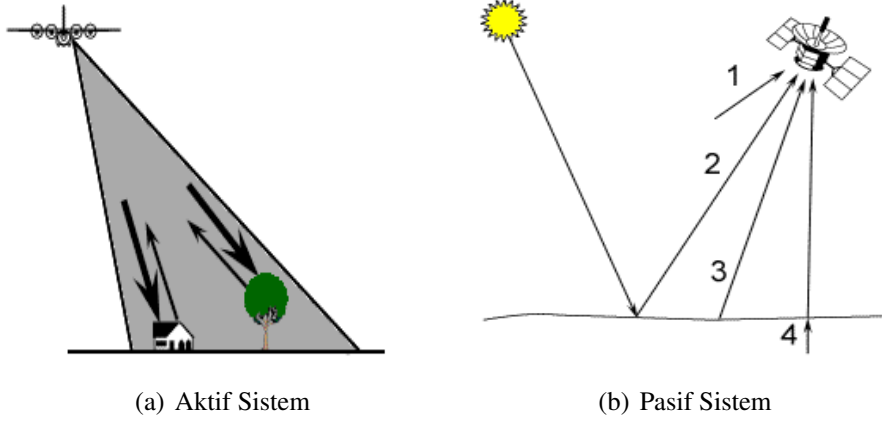
## 2. SAR SİSTEMİ VE GÖRÜNTÜLEME ALGORİTMALARI

Tüm SAR sistemleri, görüntülenmek istenen bir hedef veya bölge üzerine farklı açılardan geniş bantlı ve yüksek frekanslı darbeler gönderip, elde edilen yansımaların sentezlenmesi ile yüksek çözünürlüklü görüntü elde edilmesi için dizayn edilirler [8,9]. Bu bölümde, SAR sistemi teorisinden ve ilgili genel tanımlar,SAR tarafından elde edilen görüntünün çözünürlüğünü etkileyecek faktörler,SAR sistemi modlarından ve görüntü oluşturmak için kullanılan algoritmalar açıklanacaktır.

### 2.1 SAR Teorisi

SAR teorisinin temelleri aslında uzaktan algılamaya dayanmaktadır. Uzaktan algılama, bir cisim veya hedefin noktası hakkında arada hiçbir fiziksel bağlantı ve görüş alanı olmaksızın bilgi alma ve inceleme tekniğidir. Uzaktan algılama sistemleri verici ve alıcıların koordineli çalışmasıyla bilgi toplamayı sağlar. Verici tarafından üretilen elektromaynetik sinyaller hakkında bilgi edinmek istenen hedef noktasına doğru gönderilir. Hedef noktasına çarpıp geri yansıyan sinyaller alıcı yardımıyla toplanıp, uzaktan algılama sistemi tarafından kaydedilir. Kaydedilen verilerin işlenmesiyle hedef hakkında bilgi elde edilir [10].

Uzaktan algılama sistemleri aktif ve pasif olmak üzere iki türlü çalışmayı desteklemektedir. Pasif uzaktan algılama sistemleri cismin aydınlatılması için gerekli enerjiyi güneş gibi dış bir kaynaktan sağlayan sistemlerdir. Herhangi bir dış enerji kaynağı vasıtasıyla aydınlatılan hedef noktası üzerinden yansıyan enerji pasif uzaktan algılama sisteminde bulunan sensörlerle toplanarak sistem bünyesinde kaydedilir ve gerektiğinde işenerek hedef noktası hakkında bilgi elde edilebilmesini sağlarlar. Aktif uzaktan algılama sistemleri hakkında bilgi edinmek istenen hedef noktasının aydınlatılması için gerekli olan enerjiyi kendisinin ürettiği sistemlerdir [11]. Aktif bir uzaktan algılama sisteminin en temel uygulamalarından biri RADAR (Radio Detection And Ranging) sistemleridir [12].



**Şekil 2.1:** Uzaktan algılama sistemleri.

Radarlar, bir hedef noktasının görüntüsünü elde etmek için kullanılan uzaktan algılama sistemleridir. Radarlardan elde edilen görüntünün kalitesini azimuth ve range doğrultularında elde edilen veri miktarı belirlemektedir. Azimuth ve range doğrultusunda elde edilen görüntü çözünürlüklerini etkileyen faktörler ilerleyen kısımlarda açıklanacaktır.

Radar sistemleri kullanılan anten konumlarına göre Gerçek Açıklıklı Radar (Real Aperture Radar – RAR) ve Yapay Açıklıklı Radar (Syntetic Aperture Radar – SAR) olmak üzere ikiye ayrılır.

RAR görüntüleme sisteminde, hareketli platform üzerine hareket yönüne dik olarak yerleştirilen radar dar demetli darbelerle dar bir şeridin görüntüsünün elde edilmesini sağlar. RAR sisteminde, uçuş doğrultusunda yapılan hareket boyunca gönderilen darbelerin sıklığı azimuth çözünürlüğüne, hedef noktası üzerine gönderilen darbelerin bant genişliği ise range çözünürlüğüne etki etmektedir. Range doğrultusunda çözünürlüğü arttırmak için gönderilen darbelerinin bant genişliğinin artırılması, azimuth doğrultusunda çözünürlüğün artırılması için de darbeler arası sürelerin kısaltılması gerekmektedir. Ancak çözünürlüğü arttırmak için her iki doğrultuda yapılacak olan değişiklikler keyfi değerlerle düzenlenememektedir. Çünkü her iki doğrutuda çözünürlüğü arttırmak için yapılacak bu değişiklikler işaret gürültü oranını (Signal to Noise Ratio – SNR) azaltır ve görüntünün bozulmasına yol açar. Bunların yanısıra azimuth doğrultusunda çözünürlüğü arttırmanın bir diğer yolu da RAR’da kullanılan gerçek açıklıklı antenin boyutunun büyütülmesidir. Ancak iyi bir çözünürlük yakalamak için pratikte mümkün olmayan anten boyutlarının uygulanması

gerekmektedir. Bu nedendenlerden dolayı da RAR görüntüleme sistemleri genellikle çözünürlüğün önemli olmadığı uygulamalarda kullanılırlar [10, 13].

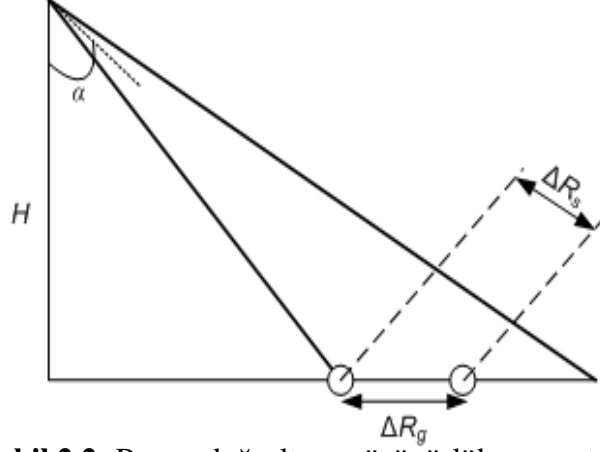
SAR görüntüleme sistemi küçük anten boyutlarıyla yüksek çözünürlüklü görüntü elde etmeyi sağlar. SAR sisteminde, hareketli platform üzerindeki radar anteni, hareket süresince görüntülenmek istenen hedef noktasına belirli aralıklarla ve farklı açı aralıkları ile geniş bantlı darbeler gönderir ve gönderdiği darbelerin yansımalarını eş zamanlı olarak toplar. SAR sistemlerinin bu özelliği sayesinde yapay açıklık oluşturulur ve çok büyük anten boyutlarına gerek kalmadan hem azimuth hem de range doğrultusunda yüksek çözünürlüklü görüntü elde edilebilir [8, 13].

## 2.2 SAR Görüntüsünün Çözünürlüğüne Etki Eden Faktörler

Bir radar görüntüsünün çözünürlüğünde önemli olan birbirine çok yakın olan iki hedef noktasının ayırt edilebilme başarımıdır. Darbe modülasyonlu bir radar sisteminde çözünürlük, darbe süresi ve anten demet genişliğine bağlıdır. Range doğrultusunda yer alan cisimlerin çözünürlüğü darbe süresinin kısa tutulmasıyla artarken, azimuth doğrultusunda ki çözünürlüğü arttırmak için dar demet genişliği gerekmektedir.

Range çözünürlüğü, aynı azimuth doğrultusunda bulunan iki cismin birbirinden ayırt edilebilmesi için aralarında bulunması gereken minimum uzaklıktır. Range doğrultusundaki çözünürlük gönderilen işaretin Darbe Tekrarlama Frekansına (Pulse Repetition Frequency – PRF) bağlıdır. PRF'in azaltılması çözünürlüğün artmasını sağlamaktadır. PRF'in azaltılması gönderilen işaretin band genişliğinin de artmasını sağlar. Bu nedenden dolayı, işaretin çözünürlüğünün artması aynı zamanda bant genişliğinin artmasına bağlıdır. Ancak, PRF'in azalmasıyla gönderilen işaretler arasındaki süre kısalmış ve buna bağlı olarak işaretin enerji seviyesi düşmektedir. Hedef noktasına düşük enerji ile gönderilip geri alınan sinyalin gerekli enerjiyi taşınamaması SNR oranının düşmesine, görüntünün bozulmasına ve hedeflerin seçilememesine yol açmaktadır [8, 13].

Aşağıdaki şekilde aynı azimuth doğrultusunda farklı uzaklıklarda olan iki hedef noktasının çözünürlük geometrisi verilmiştir. Şekilde yer alan  $\Delta R_s$  eğik menzil çözünürlüğünü (slant range resolution) ifade etmektedir ve (2.1) deklemiyle



Şekil 2.2: Range doğrultusu çözünürlük geometrisi.

hesaplanabilmektedir. Bu eşitlikte  $c$  ışık hızını,  $\tau$  ise darbe süresini ifade etmektedir.

$$\Delta R_s = \frac{c \cdot \tau}{2} \quad (2.1)$$

Bunun yanısıra, aynı azimuth doğrultusunda yer alan iki hedef noktası arasındaki minimum uzaklık yani menzil çözünürlüğü (ground range resolution)  $\Delta R_g$  ile ifade edilmektedir ve (2.2) denkleminle hesaplanabilmektedir. Bu eşitlikte kullanılan  $\alpha$  hedefe olan bakış açısını temsil etmektedir.

$$\Delta R_g = \frac{\Delta R_s}{\sin \alpha} = \frac{c \cdot \tau}{2 \cdot \sin \alpha} \quad (2.2)$$

Azimuth çözünürlüğü, uçuş doğrultusuna paralel olan iki hedef noktasının ayırt edilebilmesi için aralarında bulunması gereken minimum uzaklıktır. Radar platformunun hareketi esnasında aynı range doğrultusu üzerinde bulunan iki hedef noktasının birbirinden ayırt edilmesi ancak aynı radar demeti içerisinde bulunmasıyla mümkün olur. RAR sisteminde azimuth çözünürlüğü, range doğrultusunda hedefe olan uzaklık ile hedefin görüntülenmesi için kullanılan demet genişliğine bağlıdır ve (2.3) bağıntısı ile ifade edilir.

$$\Delta R_x = R \cdot \theta_H \quad (2.3)$$

Yukarıdaki bağıntı içerisinde bulunan  $R$  hedefe olan uzaklığı,  $\theta_H$  demet genişliğini ifade etmektedir. Demet genişliği ise kullanılan dalga boyu ( $\lambda$ ) ile anten boyuna ( $L$ ) oranına eşittir. Buna göre RAR'da azimuth çözünürlüğü (2.4) bağıntısı ile ifade edilebilir.

$$\Delta R_x = \frac{\lambda \cdot R}{L} \quad (2.4)$$

SAR’da radar demeti içerisinde kalan bir hedefin azimuth çözünürlüğü Doppler etkisi ile açıklanabilmektedir. SAR için radarın hareketinden kaynaklanan Doppler kayması gönderilen dalganın frekansına ve hareket doğrultusundaki hedef noktasının konumuna bağlıdır ve (2.5) denklemiyle hesaplanmaktadır.

$$f_D = \frac{2 \cdot V_{st} \cdot x}{\lambda \cdot R} \quad (2.5)$$

Yukarıdaki denklem içerisinde yer alan  $V_{st}$  radarın hedefe göre radyal hızını,  $R$  görüş açısı içerisindeki uzaklığı,  $x$  ise duran hedefin hareketli platformla görüntülenmesi sebebiyle azimuth ekseninde oluşan Doppler kaymasına göre hedefin yerini temsil etmektedir. Buna göre ilk hedefin yeri (2.6) denklemiyle bulunabilir. Aynı ifade kullanılarak 2. hedefinde Doppler kaymasına bağlı olan yeri hesaplanabilir.

$$x_{[1,2]} = \frac{f_{D[1,2]} \cdot \lambda \cdot R}{2 \cdot V_{st}} \quad (2.6)$$

Bu ifadeden yola çıkarak iki hedef arasındaki minimum menzil uzaklığı  $\Delta R_x$  (2.7) denklemi kullanarak hesaplanabilir.

$$\Delta R_x = \frac{\lambda \cdot R}{2 \cdot V_{st}} \Delta f_D \quad (2.7)$$

Hedefin radar demeti içerisinde kalma süresi olan  $T$  radarın hareket hızına bağlı olarak (2.8) denklemiyle hesaplanabilir.

$$T = \frac{\theta_H \cdot R}{V_{st}} \quad (2.8)$$

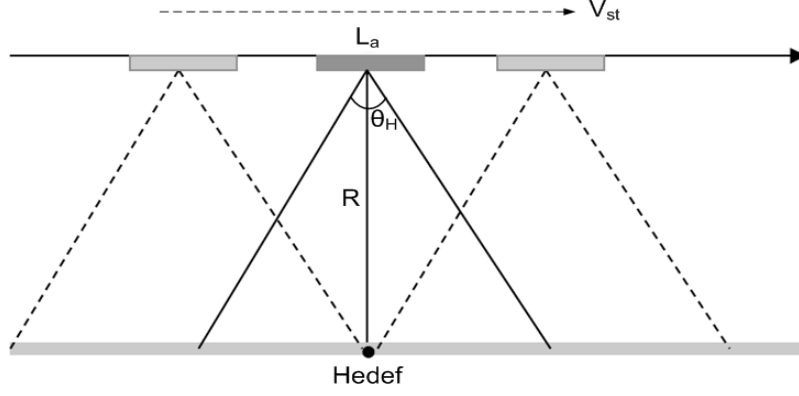
Aynı zamanda hedefin radar demeti içinde kalma süresi Doppler kaymasının tersi olarak (2.9) denkleminde olduğu gibi ifade edilebilir.

$$T = \frac{\lambda \cdot R}{L_a \cdot V_{st}} \quad (2.9)$$

Denklem (2.9) içerisinde yer alan  $L_a$  yapay açıklıklı antenin boyunu ifade etmektedir. Buna göre (2.9) denklemi (2.7) denklemi üzerinde yerine koyulursa (2.10) denklemiyle ifade edilen azimuth çözünürlüğü bağıntısı elde edilir.

$$\Delta R_x = \left( \frac{\lambda \cdot R}{2 \cdot V_{st}} \right) \left( \frac{L_a \cdot V_{st}}{\lambda \cdot R} \right) = \frac{L_a}{2} \quad (2.10)$$

Yukarıdaki denklemde görüldüğü gibi RAR'da azimuth çözünürlüğü menzile bağlı iken SAR'da kullanılan anten boyutuna bağlıdır. SAR sistemlerinde anten boyunun küçülmesi daha iyi çözünürlükte görüntü elde edilmesini sağlar. Diğer bir ifade ile anten boyunun azalması, yapay açıklığı arttırmaktadır. Yapay açıklığın artması da hedef için daha çok veri elde toplanmasını ve çözünürlüğün artmasını sağlar.



Şekil 2.3: Azimuth doğrultusu çözünürlük geometrisi.

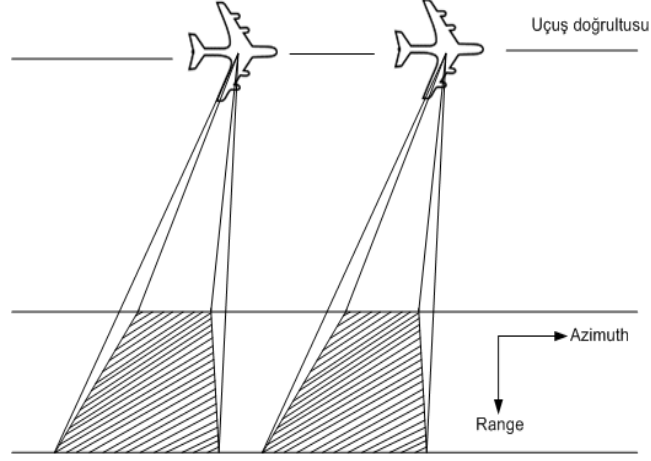
## 2.3 SAR Çalışma Modları

### 2.3.1 Şerit taramalı (Stripmap) SAR

Stripmap modunda çalışan SAR sistemleri, platformun hareket doğrultusu boyunca üzerinden geçtiği hedef noktalarının görüntülenmesini sağlayan sistemlerdir. Sabit bir hızla hareket eden platform üzerine yerleştirilen SAR sistemi, hareket süresinde alıcı ve verici antenlerin açısını değiştirmeden üzerinden geçtiği bölgeyi bir şerit şeklinde tarar. Elde edilen görüntüde görüntünün çözünürlüğü, range doğrultusunda gönderilen darbenin karakteristiğine, azimuth doğrultusunda ise platformun ne kadarlık bir alanın üzerini taradığına bağlıdır. Bu nedenden dolayı, görüntünün range doğrultusundaki çözünürlüğü sabit kalmaktadır. SAR görüntüleme sistemlerinde stripmap modu kullanılarak elde edilen veriler bir hedef noktası hakkında ayrıntılı bilgi veya görüntü almak yerine geniş bir alanda bulunan birçok hedef noktasının görüntülerinin elde edilmesini sağlar [8, 14].

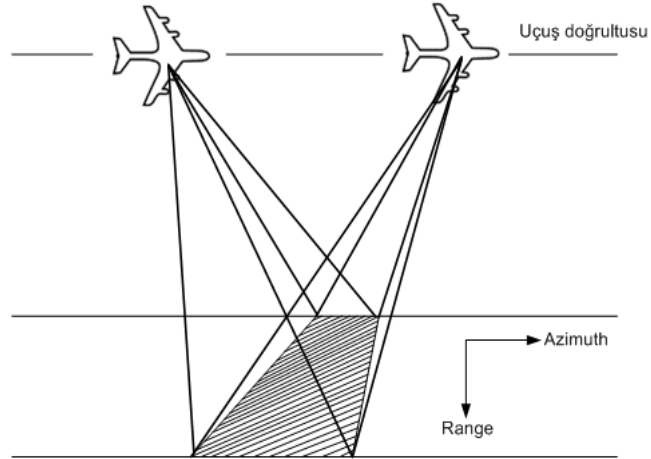
### 2.3.2 Spot aydınlatmalı (Spotlight) SAR

Spotlight modunda çalışan SAR sistemleri, platformun hareketi sırasında belirli bir hedef noktaya odaklanarak, sadece odak noktası üzerinin aydınlatılmasını ve



**Şekil 2.4:** Şerit taramalı (Stripmap) SAR modu.

görüntülenmesini sağlayan sistemlerdir. Bu modda, platformun hareketi sırasında SAR sistemi bünyesinde kullanılan alıcı ve verici antenler, platform hareketine bağımlı olarak değişen bakış açısı değerlerine göre sadece odaklanılan nokta üzerinde görüntüleme yaparlar. Bu modda elde edilen görüntülerin çözünürlüğü, hareketli platformun uçuş süresine doğrudan bağlıdır. Ne kadar fazla aydınlatılma yapılırsa, görüntü hakkında o kadar çok bilgi edinilir ve görüntü çözünürlüğü kalitesi artar [8, 14].

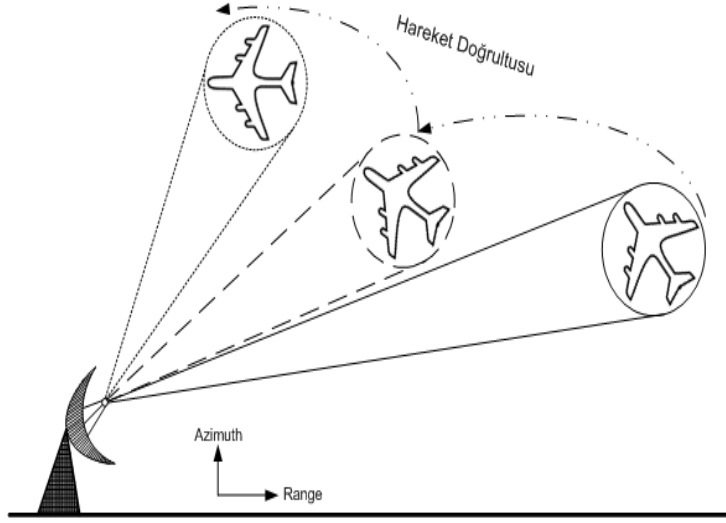


**Şekil 2.5:** Spot aydınlatmalı (Spotlight)SAR modu.

### 2.3.3 Ters SAR (Inverse SAR – ISAR)

ISAR modunda çalışan SAR sistemleri, kullanılan radarın sabit veya hareketli, gözetlenecek hedef noktasının ise hareketli olduğu zamanlarda kullanılır. Bu modda, odaklanılmış hareketli hedef noktası üzerine gönderilen sinyallerin yansımaları ISAR sistemine ait alıcı sensörler tarafından toplanır. ISAR'ın başlangıç noktası hedefin

dönme hareketi sırasında radarla hedef arasında bir koordinasyon gerektirir. ISAR sistemi elde edilen veri bakımından Spotlight SAR ile benzerlik göstermekte ve çoğu zaman aynı algoritmalar ile görüntü elde edilebilmektedir [8, 14].



**Şekil 2.6:** Ters SAR (ISAR) modu.

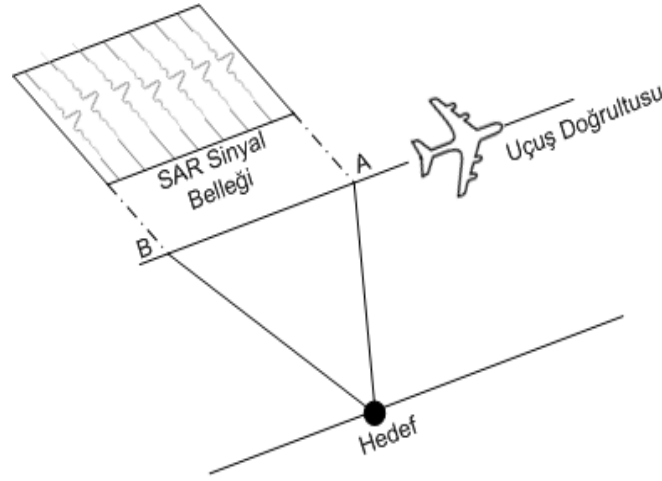
#### 2.4 SAR Görüntüleme Algoritmaları

SAR görüntüleme sistemlerinde, radar tarafından toplanan ve içerisinde hedefe gönderilip geri alınan işaretlerin genlik ve faz bilgilerini içeren iki boyutlu ham veri dijital olarak işlenerek görüntüye çevrilirler. Elde edilen ham verinin görüntüye çevrilmesi için birçok araştırmacı sürekli yeni algoritmalar geliştirmekte veya var olan algoritmaları geliştirmektedirler. Bu bölümde tez kapsamında yaptığımız çalışma bünyesinde kullanmış olduğumuz Menzil-Doppler Algoritması (RDA) ve Bilgisayar Destekli Tomografi (CAT) Algoritması açıklanacaktır.

SAR'da genel amaç hem azimuth hemde range doğrultusunda hedef üzerinden saçılan ve SAR sistemi tarafından toplanan sinyallerin enerjilerinin işlenerek görüntü oluşturulmasıdır. Ancak tek bir hedef noktası için üzerinden saçılan işaretlerin enerjisinin görüntü üzerinde tek bir piksele karşılık düşürülmesi gerekir. Hedef noktası üzerinden saçılan tüm işaretlerin dijital olarak dönüştürmesiye *sinc* formunda sinyaller elde edilir. Bir hedef noktası için elde edilen işaretlerin sayısı Bölüm 2.2'de belirtilen faktörlere bağlıdır. Elde edilen örneklerin fazlalığı, elde edilecek görüntünün de çözünürlüğünün artmasını sağlar.



SAR görüntü işlemede diğer bir problem ise yan lob problemidir. Radar ile saçıcı üzerine gönderilip geri alınan ve dijital olarak işlenmesiyle *sinc* olarak ifade edilen sinyalleri bir bütün olarak toplanmasıyla yan lob problemi ortaya çıkmaktadır. *Sinc* işareti karakteristik olarak bir tepe noktası (ana lob) ve tepe noktasının sağında ve solunda azalarak sonsuza kadar giden işaretlerin bütününden oluşur. Bir *sinc* işareti sonuza doğru uzarken ana lobun sağında ve solunda yer alan yan lobların bir hayli hızlı şekilde sıfır değerine ulaşması istenir. SAR görüntülerinde, görüntülenmek istenen saçıcının yer alacağı pikselde elde edilen *sinc* sinyalinin ana lobu, komşu piksellerinde ise yan lobları yer almaktadır. Yan loblar ne kadar hızlı sıfıra yakınsarsa komşu pikseller üzerinde bulunması muhtemel olan diğer saçıcılar o kadar az etkilenirler ve görüntülerindeki bozulma da o kadar az olur.

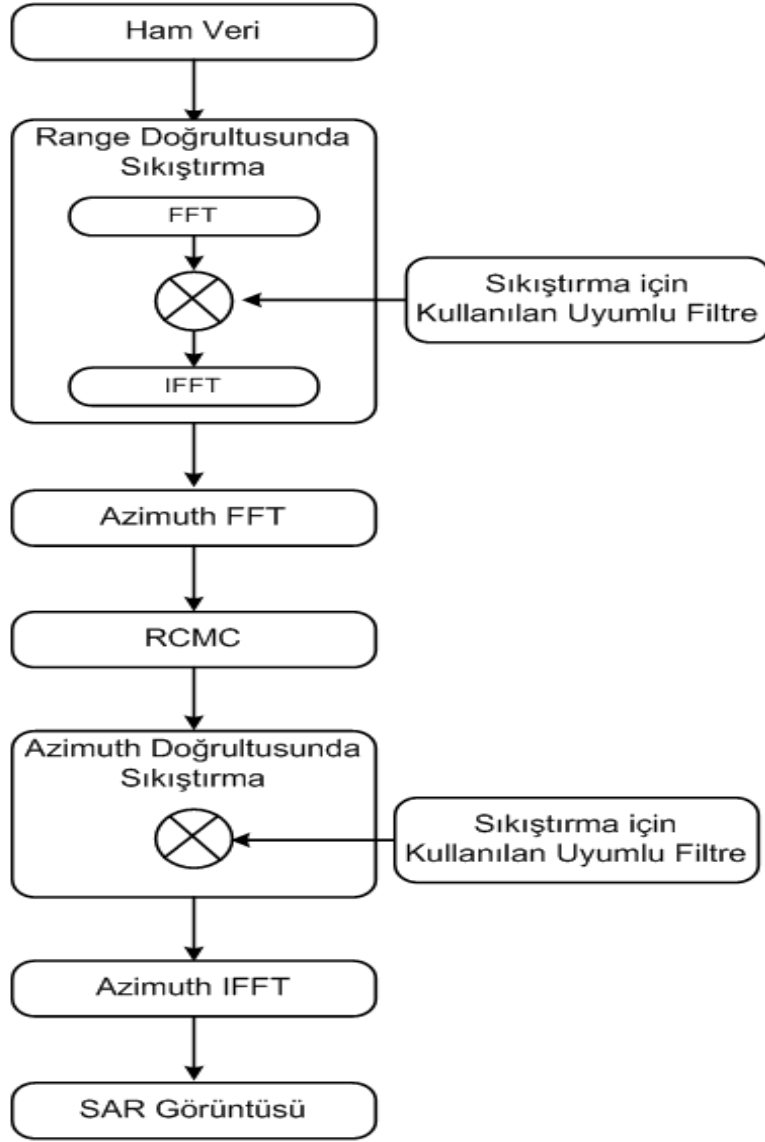


**Şekil 2.7:** SAR ile hedef üzerinden verinin toplanması.

#### 2.4.1 Menzil-Doppler algoritması

Menzil-Doppler algoritması (Range-Doppler Algorithm - RDA), 1978 yılında MacDonald Dettwiler ve Jet Propulsion Lab tarafından geliştirilen ilk SAR görüntü algoritmasıdır. Geliştirilmesinden bu yana uzun yıllar geçmesine rağmen halen sıklıkla kullanılan, birçok araştırma tarafından üzerinde geliştirmeler yapılan ve kendiinden sonra geliştirilen bir çok algoritmaya da öncü olan bir algoritmadır.

RDA algoritması temel olarak Range doğrultusunda sıkıştırma, Range-Doppler bölgesine geçiş, Menzil Hücre Göçü düzeltmesi, Azimuth doğrultusunda sıkıştırma ve zaman bölgesine geçiş adımlarından oluşmaktadır [15].



**Şekil 2.8:** Menzil-Doppler algoritması (RDA).

SAR sistemi tarafından elde edilmiş olan ham veri üzerine öncelikle range doğrultusunda sıkıştırma işlemi uygulanmaktadır. Bu işlem ham verinin, range doğrultusunda Hızlı Fourier Dönüşümü (FFT) uygulanması ve frekans bölgesine geçilmesi ile başlar. Frekans bölgesine alınan ham veri, sıkıştırma işleminde kullanılacak uyumlu filtre ile range doğrultusuna Fourier bölgesinde bir boyutlu çarpma işlemine tabi tutulur ve range doğrultusunda sıkıştırma işlemi gerçekleşir. Sıkıştırılan ham veri Ters Hızlı Fourier Dönüşümü (IFFT) ile tekrar zaman bölgesine alınır.

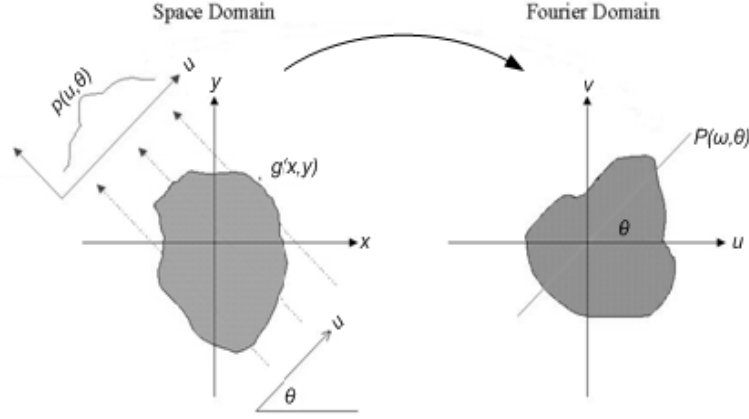
İkinci adımda verinin Range-Doppler bölgesine alınması sağlanır. Bunun için zaman bölgesindeki veri üzerine azimuth boyunca FFT uygulanır. Range-Doppler bölgesine dönülmesinin hemen ardından, algoritmanın en kritik parçası olan Menzil Hücre Göçü

Düzeltilmesi (Range Cell Migration Correction - RCMC) kısmı gerçekleştirilir. SAR platformunun hareketinden dolayı gözlemlenen hedef üzerinden toplanan işaretlerin konumlarında zamana göre bir yer değişim olmaktadır. Bu değişime Menzil Hücre Göçü (Range Cell Migration - RCM) adı verilmektedir. RCM değeri veri üzerindeki Doppler etkisine bağlı olarak değişmektedir. Bu nedenden dolayı hedefin doğru yerlerinin tespit edilebilmesi için, verilerin interpolasyon algoritmaları yardımıyla düzenlenmesi sağlanır. Ancak interpolasyon işlemleri sonucu hesaplanan veri belirli bir oranda hata içermektedir. Bu nedenle RDA algoritmasında uygulanabilecek interpolasyon algoritması için çalışmalar sürekli olarak devam etmekte, geliştirilen yeni algoritmalar ile RDA algoritmasının yeni bir versiyonu oluşturulmaktadır. RDA algoritmasına RCMC adımının uygulanması ve verilerin yerlerinin düzenlenmesinden sonra Range-Doppler bölgesi üzerinde azimuth eksininde sıkıştırma yapılması gerekir. Azimuth ekseninde sıkıştırma yapmak için RCMC sonucu düzeltilmiş veri uyumlu filtre ile azimuth doğrultusunda bir boyutlu çarpma işlemine tabi tutulur. Bu işlem sonucunda sıkıştırılan veri IFFT yardımıyla zaman bölgesine geri taşınır ve ortaya range ve azimuth eksenlerinde alınan verilerin sıkıştırılmış ve düzenlenmiş hali olan iki boyutlu bir matris ortaya çıkar. Bu matrisin görüntü olarak yorumlanması için belirli programlar kullanılır ve hedef noktasının görüntüsü elde edilir.

#### **2.4.2 Bilgisayar destekli tomografi yöntemi**

Bilgisayar destekli tomografi (Computer Aided Tomography - CAT) uzun yılladır medikal uygulamalarda kullanılan, son yıllarda da mikrodalga görüntüleme sistemlerinde kullanılmaya başlanan görüntüleme tekniğidir. Tomografik görüntüleme sistemlerinde görüntülenecek olan nesne ya da hedef noktası için farklı bakış açıları üzerinden gönderilen-alınan işaretlerin toplanıp dijital olarak işlenmesiyle yüksek çözünürlüklü görüntü oluşturulması sağlanır. Genel olarak Spotlight SAR veya ISAR üzerinden elde edilen verilerin görüntüye çevrilmesi için bu yöntem kullanılmaktadır. Medikal tomografik görüntüleme sistemleri ile Spotlight SAR'ın birbirinden bağımsız olarak geliştirilmesine rağmen hem her iki yönteminde görüntü oluşturmak için gönderilen işaretin yansımalarındaki enerjilerin değişimleri ile ilgilenmesi hem de görüntülenen alanların sınırlı olmasından dolayı CAT yöntemi her iki sistem içinde görüntü oluşturmak kullanılabilir [16].

CAT yönteminin uygulanması izdüşüm-dilim teoremi [17, 18] ile açıklanabilir. CAT görüntü oluşturma için temel olarak Fourier dönüşümünü kullanır. CAT, görüntülenmek istenen bir nesne ya da hedef noktasından farklı bakış açıları ile elde edilen izdüşümsel görüntülerine 1-B FFT uygulanmasıyla 2-B kesit görüntülerinin oluşturulması tekniğidir [2, 19].



**Şekil 2.9:** Tomografik görüntüleme geometrisi.

Görüntülenmek istenen hedef bölgesinin 2-B'li fonksiyonu  $g(x,y)$  ile ifade edilsin. Buna göre, hedef bölgesinin belirli bir açıda gönderilen paralel ışınlar yardımıyla elde edilen izdüşümü bir doğru integrali ile (2.11) denkleminde olduğu gibi ifade edilir.

$$p(u, \theta) = \int_{-\infty}^{\infty} g(x,y) dv \quad (2.11)$$

Yukarıdaki denklemde  $\theta$  izdüşümün elde edildiği açı değerini,  $u$  açısal olarak dönme hareketine göre uzamsal bölgesinde merkezden uzaklığa karşı gelen değişkendir. Buna göre, (2.11) denkleminde uzamsal bölgesini temsil eden  $x$  ve  $y$  değerlerinin  $u$  ve  $v$  değişkenleri ile temsil edilebilmesi için (2.12) bağıntısında olduğu gibi bir dönüşüm yapılır.

$$\begin{aligned} x &= u \cos \theta - v \sin \theta \\ y &= u \sin \theta - v \cos \theta \end{aligned} \quad (2.12)$$

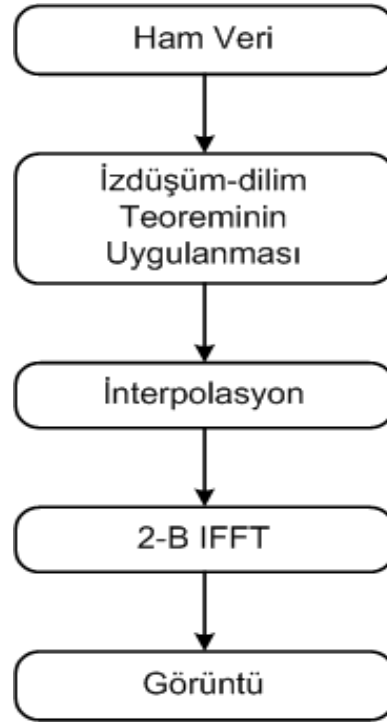
Elde edilen dönüşüme göre (2.11) ifadesi yeniden düzenlenirse (2.13) denklemi elde edilir.

$$p(u, \theta) = \int_{-\infty}^{\infty} g(u \cos \theta, u \sin \theta) dv \quad (2.13)$$

Yukarıdaki denklem ile  $\theta$  açısında elde edilen doğru integralinin 1-B Fourier dönüşümü, hedef bölgesinin 2-B'lu fonksiyonu olan  $g(x,y)$ 'nin 2-B Fourier dönüşümünün  $\theta$  açısındaki bir kesitine eşit olur ve (2.14) denklemiyle ifade edilir.

$$P(\omega, \theta) = \int_{-\infty}^{\infty} p(u, \theta) e^{-ju\omega} du \quad (2.14)$$

Buna göre, cismin farklı açılardan alınan tüm izdüşümlerinin 1-B'lu Fourier dönüşümleri birleştirilerek, hedef bölgesinin 2-B Fourier dönüşümü elde edilmiş olur. İzdüşüm-dilim teoreminin uygulanmasıyla elde edilen veri Fourier bölgesinde kutupsal koordinatlarda ifade edilir. Elde edilen sonucun görüntü olarak yorumlanabilmesi için uzamsal bölge üzerinde ifade edilebilmesi gerekmektedir. Fourier bölgesinden uzamsal bölgeye geri dönmek için 2-B IFFT işlemi uygulanması gerekmektedir. Ancak IFFT işleminin uygulanabilmesi için verinin kartezyen koordinat sistemine taşınması işleminin gerçekleşmesi gerekmektedir. Bu nedenle kutupsal koordinat sisteminden kartezyen koordinat sistemine dönüş için interpolasyon işlemi uygulanmaktadır. İnterpolasyon işleminin sonucundan elde edilen kartezyen koordinat sistemindeki 2-B veriye IFFT uygulanarak uzamsal bölgeye dönülür ve 2-B'lu görüntüye dönüştürülebilir veri elde edilir.



**Şekil 2.10:** Bilgisayar destekli tomografi (CAT) yöntemi.

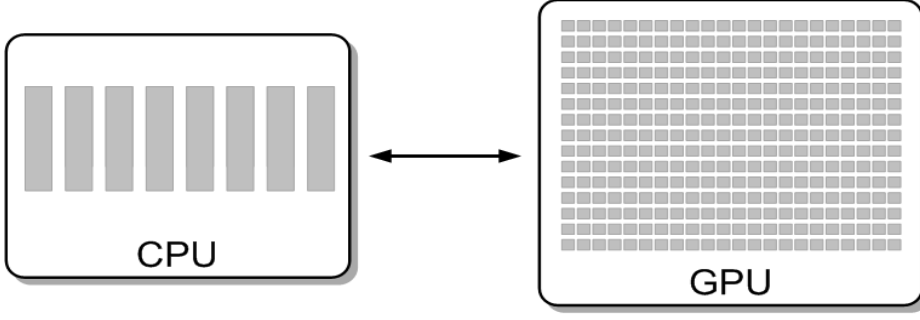
Özet olarak, CAT üç temel adımdan oluşmaktadır. Bunlardan birincisi izdüşüm-dilim teoreminin uygulanmasıyla cismin 2-B Fourier dönüşümünün elde edilmesi, ikincisi kutupsal koordinat sisteminden kartezyen koordinat sistemine geçişi sağlayan interpolasyon işleminin uygulanması, son adım ise 2-B IFFT uygulanmasıyla uzamsal bölgeye dönüş ve görüntü olarak ifade edilebilir bir verinin elde edilmesidir.

### 3. SAR ALGORİTMALARINDA GPU MİMARİSİNİN KULLANILMASI

SAR görüntüleri, içerisinde görüntülenmek istenen hedef bölgesi ile ilgili yüksek miktarda veri barındırırlar. Hedef bölgesi üzerinden toplanan yüksek miktarda verinin belirli algoritmalar ile işlenerek görüntüye çevrilmesi çok miktarda işlem gerektirmekte, dolayısıyla da işlem süresi bakımından çok zaman alabilmektedir. SAR sistemlerinin geliştirilmesi üzerine yapılan araştırmaların temel amaçlarından bir tanesi de verilerin gerçek zamanlı sistem performansında işlenmesini sağlamaktır. Bu bağlamda verilerin gerçek zamanlı olarak işlenmesi üzerine birçok araştırma yapılmış, yeni sistemler tasarlanmış, daha hızlı ve efektif işleyen algoritmalar geliştirilmiştir.

SAR verilerin gerçek zamanlı sistem performansında işlenmesi için kullanılan sistemlerin başında DSP ve FPGA yapıları gelmektedirler. Ancak DSP ve FPGA hem programlama hem de yazılan programlar üzerindeki hataların bulunup düzeltilmesi yönünden SAR sistemi tasarımcısına zor bir ortam sunmaktadır. Bu nedenle son yıllarda CPU veya GPU tabanlı çözümler SAR verilerinin işlenmesi için ön plana çıkmaktadır. Yüksek miktarda veri içeren SAR görüntülerinin CPU veya GPU kullanılarak gerçek zamanlı olarak işlenmesi işlem gerçekleştirme hızı bakımından DSP ve FPGA'e göre nispeten daha uzun zaman almaktadır. Bu nedenle CPU veya GPU kullanılarak oluşturulan sistemlerde işlemlerin paralelleştirilmesi sağlanmış ve paralel olarak çalışmaya uygun algoritmalar ve yöntemler geliştirilmiştir. Paralel işlemlerin gerçekleştirilmesi için GPU tabanlı sistemler CPU'ya göre daha efektif çözümler sunmaktadır. GPU'ların paralel çalışmaya uygun mimarisi SAR görüntülerinin kısa süre içerisinde işlenmesini ve gerçek zamanlı sistem performansının yakalanması konusunda büyük avantajlar sağlamaktadır. CPU ve GPU arasında içerdikleri işlemci çekirdekleri sayısı bakımından ciddi ölçülerde fark vardır. Günümüz teknolojisine göre genel amaçlı kullanılan CPU'lar içerisinde sekiz çekirdek barındırırken GPU'lar yüzlerce çekirdeği içerisinde barındırabilmektedir. Paralel işlemlerin hızlı gerçekleşmesi birden çok çekirdeğin bir arada ve koordineli bir şekilde çalışması ile sağlanmaktadır. Çekirdek sayısının CPU'ya göre çok fazla

olması GPU'lar üzerinde paralel olarak gerçekleştirilen işlemlerin çok daha hızlı tamamlanmasını sağlamaktadır. Bunun yanısıra GPU'lar üzerindeki her bir çekirdeğin CPU üzerindeki çekirdeklere göre matematiksel işlem gerçekleştirme hızı daha yüksektir. GPU'ların hem daha hızlı işlem gerçekleştirilmesi hem de içerdiği yüksek miktarda çekirdek sayesinde daha güçlü paralelleştirme ortamına sahip olması yüksek miktarda verinin işlenmesi esnasında büyük avantaj sağlamaktadır.



Şekil 3.1: CPU ve GPU çekirdek sayıları.

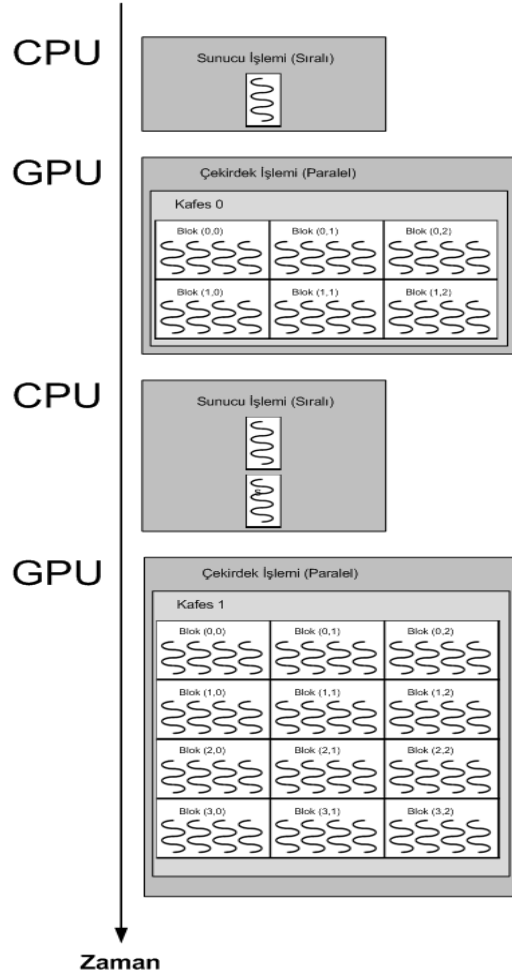
### 3.1 GPU İşlem Mimarisi

Grafik işlem birimi tabanlı olarak tasarlanan programlar *ayrık eş işlemcili hesaplama* adı verilen işletim modelini kullanarak çalışma zamanı içerisinde hem GPU hem de CPU'nun birlikte kullanılmasıyla gerçekleştirilir. Bu model kapsamında, program üzerinde paralel olarak çalıştırılması istenen kısımlar GPU üzerinde, sıralı olarak çalıştırılması istenen kısımlar CPU üzerinde gerçekleştirilir.

GPU programlama mimarisine göre oluşturulan programlar *sunucu(host)* veya *çekirdek(kernel)* fonksiyonları olarak adlandırılırlar. *Ayrık eş işlemcili hesaplama* modeline göre, CPU üzerinde çalıştırılan program parçalarına *sunucu*, GPU üzerinde çalıştırılan programlarına ise *çekirdek* fonksiyonu adı verilir. Kullanılan hesaplama modeline göre GPU üzerinde çalıştırılmak istenen bir *çekirdek* fonksiyonunun CPU üzerinde çalışan bir *sunucu* fonksiyonu tarafından uyarılarak çalıştırılması gerekir. GPU ve CPU'nun kullandığı bellek bölgelerinin birbirinden farklı olduğu için GPU üzerinde işlem yapmaya başlamadan önce işlenecek olan veriler CPU'nun kullandığı bellek bölgesi üzerinden GPU'nun belleğine kopyalanmalıdır. Aynı şekilde GPU üzerinde paralel olarak gerçekleştirilen işlemler sonucu elde edilen sonuçlar yorumlanması veya kaydedilmesi için GPU'nun bellek bölgesi üzerinden CPU'nun belleğine aktarılır. GPU ve CPU arasında bellek aktarımlarının



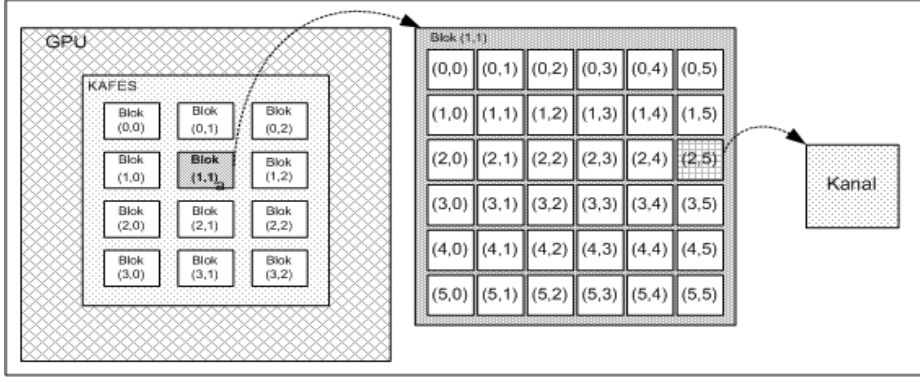
gerçekleştirilmesi için *sunucu* tarafında *cudaMemcpy* fonksiyonunun kullanılması gerekmektedir. Bu fonksiyon hem CPU'dan GPU'ya hem de GPU'dan CPU'ya veri aktarımını gerçekleştirebilmektedir. Hangi yönde veri aktarımı yapılmak istendiği fonksiyona parametre olarak verilen bir değer ile seçilebilmektedir. Buna göre, CPU'dan GPU'ya veri aktarımını sağlamak için fonksiyona parametre olarak *cudaMemcpyHostToDevice*, GPU'dan CPU'ya veri aktarımını sağlamak için ise *cudaMemcpyDeviceToHost* değerinin verilmesi gerekmektedir [20, 21]. Şekil 3.4 üzerinde *cudaMemcpy* fonksiyonunun kullanımıyla ilgili örnek gösterilmektedir.



Şekil 3.2: Ayrık eş işlemcili hesaplama modeli.

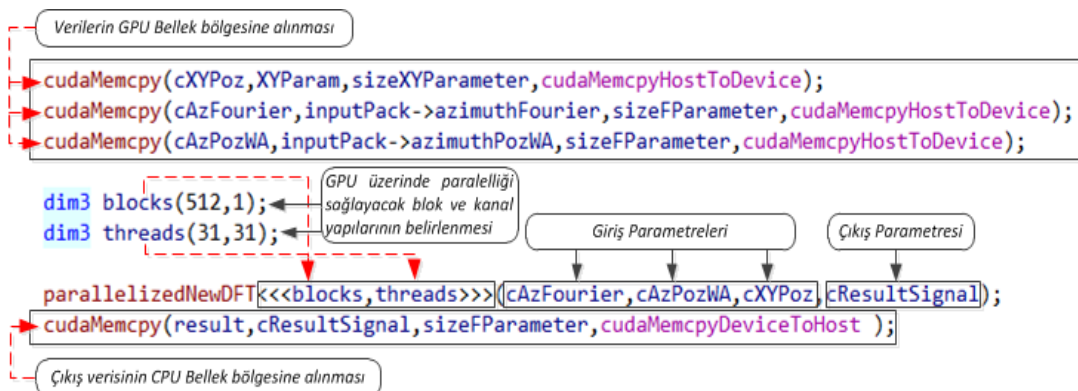
### 3.2 GPU Programlama Mimarisi

GPU üzerinde programlama yapmak için kullanılan en yaygın programlama arayüzlerinden biri NVIDIA tarafından geliştirilmiş olan *CUDA (The Compute Unified Device Architecture)* arayüzüdür. *CUDA* programlama arayüzü C programlama diline entegre edilerek *CUDA C* programlama dili olarak adlandırılmıştır.



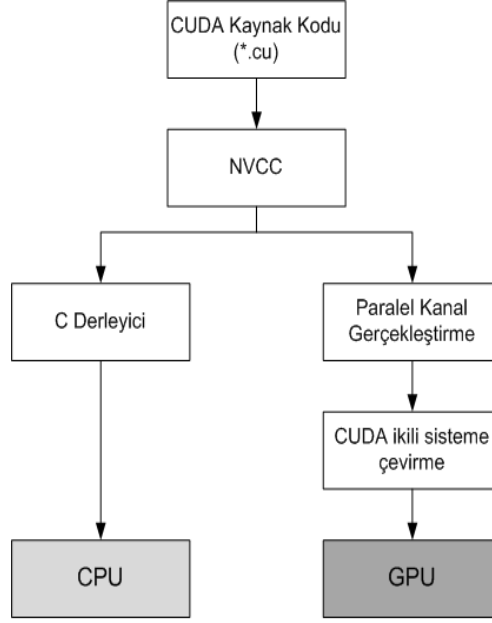
Şekil 3.3: Kafes, blok ve kanal yapıları.

Herhangi bir GPU üzerinde işletilecek programların paralelleştirilmesi için *kafes(grid)*, *blok(block)* ve *kanal(thread)* yapıları kullanılır. GPU üzerinde çalıştırılması istenen her bir program bir *kafes* olarak ifade edilir. *Kafes* içerisinde gerçekleştirilecek tüm işlemler ise *blok* ve *kanal* yapıları kullanılarak paralelleştirilirler. Buna göre *kafes* yapısı içerisinde alınan bir iş belirlenen sayı kadar *blok* üzerinde işlenmesi için bölünür. Her *blok* kendi içinde belirli sayıda *kanal* içermektedir. *Blok* yapıları kendi üzerine atanan işi içerisinde yer alan *kanallar* yardımıyla paralel olarak gerçekleştirirler. *Kafes* yapısı içerisinde kullanılacak *blok* ve *kanal* miktarları GPU üzerinde işlemleri gerçekleştirecek olan *çekirdek* fonksiyonunun CPU üzerinden uyarılması esnasında *sunucu* tarafında belirlenir. Şekil 3.4 üzerinde *sunucu* tarafından çağrılmak istenen bir *çekirdek* fonksiyona ilişkin örnek örnek gösterilmiştir. *Sunucu* üzerinden uyarılan *çekirdek* fonksiyonu *blok* ve *kanal* sayısı parametrelerini alarak gerçekleştirilmek istenen işi parçalara böler ve işletir. Bu kullanım bize toplamda  $blokSayisi \times kanalSayisi$  kadar kanal sağlar ve işlemler bu kanallar kullanılarak paralel olarak yürütülürler [21].



Şekil 3.4: Çekirdek fonksiyonun sunucu tarafında uyarılması.

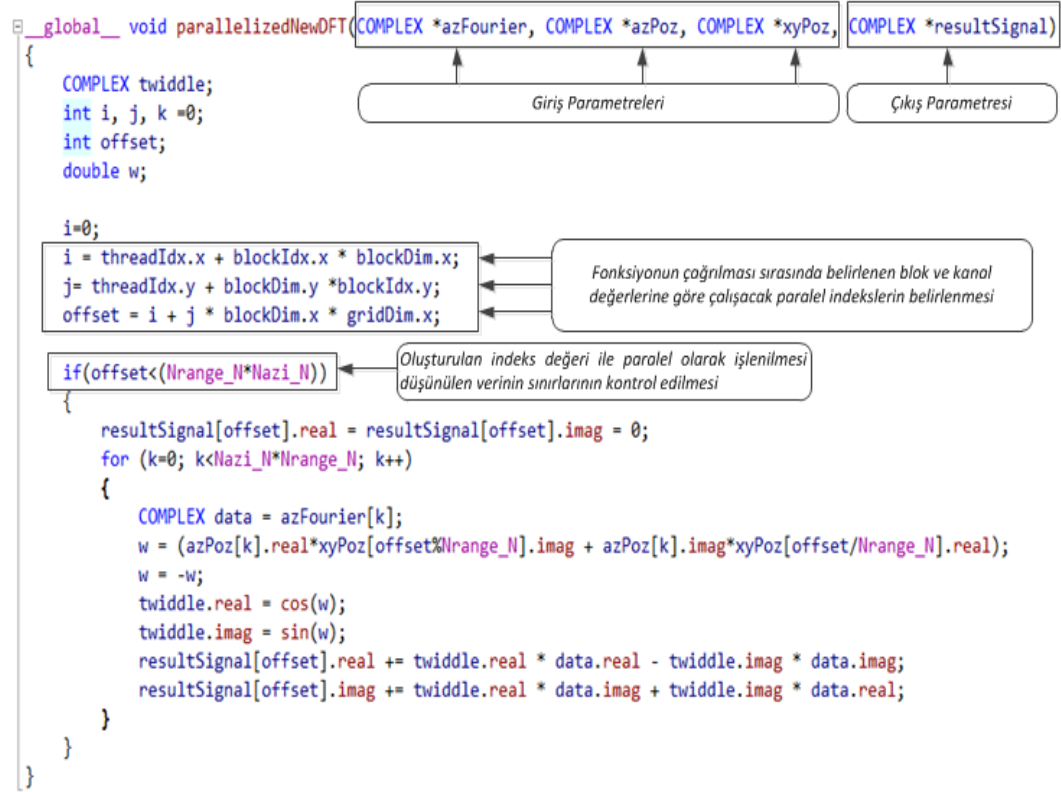
CUDA programlama arayüzü kullanılarak GPU üzerinde gerçekleştirilmesi için yazılan tüm kaynak kodlar ".cu" uzantılı dosyalarda saklanır ve *NVCC (NVIDIA's CUDA Compiler)* derleyicisi kullanılarak derlenir. GPU üzerinde işlem gerçekleştirmek için oluşturulan kaynak kodu dosyası hem *sunucu* hem de *çekirdek* üzerinde işlenecek komutları içerisinde bulundurur. NVCC derleyicisi kaynak kodları içerisinde bulunan fonksiyonları çalışacağı donanıma uygun şekilde ayrıştırır ve buna göre derleme işlemini gerçekleştirir [20].



**Şekil 3.5:** NVCC derleme süreci.

Kaynak kod dosyası içerisinde bulunan ve GPU üzerinde çalışması için yazılan fonksiyonlar tanımlama itibari ile CPU üzerinde çalışması için yazılanlardan farklıdır. GPU üzerinde çalışması için tasarlanmış olan fonksiyonlar başına "*\_\_global\_\_*", "*\_\_device\_\_*", "*\_\_host\_\_*" anahtar kelimeleri getirilerek tanımlanırlar. Buna göre "*\_\_global\_\_*" anahtar kelimesi kullanılarak tanımlanan bir fonksiyon çağırılması *sunucu* tarafından gerçekleştirilen ve GPU üzerinde gerçekleştirilen diğer bir söylemle *çekirdek* üzerinde işleyen fonksiyondur. Bu fonksiyonlar genelde GPU üzerinde gerçekleştirilecek işlemlerin başlatılmasında kullanılır. Bunun yanı sıra "*\_\_device\_\_*" anahtar kelimesi ile tanımlanan bir fonksiyon sadece *çekirdek* fonksiyonları tarafından çağrılabilen ve *çekirdek* üzerinde işleyen bir fonksiyondur. Bunların aksine "*\_\_host\_\_*" anahtar kelimesi ile çağrılan fonksiyonlarda çağırılması *sunucu* tarafından gerçekleştirilen ve *sunucu* üzerinde işleyen bir fonksiyondur. İçerisinde standart C programlama diline ait kaynak kodlar yer alır ve gerçekleştirilir. "*\_\_host\_\_*" anahtar

kelimesi kaynak kod dosyası içerisindeki fonksiyonların tanımının varsayılan halidir. Bu nedenle yazılmasına gerek yoktur [21].



Şekil 3.6: GPU üzerinde çalışması için tasarlanmış bir çekirdek fonksiyonu.

GPU üzerindeki paralelleştirme yukarıda açıklandığı gibi kafes üzerinde yer alan blok ve kanal yapıları ile gerçekleştirilmektedir. GPU üzerinde paralel olarak gerçekleşecek olan bir çekirdek fonksiyonunun kaç blok yapısına bölüneceği ve her bir bloğun kaç kanal kullanılarak gerçekleştirileceği sunucu tarafında çekirdek fonksiyonunun çağırılması için yazılan ifade üzerinde belirlenmektedir. Çekirdek fonksiyonunun çağırılması esnasında belirlenen bu parametreler çekirdek fonksiyonu içerisinde blockIdx ve threadIdx ifadeleri ile kontrol edilebilmektedir. Bir çekirdek fonksiyonu içerisinde paralelleştirmeyi sağlayan blok ve kanal yapıları paralel olarak işlenmesi istenen probleme göre bir boyutlu, iki boyutlu ve üç boyutlu olarak tanımlanabilmektedir. Buna göre çekirdek fonksiyonu içerisinde paralelliği kontrol etmek için kullanılan ifadeler blockIdx.(x/y/z) ve threadIdx.(x/y/z) olarak kullanılarak istenilen boyutun kontrol edilebilmesine olanak tanırırlar.

## 4. UYGULAMA

Bu bölümde tez kapsamında yaptığımız projenin gerçekleşmesine ilişkin detaylardan bahsedilecektir. Yaptığımız tez çalışması ile SAR ve ISAR görüntüleme algoritmalarının uygulanması sırasında kartezyen olmayan noktalardan kartezyen noktalara dönüşümü sağlayan interpolasyon adımının yerine Paralel Ayrık Fourier Dönüşümü (Parallel Discrete Fourier Transformu – P-DFT) uygulanması sağlanmıştır. P-DFT ile interpolasyon işleminin kompanse edilmesi ve interpolasyon işleminin sonuca etki eden hata oranının minimize edilmesi sağlanmıştır. Bunun yanı sıra, genel yapı itibarıyla paralelleştirilmesi zor olan interpolasyon metotlarının yerine, daha kolay paralelleştirilebilen bir metot kullanılmıştır.

Radar görüntüleme algoritmalarında interpolasyona ihtiyaç duyulmasının temel olarak iki sebebi bulunmaktadır. Bunlardan birincisi radar ile elde edilen verilerin kartezyen koordinatlar üzerinde olmamasıdır. Radar ile elde edilen veriler hem açısal, hem de doğrusal olarak eşit aralıklara sahip değildirler. Bu nedenle de verilerin hızlı olarak işlenebilmesi için FFT operasyonu uygulanamamaktadır. Radar algoritmalarının uygulanması sırasında veri üzerine FFT uygulanabilmesini sağlamak için verinin kartezyen koordinatlar ile ifade edilebilmesi gerekmektedir. Bundan ötürü, görüntü radarları ile elde edilen ve kartezyen noktalarda olmayan verilerin, kartezyen noktalar üzerinde taşınması işlemi uygulanır. Diğer sebep ise, azimuth ekseninde yapılacak olan işlemlerin range ekseninde yapılan işlemlere bağımlı olmasından dolayı ayrıştırılamaması ve interpolasyon gibi bir dönüşüm sürecine ihtiyaç duyulmasıdır.

Radar görüntüleme algoritmaları için kullanılmakta olan interpolasyon yöntemlerinin geliştirilmesi başlı başına bir çalışma konusu olarak ele alınmaktadır. Bu tip uygulamalarda en çok kullanılan interpolasyon tipi en yakın  $k$ -komşu interpolasyonudur.

### 4.1 En Yakın $k$ -Komşu İnterpolasyonu

*En yakın  $k$ -komşu interpolasyonu* düzensiz olan verinin içerisinden ele alınan bir referans noktası için en yakın  $k$  adet noktanın belirlenmesini gerektirmektedir.

Bunun için veri içerisindeki tüm noktaların taranması ve belirlenen  $k$  adet noktanın değerlerinin ortalaması hesaplanarak ilgili referans noktasının değerinin güncellenmesi sağlanır. Bu işlem veri içerisindeki tüm noktalara uygulanacağından işlem yükü fazladır. Bunun yanısıra, interpolate edilmek istenen noktanın değerinin sadece kendi değerine en yakın noktaların değerlerinin ortalamasından oluşur. Bundan dolayı, yeni noktanın değerine etkiyen hata oranı fazla olabilir. En yakın  $k$ -komşu interpolasyonu veri içerisinde bulunan noktaların değerlerinin birbirlerine yakın olduğu durumlarda daha iyi sonuç üretmektedir. Ancak noktalar arası değer farklarının fazla olduğu durumlarda güncellenen değer olması gereken değerden farklı çıkabilmekte ve interpolate edilmiş nokta üzerindeki hata oranının artmasına sebep olabilmektedir.

Radar görüntüsü oluşturmak için toplanmış olan veri hem görüntüyü oluşturacak noktaların değerlerini hem de her bir noktanın 2-B düzlemde konum bilgilerini içermektedir. Bu nedenle *en yakın k-komşu interpolasyonunu* radar görüntüleri üzerinde uygulamak için öncelikle noktaların konum bilgilerine göre birbirlerine uzaklık hesabı yapılmalı daha sonra yakın noktaların değerlerinin ortalaması dikkate alınarak interpolate edilmek istenen noktanın değeri güncellenmelidir [22]. İki boyutlu konum uzayı içerisinde yer alan  $p_i$  ve  $p_j$  noktaları arası uzaklık hem  $x$  hemde  $y$  eksenindeki konum bilgileri kullanılarak (4.1) ile ifade edilen *Öklid uzaklık bağıntısı* ile hesaplanabilir.

$$d_n = dist(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad , 1 \leq n \leq k \quad (4.1)$$

Yukarıdaki formüle göre  $p_i$  noktası için konum uzayındaki diğer tüm noktalarla aradaki uzaklık hesaplanır ve bunlardan en yakın  $k$  tanesi belirlenir. Belirlenen bu konumların içerisinde bulunan veri değerlerinin ortalaması alınarak  $p_i$  noktasının değeri güncellenir.

$$f(x_{p_i}, y_{p_i}) = \frac{\sum_{l=1}^k f(x_{p_l}, y_{p_l})}{k} \quad (4.2)$$

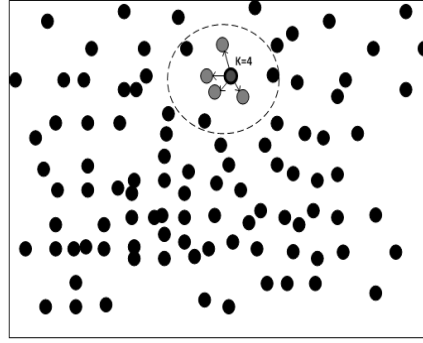
Bir noktanın interpolate edilmesi sırasında elde edilmiş  $k$  adet noktanın her birinin ilgili noktaya etkisi aynı oranda olmayabilir. Böyle bir durumda, hesaba katılacak noktaların interpolate edilecek noktaya eşit oranda katkı vermesi hatalı sonuçlar üretmesine sebep olmaktadır. Bundan dolayı bu interpolasyon tipi için  $k$  adet noktanın ağırlıklarının hesaplanarak *ağırlıklı en yakın k-komşu interpolasyon* yöntemi ile interpolate edilecek nokta için hata oranı daha düşük sonuç hesaplanabilmektedir [22, 23]. *Ağırlıklı*

en yakın  $k$ -komşu interpolasyon yönteminde de (4.1) ile ifade edilen Öklid uzaklık bağıntısı kullanılarak  $p_i$  noktasına en yakın  $k$  nokta belirlenir. Daha sonra bulunan noktaların  $p_i$  noktasına olan uzaklıklarına göre ağırlıkları (4.3) bağıntısı ile hesaplanır.

$$c_n = \frac{\sum d - d_n}{\sum d \cdot (k-1)}, 1 \leq n \leq k \quad (4.3)$$

Hesaplanan ağırlıklara göre interpolate edilmek istenen noktanın değeri (4.4) ağırlıklı ortalama bağıntısı ile hesaplanır.

$$f(x_{p_i}, y_{p_i}) = \frac{\sum_{l=1}^k c_l \cdot f(x_{p_l}, y_{p_l})}{k} \quad (4.4)$$



Şekil 4.1: En yakın  $k$ -komşu interpolasyonu.

## 4.2 Paralel Ayrık Fourier Dönüşümü

Görüntülenmek istenen hedef bölgesi için hareketli bir platform üzerine yerleştirilen radar vasıtasıyla azimuth ve range eksenlerinde toplanan verilerin birleştirilmesiyle 2-B bir veri matrisi elde edilir. İçerisinde hedef bölgesi bölgesi ile ilgili toplamış bilgileri içeren bu matris üzerine Ayrık Fourier Dönüşümü (DFT) operasyonunun uygulanarak verilerin kartezyen koordinatlara taşınabileceğinden daha önceki bölümlerde bahsetmiştik. Elde edilen verinin iki boyutlu olmasından dolayı (4.5) denklemi ile hesaplanabilen 2-B DFT operasyonunun gerçekleştirilmesi sağlanır.

$$F(X, Y) = \sum_{x=0}^N \sum_{y=0}^M f(x, y) e^{-j\pi(X \cdot x + Y \cdot y)} \begin{cases} X = 0, 1, 2, \dots, N \\ Y = 0, 1, 2, \dots, M \end{cases} \quad (4.5)$$

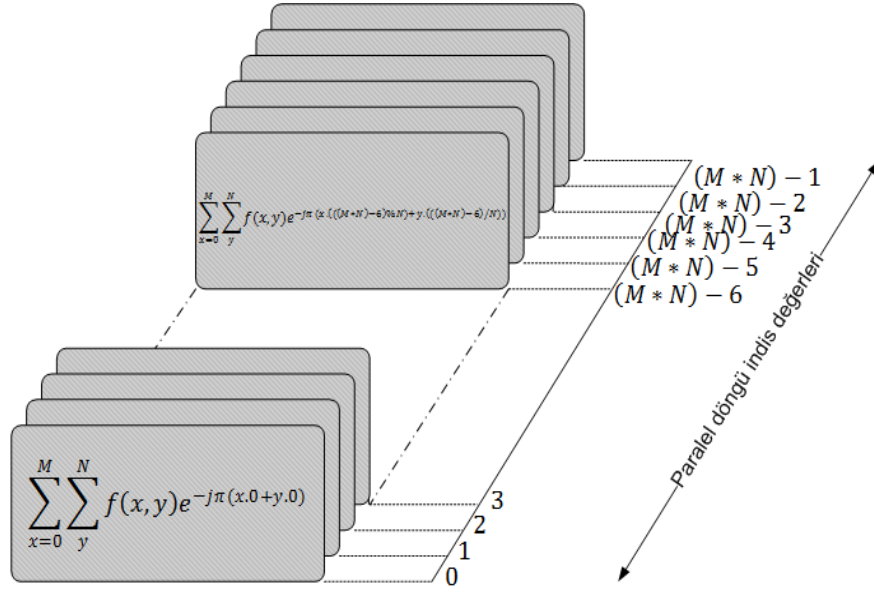
Yukarıdaki formülde de görüldüğü üzere  $N \times M$  boyularında bir görüntünün 2-B DFT operasyonu ile hesaplanması  $(N \times M)^2$  kadar işlem gerekmektedir. Taranan alanın

büyükliğünün artması sonucunda elde edilen veri sayısında artmaktadır. Bununla birlikte toplanan veri üzerindeki artış DFT'nin hesaplanması sırasında gerçekleştirilen işlemler sonucunda işlem yükünde karesel bir artış görülmektedir. Buna göre geniş bir alanın görüntüsünün yüksek çözünürlüklü olarak elde etmek istediğimizde çok fazla sayıda veri toplanması gerektiği için çok fazla işlem yükü oluşacaktır. Bu nedenden dolayı da DFT'nin hızlı hesaplanmasını sağlamak için paralelleştirilmesi sağlanmıştır. Normal şartlarda 2-B DFT işleminin hesaplanması için  $X, Y, x$  ve  $y$  için ayrı ayrı döngüler oluşturulması ve bu döngülerin iç içe kullanılarak hesaplanmanın gerçekleştirilmesi sağlanır. Ancak iç içe kullanılan döngüler üzerine yapılan paralelleştirme işlemi yalnızca bir döngü üzerinden sağlanabilir. Döngülerin hepsi paralel olarak işletilse bile her bir dış döngü kendisine göre içte yer alan döngünün bitmesini bekleyecektir. Diğer bir söylemle, her bir dış döngü kendi iç bloğunda tanımlanmış döngüye bağımlı olarak işlem görmektedir. Parallelleştirmeyi arttırmak için bu bağımlılığı ortadan kaldırmak gerekmektedir. Bunun için C programlama dili derleyicilerinin program içerisinde kullanılan dizileri boyutuna bakmaksızın tek boyuta indirgeyerek bellek üzerinde saklaması özelliği kullanılarak verilerin içerisinde saklandığı 2-B dizinin elemanlarına erişmek için kullanılan döngüler yeniden düzenlenir. Bu kapsamda (4.5) denkleminde yer alan  $F(X, Y)$  matrisinin tek bir döngü kullanılarak kontrol edilmesi ve elemanlarına erişilmesi sağlanmıştır. Döngüler üzerinde yapılan bu indirgeme işlemi  $F(X, Y)$  terimini kontrol edecek olan  $(N \times M)$ 'lik dönme işleminin tümünün paralelleştirilebilmesini sağlayacaktır. DFT hesaplamada  $F(X, Y)$  için kullanılan  $X$  ve  $Y$  indis değerleri eşitliğin sol tarafında bulunan hesaplama işleminde parametre olarak kullanılmaktadır. Bu nedenle (4.5) denkleminin sol tarafında yer alan ifade sağ tarafından gelen indis değerlerine bağımlıdır. Bundan dolayı yukarıdaki ifadenin sol tarafı için kullanılan döngü ile sağ tarafında hesaplama yapmak için kullanılan döngüler ayrıştırılamamaktadır.

Uygulama bünyesinde GPU üzerinde DFT'nin paralelleştirilmesi için  $F(X, Y)$  matrisini kontrol edecek olan döngü paralel olarak işletilmiş ve hesaplamalar gerçekleştirilmiştir. P-DFT'nin gerçekşetirmesi için kullanılan hesaplama mimarisi Şekil 4.2 üzerinde gösterilmiştir.

Buna göre her bir  $F(X, Y)$  elemanının hesaplanması için  $X$  ve  $Y$  indis değerlerini kontrol eden ortak bir döngü kurulmuştur. Kurulan döngü paralel olarak çalışması



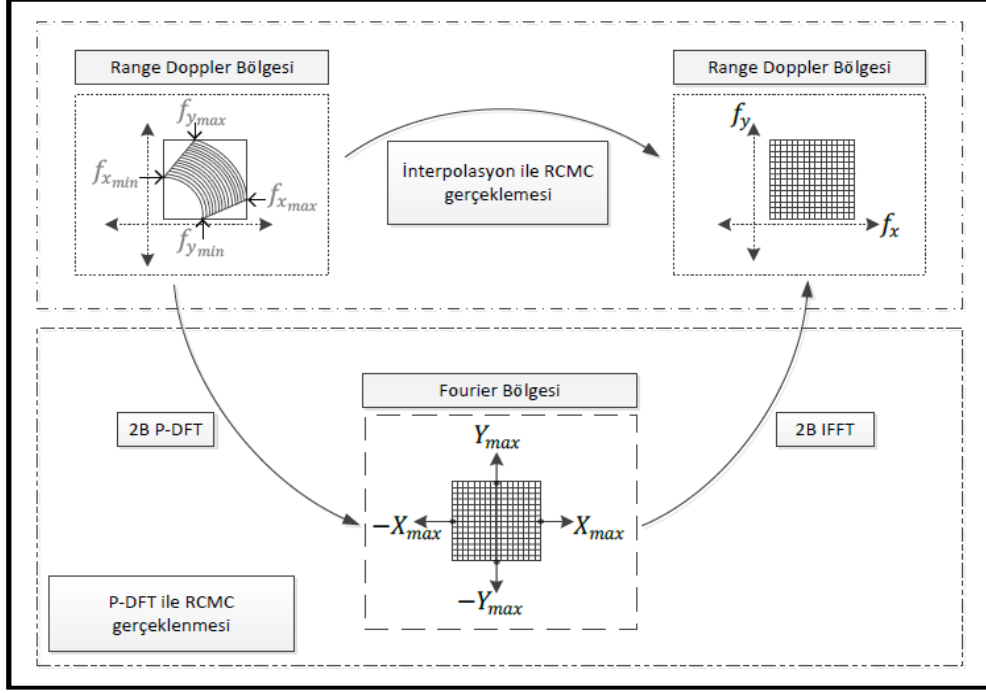


**Şekil 4.2:** Paralel ayrık fourier dönüşümü.

için tasarlanmıştır. Paralel çalışması için tasarlanan döngünün her bir indisi için DFT hesabına ilişkin hesaplama yapılmış ve hesaplanan değerler sonuç dizisi üzerinde toplanmıştır.

### 4.3 İnterpolasyon Kullanmaksızın RDA Gerçekleme

Bu bölümde RDA'nın en önemli adımı olan RCMC'nin içerisinde gerçekleştirilen interpolasyon sürecinin yerine P-DFT kullanılarak hem elde edilen görüntüdeki hata oranının minimize edilmesi, hemde sonuca yönelik hesaplama süresinin büyük oranda düşürülmesi sağlanmıştır. DFT operasyonu bir verinin değerini hesaplarken veri kümesi içerisindeki tüm noktaların değerlerini de hesaba katar ve bu sayede ilgili nokta üzerine etkiyecek hatanın minimuma inmesini sağlar. RDA'nın uygulanması sırasında RCMC adımıyla interpolasyona uygulayacak veri Range-Doppler bölgesinde bulunmaktadır. İnterpolasyon işlemi tüm hesaplamalarını bu bölge üzerinde gerçekleştirir ve yine aynı bölge üzerinde sonuç üretir. Ancak RCMC adımıyla uygulanan DFT ile birlikte Range-Doppler bölgesinden Fourier bölgesine geçiş yapılır. Ancak RDA'da RCMC adımının sonucunda elde edilen verinin Range-Doppler bölgesinde olması gerektiğinden DFT ile elde edilen sonuca IFFT uygulanır ve Range-Doppler bölgesine geri dönlür. Şekil 4.3 üzerinde belirtilen blok diyagram üzerinde RCMC'nin hem interpolasyon hem de P-DFT uygulanarak gerçekleştirilmesi gösterilmiştir.



**Şekil 4.3:** RCMC üzerinde P-DFT gerçekleştirme.

Daha öncede bahsedildiği gibi RCMC üzerinde gerçekleştirilen interpolasyon kartezyen olmayan koordinatlarda bulunan değerleri kartezyen koordinatlar üzerinde ifade edebilmek için kullanılır. İnterpolasyon ile gerçekleştirilen bu işlemin DFT ile gerçekleştirilmesi için görüntünün kartezyen koordinat düzlemindeki yerlerine ihtiyaç duyulmaktadır. Kartezyen noktaların yerleri kartezyen olmayan koordinatlarda bulunan verinin destek bölgesi kullanılarak analitik olarak hesaplanabilmektedir. Kartezyen noktalarda elde edilen görüntünün boyutu azimuth ve range ekseninde alınan örneklerin sayısı olan  $N_{Azimuth}$  ve  $N_{Range}$  ile ifade edilir. Buna göre elde edilecek görüntü  $N_{Azimuth} \times N_{Range}$  boyutunda olacaktır. Kartezyen noktalarda olmayan verinin destek bölgesininin kullanılmasıyla elde edilen kartezyen koordinatları (4.6) bağıntısı ile ifade edilir.

$$X[i] = -X_{max} + (i - 1)\Delta X$$

$$Y[i] = -Y_{max} + (i - 1)\Delta Y \quad (4.6)$$

Yukarıdaki bağıntıda yer alan  $\Delta X$  ve  $\Delta Y$  kartezyen noktalar arasındaki uzaklığı ifade etmektedir ve (4.7) bağıntısı ile hesaplanabilmektedir.

$$\Delta X = \frac{2X_{max}}{N_{Range} - 1}$$

$$\Delta Y = \frac{2Y_{max}}{N_{Azimuth} - 1} \quad (4.7)$$

Bağıntı (4.7) içerisinde yer alan  $X_{max}$  ve  $Y_{max}$  ise Fourier bölgesindeki maksimum ve minimum noktaları ifade etmektedir. Bu ifadeler (4.8) bağıntısı kullanılarak hesaplanabilmektedir.

$$\begin{aligned} X_{max} &= \frac{1}{2\Delta\hat{f}_x} \\ Y_{max} &= \frac{1}{2\Delta\hat{f}_y} \end{aligned} \quad (4.8)$$

Yukarıdaki bağıntıda yer alan  $\Delta\hat{f}_x$  ve  $\Delta\hat{f}_y$  Range-Doppler bölgesinde destek bölgesindeki hem  $x$  hem de  $y$  eksenine düzenli dağılımı ifade eder ve (4.9) bağıntısı ile hesaplanabilmektedir.

$$\begin{aligned} \Delta\hat{f}_x &= \frac{f_{xmax} - f_{xmin}}{N_{Range} - 1} \\ \Delta\hat{f}_y &= \frac{f_{ymax} - f_{ymin}}{N_{Azimuth} - 1} \end{aligned} \quad (4.9)$$

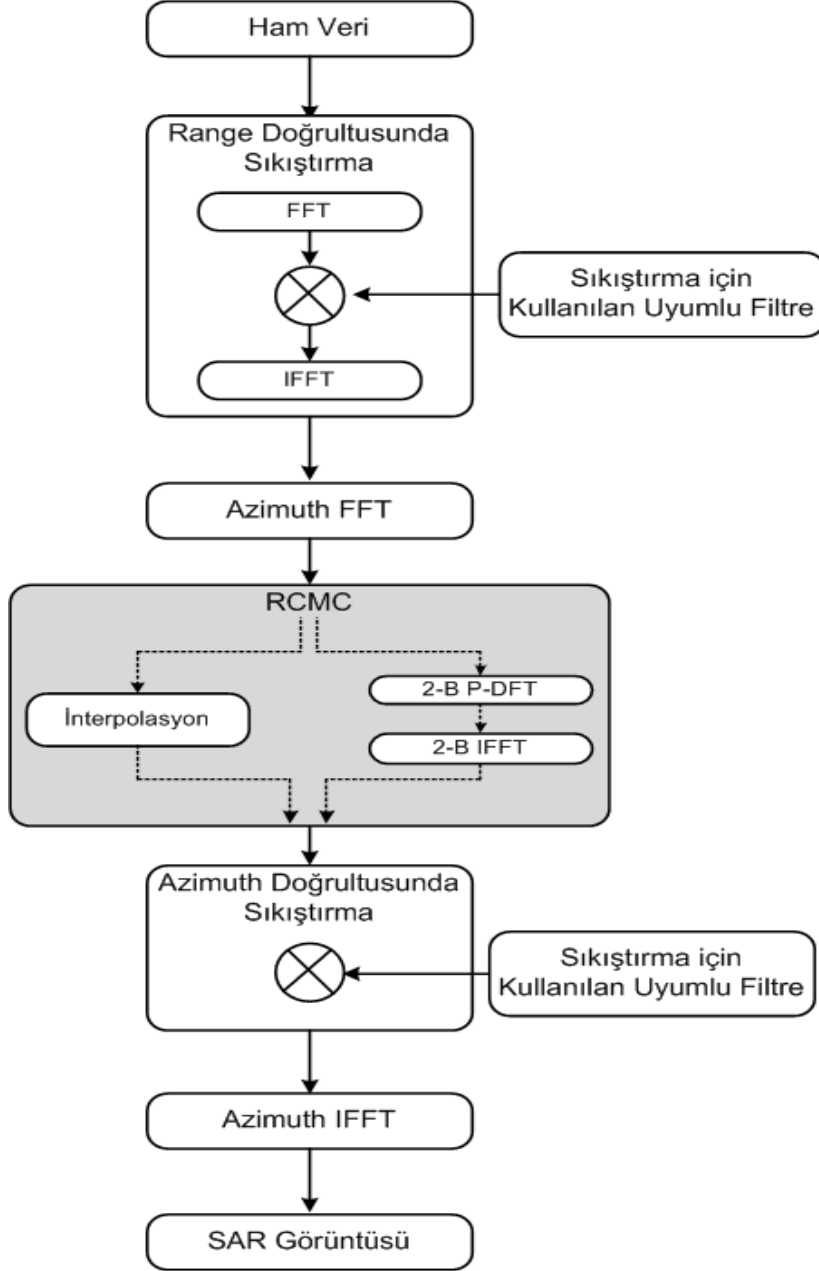
Yukarıdaki bağıntılarla elde edilmek istenen görüntünün kartezyen koordinatlardaki yerleri belirlendikten sonra verinin kartezyen olmayan koordinatlardaki bilgisi ve verinin ilgili koordinattaki değerini kullanılarak 2-B DFT operasyonu (4.10) denklemi ile gerçekleştirilir.

$$F(X, Y) = \sum_{x=0}^{N_{azimuth}} \sum_{y=0}^{N_{range}} f(x, y) e^{-j\pi(X \cdot x + Y \cdot y)} \quad (4.10)$$

Bu denklemde  $X$  ve  $Y$  verinin kartezyen koordinatlardaki yerlerini,  $x$  ve  $y$  kartezyen olmayan koordinatlardaki yerlerini ve  $f(x, y)$  ise verinin  $x$  ve  $y$  noktalarındaki değerini ifade etmektedir. DFT operasyonu uygulanması yönüyle işlem yükü fazla bir işlemdir. Yukarıdaki (4.10) denklemde görüldüğü gibi her bir  $(X, Y)$  noktası için  $N_{azimuth} \times N_{range}$  kadar işlem yapılmaktadır. Bunun yanısıra  $F(X, Y)$  için  $X$  ekseninin  $N_{azimuth}$ ,  $Y$  ekseninin de  $N_{range}$  aralığında değer aldığı göz önünde bulundurulursa DFT işleminin uygulanması  $(N_{azimuth} \times N_{range})^2$  adet işlem gerektirecektir. Elde edilmek istenen görüntüdeki nokta sayısı arttıkça da DFT'deki işlem yükü karesel olarak artacaktır. Bununla doğru orantılı olarak da verinin işlenme süreside artacaktır. Bu nedenle DFT işleminin GPU üzerinde paralel olarak işlenmesiyle işlem zamanının düşürülmesi sağlanmıştır.

Range-Doppler bölgesi üzerinde bulunan veriye DFT operasyonu uygulanmasıyla Fourier bölgesine geçiş yapılmaktadır. Ancak RCMC'den sonra gelen azimuth doğrultusunda sıkıştırma adımı Range-Doppler bölgesi üzerinde uygulanmaktadır. Bu

nedenle geçiş yapılan Fourier bölgesinde DFT operasyonu uygulanmasıyla elde edilen sonuca 2-B IFFT uygulanmasıyla Range-Doppler bölgesine geri dönülmesi sağlanır ve azimuth doğrultusunda sıkıştırma işlemi uygulamaya alınır.

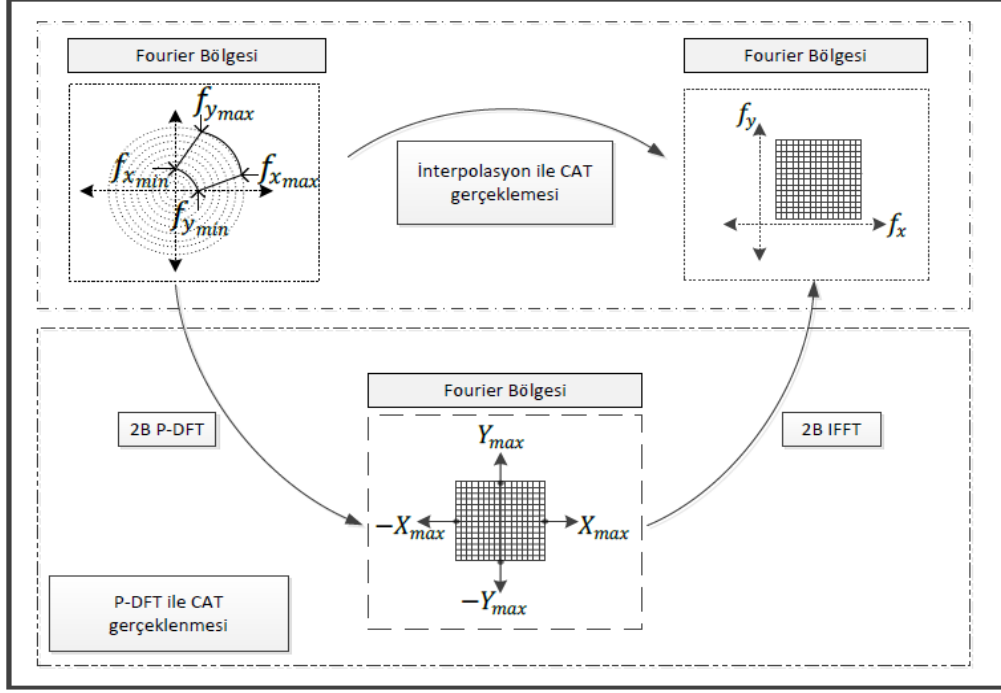


Şekil 4.4: Paralel Menzül-Doppler algoritması.

#### 4.4 İnterpolasyon Kullanmaksızın CAT Gerçekleme

Bu bölümde CAT yönteminin temelini oluşturan izdüşüm-dilim teoreminin uygulanması sonucunda elde edilen kutupsal koordinatlardaki verinin kartezyen koordinat sistemine taşınması işlemi için interpolasyon sürecinin uygulanması yerine

P-DFT yönteminin uygulanmasından bahsedilecektir. Şekil 4.5 üzerinde belirtilen blok diyagram üzerinde CAT algoritmasının hem interpolasyon hem de P-DFT uygulanarak gerçekleştirilmesi gösterilmiştir. Bölüm 4.3’de açıklandığı gibi P-DFT kullanımı hem interpolasyon sürecinin uygulanmasından doğan hataların minimize edilmesini sağlar hem de daha kolay paralelleştirilebilir yapıya sahip olmasından dolayı sonuca yönelik hesaplama süresinin büyük oranda düşürülmesini sağlar. CAT



Şekil 4.5: CAT üzerinde P-DFT gerçekleştirme.

yönteminin gerçekleştirilmesi sırasında izdüşüm-dilim teoreminin uygulanması sonucu elde edilen veri Fourier bölgesi üzerinde ve kutupsal koordinat düzleminde ifade edilir. Bilindiği gibi SAR görüntü algoritmalarının uygulanmasıyla elde edilen sonucun görüntü olarak işlenebilmesi için uzamsal bölge üzerinde ifade edilmesi gerekmektedir. Bu nedenle Fourier bölgesinden uzamsal bölgeye geçiş yapılması için 2-B IFFT uygulanması gerekmektedir. Ancak IFFT operasyonunun veri üzerine uygulanması için kutupsal koordinat sisteminden kartezyen koordinat sistemine geçiş yapılması gerekmektedir. Bu dönüşümü sağlamak için interpolasyon uygulanması gerekmektedir.

Yöntemin uygulanması esnasında interpolasyon yerine P-DFT kullanılması Bölüm 4.3’deki uygulama örneğinden biraz daha farklı gerçekleştirilmektedir. RDA algoritması üzerinde P-DFT’nin veri üzerine uygulanmasıyla Range-Doppler

bölgesinden Fourier bölgesine geçiş yapılmakta ve daha sonra 2-B IFFT ile tekrar Range-Doppler bölgesine geri dönülmektedir. Ancak CAT üzerinde P-DFT uygulanması sırasında verinin Fourier bölgesinde olmasından dolayı bulunan bölge değişmeden sadece kutupsal koordinatlardan kartezyen koordinatlara geçişi sağlamaktadır. Bundan dolayı P-DFT'nin uygulanmasından sonra gelen adımda uygulanan 2-B IFFT Fourier bölgesinden uzamsal bölgesine geçişi sağlar.

CAT yöntemi üzerinde P-DFT'nin uygulanması için öncelikle kartezyen koordinatların yerleri analitik olarak hesaplanmalıdır. Bunun için elde edilen veriden görüntülenmek istenen alanın sınırları hem  $X$  hem de  $Y$  eksenlerinde belirlenir. Belirlenen sınır noktalarına göre azimuth ve range doğrultusunda elde edilen örnek sayıları kullanılarak kartezyen noktalar arası uzaklıklar belirlenir ve kartezyen koordinatlar elde edilir. Kartezyen koordinatların yeri (4.11) bağıntısı ile ifade edilir.

$$\begin{aligned} X[i] &= X_{min} + (i - 1)\Delta X \\ Y[i] &= Y_{min} + (i - 1)\Delta Y \end{aligned} \quad (4.11)$$

Yukarıdaki bağıntıda yer alan  $X_{min}$  ve  $Y_{min}$  görüntülenmek istenen alanın kartezyen koordinat sistemine göre sırasıyla  $X$  ve  $Y$  eksenlerindeki minimum sınır noktalarını,  $\Delta X$  ve  $\Delta Y$  ise kartezyen noktalar arasındaki uzaklığı ifade etmektedir ve (4.12) bağıntısı ile hesaplanabilmektedir.

$$\begin{aligned} \Delta X &= \frac{X_{max} - X_{min}}{N_{Range} - 1} \\ \Delta Y &= \frac{Y_{max} - Y_{min}}{N_{Azimuth} - 1} \end{aligned} \quad (4.12)$$

Bağıntı (4.12) içerisinde yer alan  $X_{max}$  ve  $Y_{max}$  ise görüntülenmek istenen alanın kartezyen koordinat sistemine göre maksimum sınır noktalarını ifade etmektedir. Yukarıda açıklanan görüntülenmek istenen alanın sınır değerleri parametrik olarak görüntüleme belirlenebilmektedir. Bu sayede kutupsal koordinatlardan kartezyen koordinatlara dönüşüm sırasında aynı zamanda görüntünün çözünürlüğünde değiştirilebilmektedir.

Bağıntı (4.11) ile ifade edilen kartezyen koordinat düzlemindeki noktaların yerlerinin elde edilmesinden sonra (4.10) denklemi ile ifade edilen DFT operasyonu uygulanarak

verinin kartezyen koordinat düzelemi üzerine interpolasyon kullanmaksızın taşınması sağlanır.

#### 4.5 Uygulama Sonuçları

Bu bölümde SAR görüntüleme algoritmalarının interpolasyon kullanmaksızın P-DFT yöntemi ile GPU üzerinde gerçekleştirilmesi için yapılan çalışmalar ile elde edilen sonuçlardan ve uygulamaların gerçekleştirildiği bilgisayar konfigürasyonundan bahsedilecektir. Yapılan çalışmalar hem CPU hem GPU üzerinde gerçekleştirilmiş ve elde edilen sonuçların karşılaştırılması yapılarak GPU'nun paralel işleme konusunda sağladığı avantajlar elde edilen sonuçlarla birlikte desteklenmiştir. Bunun yanı sıra, radar tarafından elde edilmiş veri üzerine hem interpolasyon hemde P-DFT yöntemi uygulanarak ortaya çıkan sonuç verisinin karşılaştırılması yapılmıştır. Yapılan çalışmanın gerçekleştirildiği bilgisayar konfigürasyonu Çizelge 4.1 üzerinde gösterilmektedir.

**Çizelge 4.1:** Test ortamı.

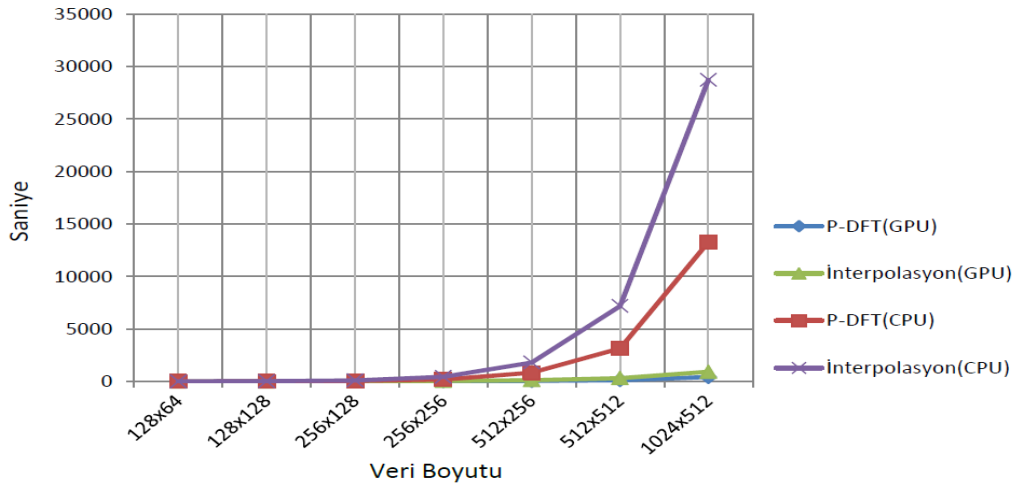
CPU	Intel Core I7-3820
CPU Frekansı	3.6 GHz
CPU Ön Bellek	10 MB
Komut Seti	64-bit
RAM	64 GB
GPU	NVIDIA Tesla C2075
CUDA Çekirdek Sayısı	448
GPU Bellek	6GB GDDR5
GPU Bellek Bant Genişliği	144 GB/sec
İşletim Sistemi	64-bit Windows 7 Professional

Yapılan çalışmanın performansının gösterilmesi için interpolasyon ve P-DFT yönteminin hem GPU hem de CPU üzerinde paralel olarak ayrı ayrı gerçekleştirilmesi sağlanmış ve elde edilen sonuçlar saniye olarak ifade edilmiştir. CPU üzerinde paralelleştirme yapabilmek için Açık Çoklu İşleme (Open Mutli-Processing – OpenMP) kullanılmıştır. OpenMP paralel olarak işlenmesi istenen bölümlerin başına özel direktif komutlarının yazılmasıyla o bölümün paralel olarak işlenmesini sağlar. GPU üzerinde paralelleştirme yapılması için ise CUDA kullanılmıştır. Sonuçların gösterilmesi için yapılan testler kapsamında range doğrultusunda kullanılan örnek sayısı olan  $N_{range} = 128$  ve azimuth doğrultusunda kullanılan örnek sayısı olan

$N_{azimuth} = 64$  olarak kullanılmıştır. Bunun yanısıra,  $N_{range}$  ve  $N_{azimuth}$  parametreleri sırasıyla (128,128), (256,128), (256,256), (512,256), (512,512) ve (1024,512) olarak güncellenerek yapılan testler tekrarlanmış ve sonuçların karşılaştırılması yapılmıştır.

**Çizelge 4.2:** RCMC üzerinde interpolasyon ve P-DFT performansı.

Veri Boyutu	İnterpolasyon		P-DFT Algotirması	
	CPU	GPU	CPU	GPU
128 × 64	7.05s	2.46s	3.28s	1.61s
128 × 128	28.15s	6.12s	12.35s	6.28s
256 × 128	112.70s	25.89s	51.54s	12.67s
256 × 256	450.28s	69.53s	196.83s	25.82s
512 × 256	1803.03s	135.38s	840.512s	55.80s
512 × 512	7189.88s	327.85s	3151.48s	135.38s
1024 × 512	28719.00s	927.39s	13267.59s	424.15s



**Şekil 4.6:** RCMC üzerinde interpolasyon ve P-DFT performansı.

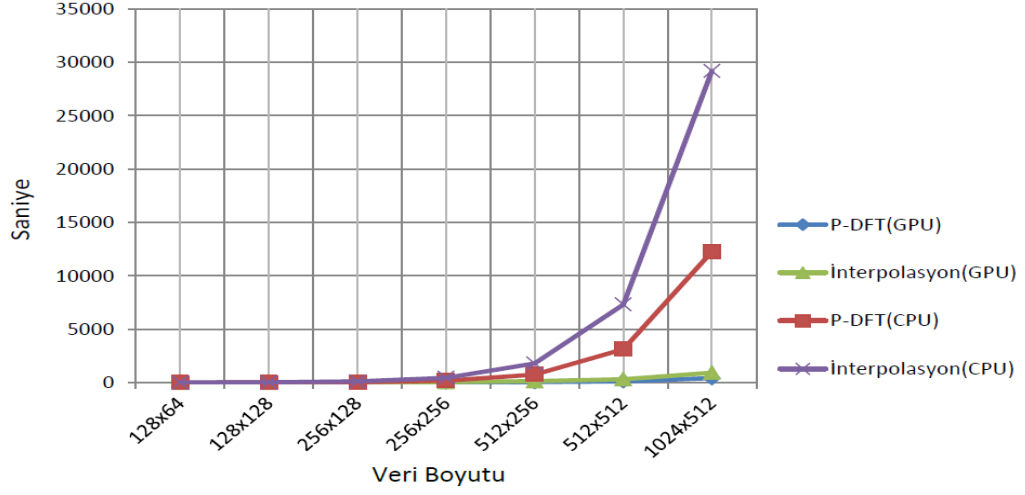
**Çizelge 4.3:** CAT üzerinde interpolasyon ve P-DFT performansı.

Veri Boyutu	İnterpolasyon		P-DFT Algotirması	
	CPU	GPU	CPU	GPU
128 × 64	7.06s	2.20s	2.98s	1.60s
128 × 128	28.39s	7.04s	12.45s	6.22s
256 × 128	124.58s	24.43s	47.84s	12.65s
256 × 256	462.94s	68.77s	195.02s	25.75s
512 × 256	1796.30s	136.65s	759.60s	55.44s
512 × 512	7325.08s	329.78s	3141.92s	134.27s
1024 × 512	29173.70s	931.15s	12253.52s	426.26s

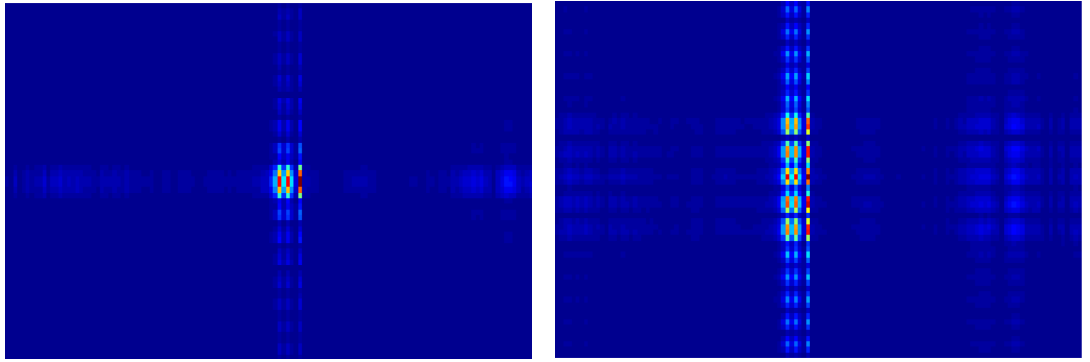
Yapılan çalışmanın performansı Çizelge 4.2 ve Çizelge 4.3 üzerinde hem RDA hem de CAT için ayrı ayrı gösterilmiştir. Buna göre P-DFT yönteminin uygulanmasının



interpolasyondan daha hızlı sonuç ürettiğini görülebilmektedir. Bunun yanında veri boyutunun artması durumunda interpolasyonun veri işleme zamanının büyük ölçüde arttığı da gözlemlenebilmektedir.

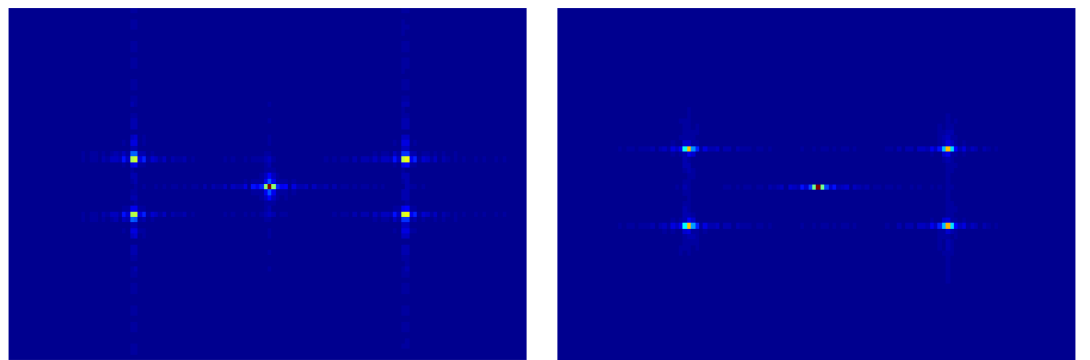


Şekil 4.7: CAT üzerinde interpolasyon ve P-DFT performansı.



(a) GPU kullanılarak elde edilen P-DFT sonucu (b) GPU kullanılarak elde edilen interpolasyon sonucu

Şekil 4.8: RDA üzerinde elde edilen sonuçlar.



(a) GPU kullanılarak elde edilen P-DFT sonucu (b) GPU kullanılarak elde edilen interpolasyon sonucu

Şekil 4.9: CAT algoritması üzerinde elde edilen sonuçlar.

Interpolasyonun ve P-DFT yönteminin hem RDA hem CAT üzerinde uygulanmasıyla elde edilen görüntüler Şekil 4.8 ve Şekil 4.9 üzerinde örneklenmiştir. Şekil üzerinde

gösterilen görüntülerin range ve azimuth eksenindeki örnekleri 128 ve 64 olarak belirlenmiştir. Şekil 4.8(a) RDA algoritmasında üzerinde gerçekleştirilen P-DFT yöntemi ile elde edilen görüntüyü, Şekil 4.8(b) ise RDA algoritması üzerinde gerçekleştirilen interpolasyon ile elde edilen görüntüyü göstermektedir. Aynı şekilde Şekil 4.9(a) CAT algoritmasının üzerinde gerçekleştirilen P-DFT yöntemi ile elde edilen görüntüyü, Şekil 4.9(b) ise CAT algoritması üzerinde gerçekleştirilen interpolasyon ile elde edilen görüntüyü göstermektedir.

## 5. SONUÇ VE ÖNERİLER

Yapılan tez çalışması ile SAR görüntü algoritmalarının akışında yer alan interpolasyon algoritmasının kaldırılarak yerine eş değer bir operasyon olarak düzenlenen P-DFT operasyonunun uygulanması sağlanmıştır. Radar vasıtasıyla ilgili hedef bölgesi üzerinden toplanan veri kartezyen koordinat sisteminde olmayan noktalar ile ifade edilmektedir. Bu sebeple radar görüntüleme algoritmalarının uygulanması sırasında veri üzerine FFT operasyonu uygulanamamaktadır. Bundan dolayı verilerin kartezyen noktalar üzerine taşınması gerekmektedir. Bu işlemi gerçekleştirmek için radar görüntüleme algoritmalarının hemen hemen hespi interpolasyon algoritmalarını kullanmaktadır. Ancak interpolasyon sonucu elde edilen veri belirli bir hata payı içermekte ve bu hata payının büyük olması istenilen görüntünün kalitesini bozmaktadır. Yapılan çalışmada radar görüntüleme algoritmaları içerisinde bulunan interpolasyon algoritması yerine DFT operasyonunun uygulanması veri üzerine etkileyen hata oranının minimuma indirilmesi sağlanmıştır. Aynı zamanda içerisinde belirsiz bir arama süreci bulunan ve bu nedenden dolayı da paralelleştirilmesi zor olan interpolasyon algoritmasının yerine daha kolay paralelleştirilebilen DFT operasyonu uygulanması sağlanmıştır. Yapılan paralelleştirme çalışmaları sırasında GPU'nun paralel çalışmaya yatkın olan mimarisi kullanılmış ve DFT'nin GPU üzerinde paralel olarak yürütülmesi sağlanmıştır.

Yapılan tez çalışması RDA algoritması içerisinde bulunan RCMC adımı üzerinde ve CAT algoritmasının işletimi sırasında kutupsal koordinat sisteminden kartezyen sisteme dönüşümün sağlandığı adım üzerinde uygulanmıştır. Yapılan test çalışmalarında RDA ve CAT algoritmaları hem interpolasyon hem de P-DFT kullanılarak gerçekleştirilmiştir. RDA ve CAT algoritmalarının interpolasyon ve P-DFT kullanılarak gerçekleştirilmesi işlemi de hem CPU hem de GPU üzerinde ayrı ayrı olarak gerçekleştirilmiştir. Bu işlemler sonrasında paralel olmayan çözüme göre zaman yönünden avantaj sağlamıştır. Bunun yanısıra interpolasyon uygulanmasının sonucunda elde edilen görüntü üzerine etkileyen bozulmalar açıkça görülmektedir. Bu kapsamda P-DFT'nin

hem CPU hem de GPU üzerinde iřletilmesi paralel interpolasyon uygulaması sonuçlarına göre hem zaman hem de doęruluk yönünden avantaj saęlamıřtır. Ayrıca görüntüde bulunan veri sayısının artmasıyla oluşturulan uygulamanın daha hızlı sonuç üretilmesi bağlamında GPU üzerinde P-DFT uygulaması zaman yönünden büyük avantaj saęlamaktadır.

Tez kapsamında yapılan uygulama nokta saçıcı görüntüsü üzerinde gerçekleştirilmiş ve sonuçları ortaya konulmuřtur. Tez çalışmasının gerçekleştirilmesiyle elde edilen sonuçlar üzerinden uluslararası bir konferansta bildiri yayını yapılmıřtır. Bundan sonra yapılacak çalışmalarda bir bölge üzerinden toplanan gerçek veriler kullanılarak P-DFT yönteminin başarımını test edilecektir. P-DFT ile elde edilecek gerçek görüntüler üzerine belirli görüntü işleme operasyonlarının uygulanması saęlanacak ve radar görüntüleri üzerinden bilgi elde edilmesi üzerine çalışmalar gerçekleştirilecektir. Elde edilen sonuçlar ile yayın çalışmalarına devam edilecektir.

## KAYNAKLAR

- [1] **Yazgan Erer, I.** (2001). İki Boyutlu Kompleks Kafes Yapısı ile Yüksek Çözünürlüklü Terz Yapay Açıklıklı Radar Görüntüleme, *Doktora Tezi*, İstanbul Teknik Üniversitesi.
- [2] **Munson, D.C., O'Brien, J.D. ve Jenkins, W.K.** (1983). A tomographic formulation of spotlight-mode synthetic aperture radar, *Proceedings of the IEEE*, **71**(8), 917–925.
- [3] **Lawton, W.**, (1988), A new polar Fourier transform for computer-aided tomography and spotlight synthetic aperture radar.
- [4] **Bin Liu, Kaizhi Wang, X.L. ve Yu, W.** (2009). An Efficient SAR Processor Based on GPU via CUDA, *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, s.1 –5.
- [5] **Ning, X., Yeh, C., Zhou, B., Gao, W. ve Yang, J.** (2011). Multiple-GPU accelerated range-Doppler algorithm for synthetic aperture radar imaging, *Radar Conference (RADAR), 2011 IEEE*, s.698 –701.
- [6] **Liu, B., Wang, K., Liu, X. ve Yu, W.** (2010). Range Cell Migration Correction using texture mapping on GPU, *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*, s.2172–2175.
- [7] **Ozgun Altun, Selcuk Paker, M.K.** (2013). Realization of Interpolation-free Fast SAR Range-Doppler Algorithm Using Parallel Processing on GPU, *PIERS 2013-Taipei*, s.1020–1024.
- [8] **Cumming, I.G. ve Wong, F.H.C.** (2005). *Digital Processing Of Synthetic Aperture Radar Data: Algorithms And Implementation*, The Artech House Remote Sensing Library, Artech House.
- [9] **Hartley, T.D.R., Fasih, A.R., Berdanier, C.A., Ozguner, F. ve Catalyurek, U.V.**, (2009), Investigating the use of GPU-accelerated nodes for SAR image formation.
- [10] **Campbell, J.B. ve Wynne, R.H.** (2011). *Introduction to Remote Sensing*, Guilford Press.
- [11] **Camps-Valls, G., Tuia, D., Gomez-Chova, L., Jimenez, S. ve Malo, J.** (2011). Remote Sensing Image Processing, *Synthesis Lectures on Image, Video, and Multimedia Processing*, **5**(1), 1–192.
- [12] **Soergel, U.** (2010). *Review of Radar Remote Sensing on Urban Areas*, Remote Sensing and Digital Image Processing, Springer Netherlands.

- [13] **Chan, Y.K. ve Koo, V.C.** (2008). An Introduction To Synthetic Aperture Radar(SAR), *Progress In Electromagnetics Research B*, 2(6), 27–60.
- [14] **Franceschetti, G. ve Lanari, G.** (1999). *Synthetic Aperture Radar Processing*, CRC Press.
- [15] **Cumming, I. ve Bennett, J.** (1979). Digital processing of Seasat SAR data, *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '79.*, cilt 4, s.710–718.
- [16] **Jin, M.Y.**, (1997), High quality spotlight SAR processing algorithm designed for LightSAR mission.
- [17] **Scudder, H.J.**, (1978), Introduction to computer aided tomography.
- [18] **Herman, G.T.** (1995). Image reconstruction from projections, *Real-Time Imaging*, 1(1), 3–18.
- [19] **L. Mensa, D.** (1991). *High Resolution Radar Cross-Section Imaging*, Artech House.
- [20] **NVIDIA** (2008). *NVIDIA CUDA: Compute Unified Device Architecture Programming Guide*, NVIDIA Corporation.
- [21] **Kirk, D.B. ve Hwu, W.m.W.** (2010). *Programming Massively Parallel Processors: A Hands-on Approach*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st sürüm.
- [22] **Kawabata, T.** (1991). Generalization effects of k-neighbor interpolation training, *Neural Comput.*, 3(3), 409–417.
- [23] **Mobarakeh, A., Rizi, S., Khaniabadi, S., Bagheri, M. ve Nazari, S.** (2012). Applying Weighted K-nearest centroid neighbor as classifier to improve the finger vein recognition performance, *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on*, s.56–59.

## ÖZGEÇMİŞ

**Ad Soyad:** Özgür Altun

**Doğum Yeri ve Tarihi:** Eminönü - 17.02.1987

**E-Posta:** ozguraltun@gmail.com

**Lisans:** Kocaeli Üniversitesi - Bilgisayar Mühendisliği

**Y. Lisans:** İstanbul Teknik Üniversitesi - Uydu Haberleşmesi ve Uzaktan Algılama Programı

### TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

- Ozgur Altun, Selçuk Paker, Mesut Kartal, 2013: Realization of Interpolation-free Fast SAR Range-Doppler Algorithm Using Parallel Processing on GPU *PIERS 2013 - Taipei*, March 25-28, 2013 Taipei, Taiwan.