

**İŞLENMEMİŞ YAPAY AÇIKLIKLI RADAR
VERİLERİNİN SIKIŞTIRILMASI**

107079

YÜKSEK LİSANS TEZİ
Müh. N. Gökhan KASAPOĞLU
504961062

107079

Tezin Enstitüye Verildiği Tarih : 5 Haziran 2000
Tezin Savunulduğu Tarih : 21 Haziran 2000

Tez Danışmanı : Prof.Dr. Bingül YAZGAN
Diğer Jüri Üyeleri Prof.Dr. Ahmet H. KAYRAN (İ.T.Ü.)
Doç.Dr. Osman N. UÇAN (İ.Ü.)

HAZİRAN 2000

ÖNSÖZ

Tez çalışmamı yönlendiren Tez Danışmanım Sayın Prof. Dr. Bingöl YAZGAN'a ve bana her konuda yardımcı olan aileme teşekkür ederim.

Haziran, 2000

N.Gökhan KASAPOĞLU

YAZGAN
BİNGÖL
KASAPOĞLU
N. GÖKHAN



İÇİNDEKİLER

ÖNSÖZ	ii
KISALTMALAR	iv
TABLO LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÖZET	viii
SUMMARY	xii
1. GİRİŞ	1
2. YAPAY AÇIKLIKLI RADAR	4
2.1 YAR Görüntüleme Sistemi	4
2.2 İşlenmemiş YAR Verisi İstatiksel Özellikleri	9
2.3 YAR Veri Sıkıştırma Modeli	11
2.3.1 Dönüşüm	13
2.3.2 Kayıpsız Kodlama	15
3.İŞLENMEMİŞ YAR VERİSİ İÇİN SIKIŞTIRMA YÖNTEMLERİ	18
3.1 Blok Adaptif Kuantalama Algoritması	18
3.2 Dönüşüm Domeninde Sıkıştırma	22
3.2.1 JPEG ile Görüntü Kodlama	22
3.2.2 İşlenmemiş YAR Verilerinin Dönüşüm Domeninde Sıkıştırılması	24
4.İŞLENMİŞ YAR VERİSİ SIKIŞTIRMA TEKNİKLERİ PERFORMANS ANALİZİ	26
4.1 İşlenmiş YAR Verisi Sıkıştırma Teknikleri Performans Kriterleri	26
4.2 Performans Analizinde Kullanılan Örnek Veriler	28
4.3 Blok Adaptif Kuantalama Algoritması Performans Analizi	30
4.4 Dönüşüm Domeninde DCT-BAQ Algoritması Performans Analizi	42
4.5 Zaman Domeninde BAQ ile Dönüşüm Domeninde DCT-BAQ Algoritmalarının Birbirleriyle Karşılaştırılması	48
5. SONUÇLAR VE ÖNERİLER	53
KAYNAKLAR	56
ÖZGEÇMİŞ	58

KISALTMALAR

A/D	: Analog -to-Digital Transform	Analog Sayısal Dönüştürücü
BAQ	: Block Adaptive Quantization,	Blok Adaptif Kuantalama
CR	: Compression Ratio,	Sıkıştırma Oranı
DCT	: Discrete Cosine Transform,	Ayrık Kosinüs Dönüşümü
FFT	: Fast Fourier Transform,	Hızlı Fourier Dönüşümü
I	: In Phase,	Aynı Fazda
JPEG	: Joint Photographics Expert Group	
KL	: Karhunen-Loeve Transform	
MSE	: Mean Square Error,	Ortalama Karesel Hata
NMSE	: Normalized Mean Square Error,	Normalize Ortalama Karesel Hata
PSNR	: Pick Signal to Noise Ratio,	Tepe İşaret Gürültü Oranı
DST	: Discrete Sine transform	Ayrık Sinüs Dönüşümü
Q	: In-Quadrature,	90 Derece Faz Farklı
SAR	: Synthetic Aperture Radar	Yapay Açıklıklı Radar
SNR	: Signal to Noise Ratio,	İşaret Gürültü Oranı
YAR	: Yapay Açıklıklı Radar	

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. Huffman kodlaması için sembollerin birleştirilmesi	16
Tablo 2.2. Huffman kodlaması için sembollerin ayrıştırılması ve kod kelimelerinin atanması	17
Tablo 3.1. Uniform olmayan Gauss (BAQ) Kuantalayıcı sınırları ve tekrar oluşturma seviyeleri	20
Tablo 3.2. Uniform olmayan Gauss (BAQ) Kuantalayıcı sınırları ve tekrar oluşturma seviyeleri	21
Tablo 4.1. BAQ 2 Bit Algoritması Performans Analizi	39
Tablo 4.2. BAQ 3 Bit Algoritması Performans Analizi	40
Tablo 4.3. BAQ 4 Bit Algoritması Performans Analizi	41
Tablo 4.4. Dönüşüm domeninde DCT-BAQ Alogoritması Performans Analizi	47

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1 : Menzil yönünde radar görüntüleme geometrisi.....	4
Şekil 2.2 : Azimuth yönünde YAR Görüntüleme Geometrisi.....	5
Şekil 2.3 : Kayıplı Sıkıştırma Blok Diyagramı.....	13
Şekil 2.4 : Dalgacık dönüşümü temelli sıkıştırma sistemi blok diyagramı.....	14
Şekil 3.1 : BAQ blok diagramı	18
Şekil 3.2 : İki bitlik BAQ algoritması için kodlama.....	20
Şekil 3.3 : YAR sistemi için BAQ ve BFPQ algoritmaları.....	21
Şekil 3.4 : 8x8 dönüşüm için zig zag tarama yapısı.....	23
Şekil 3.5 : İşlenmemiş YAR 8x8 veri bloğunun DCT katsayıları.....	23
Şekil 3.6 : İşlenmemiş YAR verilerinin dönüşüm domeninde sıkıştırılması...	25
Şekil 4.1 : İşlenmemiş YAR verileri performans karşılaştırılması için sıkıştırma sistemi.....	27
Şekil 4.2 : İşlenmemiş YAR verisi a) reel (I) bileşeni ve b) imajiner (Q) bileşeni.....	29
Şekil 4.3 : Uydudan elde edilmiş İşlenmiş YAR görüntüleri a)Noktasal gürültüden arındırılmış L.A. görüntüsü b)Noktasal gürültülü Hollanda görüntüsü.....	29
Şekil 4.4 : Uçaktan elde edilen Havaalanı a)YAR görüntüsü b)Optik görüntü	30
Şekil 4.5 : Lena	30
Şekil 4.6 : İşlenmemiş YAR 1 verisi 2 Bit BAQ Blok Boyutuna Bağlı Analiz.....	31
Şekil 4.7 : İşlenmemiş YAR 2 verisi 2 Bit BAQ Blok Boyutuna Bağlı Analiz.....	31
Şekil 4.8 : İşlenmemiş YAR 1 verisi I Bileşeni 2 Bit BAQ SNR=12.7 dB ...	32
Şekil 4.9 : İşlenmemiş YAR 1 I Bileşeni 2,3 ve 4 Bitle Kodlama durumunda SNR.....	33
Şekil 4.10 : İşlenmemiş YAR1 Q Bileşeni 2,3 ve 4 Bitle Kodlama durumunda SNR.....	33
Şekil 4.11 : İşlenmemiş YAR verisi I Bileşeni a)3 Bit BAQ SNR=20.14 dB b)4 Bit BAQ SNR=32.24 d B.....	33
Şekil 4.12 : İşlenmemiş YAR 2 I Bileşeni 2,3 ve 4 Bitle Kodlama durumunda SNR.....	34
Şekil 4.13 : İşlenmemiş YAR2 Q Bileşeni 2,3 ve 4 Bitle Kodlama durumunda SNR	34
Şekil 4.14 : BAQ 2,3 ve 4 Bit için Blok boyutuna bağlı sıkıştırma oranı.....	34
Şekil 4.15 : İşlenmiş Uydu YAR L.A. Görüntüsü BAQ Blok boyutuna bağlı SNR.....	35
Şekil 4.16 : İşlenmiş Gürültülü Hollanda YAR Görüntüsü BAQ Blok boyutuna bağlı SNR.....	35

Şekil 4.17 : İşlenmiş Gürültülü YAR Hollanda görüntüsü 2 Bit BAQ SNR=20.54 dB.....	36
Şekil 4.18 : İşlenmiş Uçak YAR Havaalanı Görüntüsü Blok Boyutuna Bağlı SNR.....	37
Şekil 4.19 : İşlenmiş Uçak Optik sensörden elde edilmiş Havaalanı Görüntüsü BAQ Blok boyutuna bağlı SNR.....	37
Şekil 4.20 : Lena için BAQ blok dağılımına bağlı SNR.....	38
Şekil 4.21 : 8x8 Blok boyutlu 2 Bit BAQ için a)Elde edilmiş Lena SNR=25.28 dB, b)Yapılan Hata (Orijinal görüntüden farkı).....	38
Şekil 4.22 : İşlenmemiş YAR 1 verisi I ve Q bileşenleri DCT-BAQ için Blok boyutuna bağlı SNR' ın değişimi.....	42
Şekil 4.23 : İşlenmemiş YAR 1 verisi I Bileşeni CR=4 a) DCT-BAQ SNR=20.08 dB b) JPEG SNR=13.4 dB.....	43
Şekil 4.24 : İşlenmemiş YAR 2 verisi I ve Q bileşenleri DCT-BAQ için SNR' ın blok boyutuna bağlı değişimi.....	43
Şekil 4.25 : DCT-BAQ için sıkıştırma oranının blok boyutuna bağlı değişimi.	44
Şekil 4.26 : İşlenmiş Uydu YAR L.A. ve İşlenmiş Uydu Gürültülü Hollanda verisinin DCT-BAQ performans Analizi.....	44
Şekil 4.27 : İşlenmiş Gürültülü Uydu YAR Hollanda Görüntüsü CR=3.55 a)DCT-BAQ SNR=23.1 dB b)JPEG SNR=24.4.....	45
Şekil 4.28 : İşlenmiş Uçak YAR ve Optik verilerinin DCT-BAQ Performans Analizi.....	45
Şekil 4.29 : Lena için DCT-BAQ Performans Analizi.....	45
Şekil 4.30 : Tüm örnek veriler için DCT-BAQ performans Analizi.....	46
Şekil 4.31 : İşlenmemiş YAR 1 I ve Q bileşenlerinin BAQ ve DCT-BAQ algoritmaları performans analizi.....	48
Şekil 4.32 : İşlenmemiş YAR 2 verisi için BAQ ve DCT-BAQ algoritmaları performans analizi.....	49
Şekil 4.33 : İşlenmiş Uydu YAR L.A için BAQ ve DCT-BAQ performans analizi.....	49
Şekil 4.34 : İşlenmiş Gürültülü Uydu YAR Hollanda için BAQ ve DCT- BAQ performans analizi.....	50
Şekil 4.35 : İşlenmiş Gürültülü Uydu YAR Hollanda için BAQ ve DCT- BAQ performans analizi.....	50
Şekil 4.36 : İşlenmiş Uçak Optik Havaalanı görüntüsü için BAQ ve DCT- BAQ performans analizi.....	51
Şekil 4.37 : Lena için BAQ ve DCT BAQ algoritmaları performans analizi....	51
Şekil 4.38 : BAQ ve DCT-BAQ algoritmaları için sıkıştırma oranını blok boyutuna bağlı değişimi.....	52

İŞLENMEMİŞ YAPAY AÇIKLIKLI RADAR VERİLERİNİN SIKIŞTIRILMASI

ÖZET

Yapay Açıklıklı Radar, yer yüzeyinin görüntülenmesini sağlayan uzaktan algılama sistemidir. YAR verilerinin işlenmesi sonucunda çok farklı uygulamalar için veriler elde edilir. Bununla birlikte YAR yardımıyla geniş alanların görüntülenmesi sonucu elde edilen veriler oldukça büyük miktardadır. Bu ölçüde büyük verilerin iletilmesi için gerekli olan link hızı çoğu zaman başarıl原因abilir. Bu yüzden işlenmemiş YAR verilerinin sıkıştırılması gerekir. Sadece işlenmemiş YAR verilerinin iletilmesi için değil, aynı zamanda bu verilerin uzaktan algılama yer istasyonlarında arşivlenmesi için de sıkıştırma algoritmalarına ihtiyaç duyulur.

YAR sensörü tarafından alınan analog işaret ilk olarak IF frekanslarına indirilir ve daha sonra da temel bantda aynı fazda (I) ve 90 derece faz farklı (Q) kanallarına ayrıştırılır, Nyquist frekansında örneklenir ve sayısallaştırılır. Sayısal işaret, işaret işlemenin yapılacağı yer istasyonuna gerçek zamanda iletilmelidir.

Aşağı link data hızı, darbe tekrarlama frekansı, alınan darbeler için örnekleme frekansı ve her bir örnek için kullanılan kod sözcüklerinin uzunluğuna bağlıdır. Sonuç olarak aşağı link data hızı oldukça yüksek gerçekleşir. Örnek olarak ERS-1 için bu hız 105 Mbit/s'dir.

Uçaklara yerleştirilmiş YAR sensörleri için aşağı link iletim band genişliği sınırlaması olmadığından alınan işaret 8 Bitle kuantalanır. Alınan işlenmemiş YAR verilerinin 8 Bitle kodlanması işlenmemiş YAR verilerinin tüm dinamik aralığının gösterilmesini sağlar. Data sıkıştırma uydu YAR sensörlerinde, uyduda geçici olarak verilerin saklanması, uydudan yer istasyonuna olan aşağı link data hızının azaltılması ve yer istasyonunda verilerin arşivleme amacıyla kaydedilmesi için önemli bir ihtiyaçtır.

İşlenmemiş YAR verilerinin sıkıştırılması için konumsal uzayda ve dönüşüm uzayında olmak üzere genellikle iki farklı kodlama tekniği kullanılır.

Konumsal uzayda en iyi bilinen sıkıştırma tekniği Blok Adaptif Kuantalamadır. İşlenmemiş YAR verilerinin istatistiksel özelliklerinden yararlanarak uygun sıkıştırma teknikleri geliştirilebilir. Blok Adaptif Kuantalama algoritması (BAQ) işlenmemiş YAR verisinin aynı fazda I ve 90 derece faz farklı Q bileşenlerinin istatistiksel özelliklerinden yararlanılarak gerçekleştirilir. İşlenmemiş YAR I ve Q bileşenleri gauss dağılımlı ve bilinmeyen varyanslıdır. Bu istatistiksel özelliklerden yararlanarak oluşturulacak kuantalayıcının eşikleri standart sapmaya bağlı olarak belirlenir. Her bir blok için bulunan standart sapma alıcı tarafta kod şemasının tekrar oluşturulması amacıyla 8 Bitle kodlanarak gönderilir. İletilen standart sapma değerlerine bağlı olarak her bir blok için kod şeması belirlenir. BAQ algoritması özetle işlenmemiş YAR verisini alt bloklara bölerek her bir blok için kuantalama hatasını minimize edecek şekilde verilerin olasılık yoğunluk fonksiyonları yardımıyla kodlanmasını sağlar

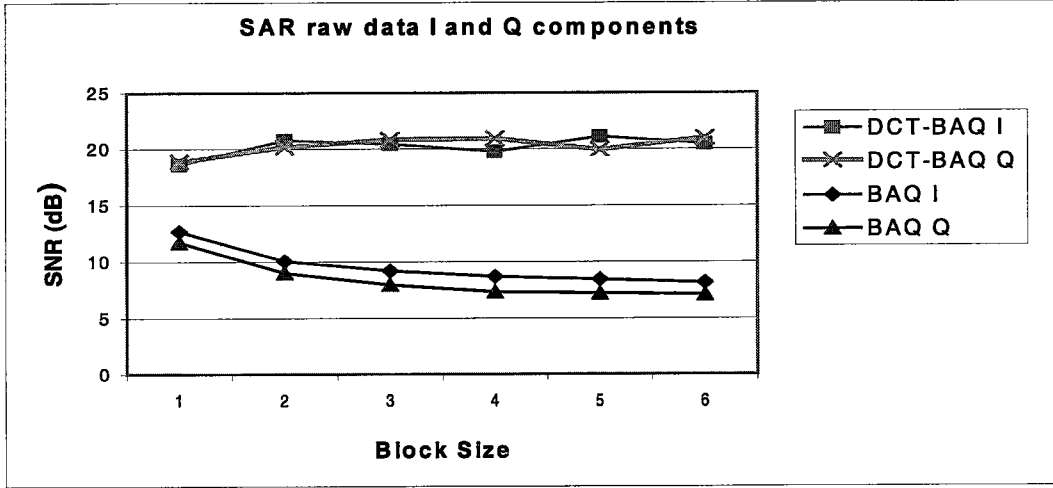
Bu kuantalama yaklaşımıyla az sayıdaki kuantalama seviyesi için kuantalayıcı çıkışında daha geniş bir dinamik aralık sağlanır. Sonuç olarak işaret gücünün dinamik aralığı tek blok için tüm dataya göre daha dar olur.

Bu algoritma için en önemli noktalardan biri blok boyutunun seçimidir. Blok boyutu, blok içinde gauss dağılımını sağlaması için yeteri kadar büyük olmalıdır. Ayrıca işaret gücünün blok içinde menzile zayıflaması etkilerinden kaçınmak için ise yeterince küçük seçilmelidir.

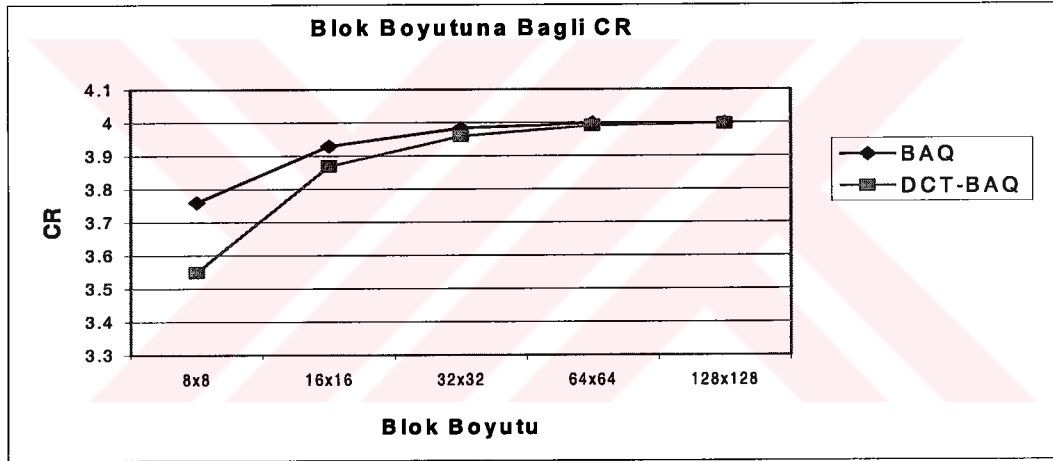
Sıkıştırma algoritmasının ilişkisizlendirme aşaması dönüşüm domeninde gerçekleştirilir. Etkin bir veri sıkıştırma işlemi için dönüşümün bazı özellikleri aşağıdaki gibi olmalıdır.

- Ortogonal serilere açılım yardımıyla elde edilen dönüşüm katsayıları birbirleriyle ilişkisiz olmalıdır.
- Dönüşüm uzayında katsayıların enerjileri belli bir bölgede yoğunlaşmalıdır.
- Hızlı bilgisayar algoritmalarına sahip lineer dönüşümler kullanılmalıdır.

Dönüşüm domeninde dönüşüm katsayıları gauss dağılımına uyuyorsa, dönüşüm sonrası BAQ uygulanabilir. Bu çalışmada dönüşüm olarak Ayrık Kosinüs Dönüşümü seçilmiştir. Hızlı algoritması ve reel katsayıları DCT ile işlemi cazip kılar. Bununla birlikte, DCT dönüşüm domeninde enerji yoğunlaşmasını gayet başarılı bir şekilde gerçekleştirir. İşlenmemiş YAR verisi için dönüşüm katsayılarını dinamik aralığı oldukça yüksek olduğu görülür. Dönüşüm sonrası BAQ uygulamak için dinamik aralığın azaltılması gerekir. Dinamik aralığın azaltılması için bu tezde her bir blok için DCT katsayılarının DC bileşenlerinin standart sapmayla birlikte 8 Bitle ayrı kodlanması yöntemi geliştirilmiştir. DC bileşen yerine sıfır eklenerek oluşan dinamik aralığı azaltılmış yeni veri kümesi BAQ yardımıyla kodlanmıştır. Karşı tarafta yeniden elde edilen veri kümelerinde sıfır eklenen datanın yerine DC bileşen yerleştirilir ve ters dönüşüm yardımıyla işlenmemiş YAR verisi alıcı tarafta tekrar elde edilir. Dönüşüm domeninde DCT-BAQ ile BAQ'ya göre yaklaşık 10 dB'lik daha iyi işaret gürültü oranı elde edilmiştir. Şekil 1'de İşlenmemiş YAR verisi I ve Q Bileşenleri için BAQ ve DCT-BAQ algoritmaları SNR performans analizi blok boyutuna bağlı gösterilmiştir. Şekil 2' de ise blok boyutuna bağlı sıkıştırma oranının değişimi gösterilmiştir.



Şekil 1 İşlenmemiş YAR verisi I ve Q Bileşenleri için
Blok Boyutuna Bağlı SNR Performansı



Şekil 2 Blok Boyutuna Bağlı Sıkıştırma Oranı (CR)

SAR RAW DATA COMPRESSION

SUMMARY

Synthetic Aperture Radar, (SAR) is a remote sensing system, which is used for earth observation. There are several applications of SAR data is used for observation systems. The developments of advanced SAR systems for earth observation from satellite, offers flexible microwave sensors, capable of multiple imaging modes and resolutions. These features together with the wide swath requirements, determines the production of a huge amount of data. The amount of produced data is constrained by storage and downlink bandwidth. Additional we need to use SAR raw data compression not only the down link requirements but also at the remote sensing ground stations for archiving and transmitting raw data to the end users. To manage this problem, many efforts have been taken place in order to develop suitable compression techniques for raw data coding.

The analog signal received by SAR sensor is generally treated as follows. First it is heterodyned to an intermediate frequency, then it is demodulated to baseband in-phase (I) and in-quadrature (Q) channels, sampled according to the Nyquist rate and finally it digitized. The digital signal may be stored onboard and eventually downlinked to the Earth station where processing is made. For continuous operations the downlink must be in real time

The data rate is proportional to the pulse repetition frequency, to the sampling rate for each received pulse, and to the number of quantization bits for each sample. The resulting rate is generally very high. For example 105 Mbit/s for ERS-1 mission.

For airborne SAR sensors, which do not have to meet restrictions imposed by downlink transmission bandwidths, often quantize the received signal to 8 Bit. This is sufficient to represent the full dynamic range of SAR raw data.

Data compression is an essential requirement for spaceborn SAR sensors to reduce the data volume that must be temporarily stored onboard, transmitted from satellite to ground station, and then stored on the ground.

For SAR raw data compression, two encoding approaches are generally used. Direct encoding in spatial domain and transform encoding in transform domain.

A well-known spatial domain compression technique is based on Block Adaptive Quantization (BAQ). The BAQ algorithm is based on the observation that SAR raw data can be modeled as a Gaussian distributed random variable with a slowly varying standard deviation value. BAQ encoding is implemented by dividing the SAR raw data set into data blocks of small size and by estimating, for each block, the standard deviation value. Following this step each sample within the block is normalized to the estimated standard deviation value. Finally, the normalized sample is compared to the optimum quantization levels of an n bit quantizer with unitary standard deviation. The quantized samples and the standard deviation of each block are transmitted to allow data decoding. The chosen number n bits depend on the system data rate constraints.

This quantization approach achieves, for the small number of quantization levels, a wider dynamic range at the quantizer output than simple truncating of data. The reason is that dynamic range of the signal power for a single block is much less than that of the overall data

A delicate point of the algorithm is represented by the proper selection of block size dimension. Block size can not be too small to guarantee the Gaussian statistics within the block, but not too large to avoid the effect of the antenna pattern modulation and of the range attenuation of the signal power within the block.

Using Transform domain for data compression is the decorrelation stage of the compression algorithm. For efficient data compression process the transform should be:

- Orthogonal series expansions as it provide series of coefficients, which are uncorrelated.
- Unitary as it preserves the signal energy and pack most of the information into a smallest number of them
- Linear as it offers a fast computer algorithm.

In this study we chose Discrete Cosine Transform (DCT) for SAR raw data compression. DCT is a fast transform, requires real operations, near optimal substitute for the Karhunen –Loeve (KL) transform for highly correlated images. Low complexity arithmetic and has very good energy compaction.

In transform domain if the transform coefficients are Gaussian distributed, BAQ can be used for data quantization. Transform domain, coefficients have high dynamic range, because of this the performans of the compression is reduced. For reduction of dynamic range we can code DC component of the transform coefficients with 8 Bits for each block and transmit standard deviation and DC component separately. We code DC component separately and rest of the transform coefficients are Gaussian distributed. BAQ is used for rest of the transform coefficients. In Figure 1 we can see SNR performance of both BAQ and DCT-BAQ algorithm for I and Q components of SAR raw data versus block size. Compression ratio versus block size is shown for both algorithms in Figure 2.

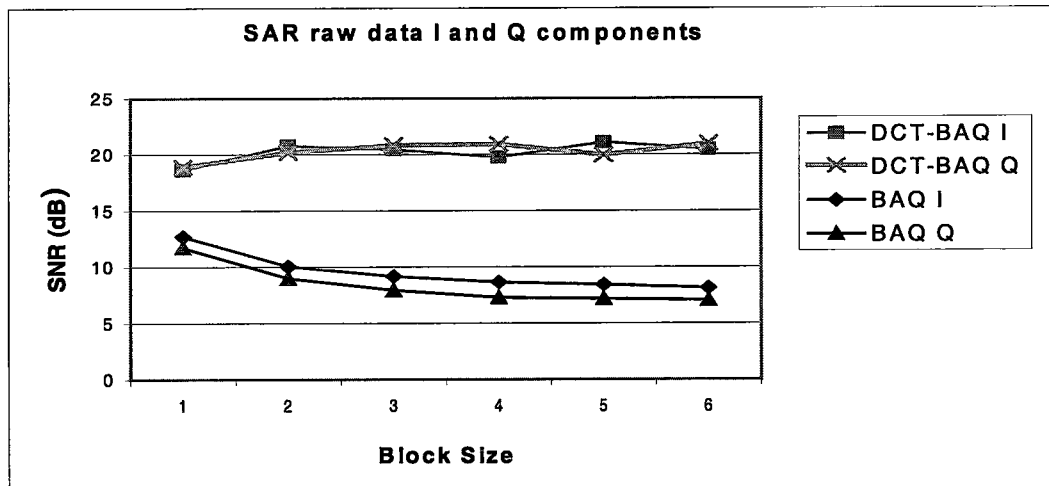


Figure 1 SNR performance of both BAQ and DCT-BAQ algorithm for I and Q components of SAR raw data versus block size.

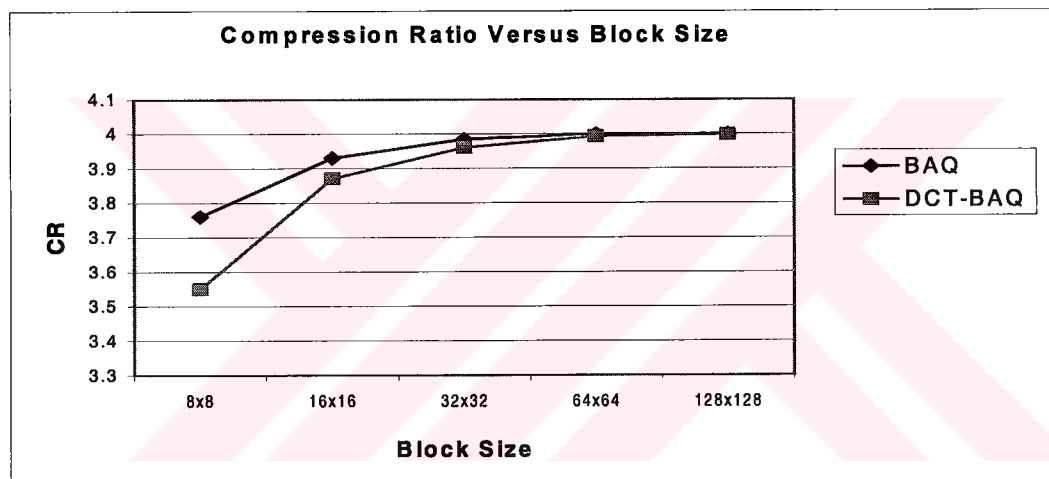


Figure 2 Compression ratio versus block size

1. GİRİŞ

Yapay Açıklıklı Radar (YAR) verileri günümüzde oldukça yaygın bir şekilde kullanılmaktadır. Yerleşim Alanlarının gözlenmesi ve planlaması, tarımda ürün rekoltesinin tesbiti, arazi yapısının, orman alanlarının, maden kaynaklarının belirlenmesi, doğal afetler için risk yönetimi ve analizi, deniz kirliliğinin tesbiti gibi uygulamalar YAR verilerinin uygun biçimde işlenmesiyle gerçekleştirilir. YAR verileri için önemli özelliklerden biri de YAR'ın ayırcılığıdır. YAR için ayırcılık arttığında sensörden yer istasyonlarına iletilmesi gereken işlenmemiş YAR veri miktarıda artar. İşlenmemiş YAR verisinin yer istasyonuna iletilmesi sınırlı band genişliği sebebiyle veri sıkıştırmasını zorunlu kılar. YAR sensörü tarafından aydınlatılan alandan yansıyan işaret alınır, alınan analog işaret ara frekans bölgesine indirilir ve temel bandda eş fazlı I ve 90 derece faz farklı Q bileşenleri elde edilir. İşlenmemiş YAR verisi Nyquist frekansında genellikle 3 bit ile 8 bit arası kodsözcükleri ile kodlanarak sayısallaştırılır. YAR sensörü ile YAR verilerinin işlendiği yer istasyonları arasındaki aşağı link gerçek zamanda veri iletimini sağlamalıdır.

İşlenmemiş YAR verileri için veri hızı darbe tekrarlama frekansı, her alınan darbeye, uygulanan örnekleme frekansı ve YAR verilerinin kaç bitle kodlandığına bağlıdır. Bundan dolayı aşağı link veri hızı oldukça yüksektir. Örnek olarak ERS-1 ve ERS-2 için bu hız 105 Mb/sn'dir.(Evert and Attema, 1991, [2]) Geleneksel Uçak YAR için ise aşağı link band genişliği sınırlaması olmadığından YAR verisi 8 Bitle kodlanır. Bu durumda YAR verisi tüm dinamik aralığıyla temsil edilebilmektedir.

Uydu YAR sensörleri için veri sıkıştırma, YAR'da geçici olarak sensörün bulunduğu platformda kaydedilen data miktarını azaltmak, uydu yer istasyonu aşağı linki için ihtiyaç duyulan gerekli veri hızını azaltmak ve yer istasyonunda YAR verilerinin arşivlenmesi ve son kullanıcılara iletilmesinde veri hızını düşürmek amacıyla gereklidir. Veri sıkıştırma beraberinde bilgi kaybına neden olur, bunun sonucunda da görüntü kalitesi bozulabilir. Kullanıcıların radar görüntülerini hangi amaçla

kullanacaklarının bilinmesi durumunda kullanıcıların ihtiyaç duydukları görüntü kalitesi sağlanabilir.

İşlenmemiş YAR verisi direk konumsal uzayda ve dönüşüm uzayında olmak üzere genellikle iki tür kodlama yapısı kullanılarak sıkıştırılabilir. En iyi bilinen konumsal uzay sıkıştırma tekniği Blok Adaptif Kuantalamaya dayanır ve bu sıkıştırma algoritması Magellan için uygulanmıştır.(Kwok ve Johnson,1989, [9]). Blok Adaptif Kuantalama (Block Adaptive Quantization, BAQ) İşlenmemiş YAR verisini küçük değişimli standart sapmalı rasgele Gauss dağılımlı bir işaret olarak modeller.(Zeoli, 1976, [1]). BAQ İşlenmemiş YAR verisini küçük boyutlu bloklara böler ve her bir blok için standart sapmayı belirler. Veri bloğunun standart sapmasına bağlı olarak kuantalayıcı eşikleri belirlenir. BAQ için önemli konulardan biri de blok boyutunun seçimidir. Blok boyutu, blok içindeki Gauss dağılımını garanti edecek biçimde büyük, aynı zamanda anten pateni modülasyonu ve işaret gücünün menzile zayıflaması etkilerini göstermemesi için ise yeterince küçük seçilmelidir. Daha iyi performans kuantalayıcının vektörel seçilmesiyle sağlanır. Bu durumda ilk adım vektörel kuantalama (Vectorel Quantization, VQ) için uygun kod kitabının hazırlanmasıdır. Daha sonra her giriş vektörü için vektörel kodlayıcı minimum vektör hatası için kod kitabını araştırır. BAQ ve VQ kuantalayıcının birlikte kullanılmasıyla işlenmemiş YAR verisinin daha iyi sıkıştırılması sağlanır. Bu algoritma ise Blok adaptif vektör kuantalayıcı (Block Adaptive Vector Quantization, BAVQ) adını alır. (Benz, 1995, [4]) İlk önce veri BAQ ile sıkıştırılır daha sonra ise sıkıştırılmış veriye vektörel kuantalama uygulanır. BAVQ için kodaçma işleminde ise ilk olarak VQ ile daha sonra ise BAQ ile kod sözcükleri elde edilir.

İşlenmemiş YAR verileri için en düşük veri hızını, işaret kodlayıcı (Signal Coding, SC) sağlar. SC' de her bir örnek için 1 bit atanır. Steinberg, 1987, [3] ve Franceschetti 1991, [5] ve 1999, [6] tarafından göstermiştir ki her bir örnek için daha az sayıda bit atanmasının işaret gürültü oranını (Signal Noise Ratio, SNR) ve görüntü kalitesini önemli ölçüde korur.

Dönüşüm bazlı sıkıştırma tekniklerinin İşlenmemiş YAR verilerine uygulanması lineer dönüşümler için gerçekçidir. Dönüşüm temelli kodlama problemi için verilen bir rasgele vektörün temel vektörler cinsinden ifadesinde ortalama karesel hatanın (Mean Square Error, MSE) minimize edilmesi istenir. Karhunen-Loeve dönüşümü

(Karhunen-Loeve Transformation, KLT) ortalama karesel hatayı minimum yapan dönüşümdür. (Lim, 1990, [7]), fakat işlem yoğunluğu oldukça fazladır. Ayrık cosinüs dönüşümü (DCT) oldukça sık kullanılır çünkü KL dönüşüme en iyi yaklaşımı sağlar ve hızlı algoritmalarla etkin uygulamalara sahiptir.

Bu çalışmada Bölüm 2’de Yapay Açıklıklı Radar (YAR) sistemi tanıtılmıştır. YAR için temel görüntüleme geometrisi açıklanmış, enine ve menzil doğrultusunda radarın ayırcılığı belirlenmiştir. Radar tarafından aydınlatılmış bölgeden yansıyan işaretler radarın bulunduğu platformda temel banda indirgenerek sayısallaştırılmış ve İşlenmemiş YAR verisi için aynı fazda I ve 90 derece faz farklı Q bileşenlerinin matematiksel ifadesi oluşturulmuştur. Daha sonra YAR ile yer istasyonu arasındaki aşağı link hızı hesaplanmıştır. İşlenmemiş YAR verisinin istatistiksel özellikleri ortaya konmuştur. Son olarak ise İşlenmemiş YAR verisinin sıkıştırma modeli en genel anlamda açıklanmıştır.

Bölüm 3’te İşlenmemiş YAR verileri için sıkıştırma algoritmaları tanıtılmıştır.. İlk olarak İşlenmemiş YAR verileri için standart kabul edilen BAQ algoritması tanıtılmıştır. Daha sonra dönüşüm domeninde sıkıştırma algoritmaları tanıtılmıştır. Optik görüntüler için kullanılan sıkıştırma algoritmalarının neden İşlenmemiş YAR verisi için kullanılmadığı açıklanmıştır. Bu amaçla JPEG algoritması tanıtılmıştır. Dönüşüm algoritması olarak DCT kullanılarak dönüşüm domeninde işlenmemiş YAR verilerinin sıkıştırılması anlatılmıştır.

Bölüm 4’te İşlenmemiş YAR verisi için sıkıştırma teknikleri performans analizi yapılmıştır. İlk olarak performans kriterleri ve kullanılan örnek veriler tanıtılmıştır. Zaman domeninde BAQ ve dönüşüm domeninde ise DCT-BAQ algoritmaları için çeşitli örnek veriler için başarımlar incelenmiştir.

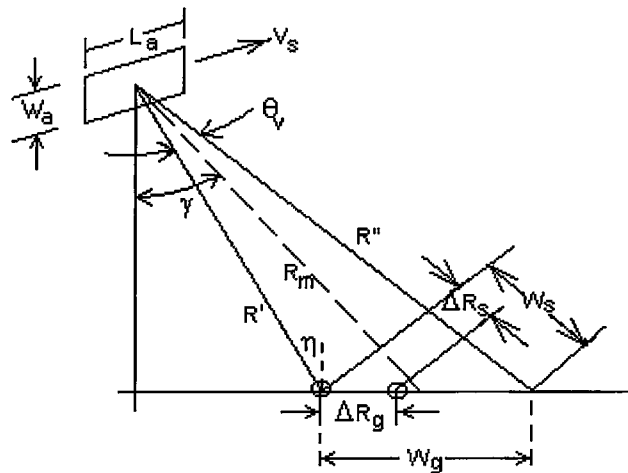
Bölüm 5’de ise bu çalışmada elde edilen sonuçlar belirtilmiş ve öneriler açıklanmıştır.

2. YAPAY AÇIKLIKLI RADAR SİSTEMİ

Bu bölümde yapay açıklıklı radar sistemi tanıtılmıştır. 1978’ de ilk yapay açıklıklı radar sensörü taşıyan Seasat uydusundan sonra uzaktan algılama uydularının kullanılması yaygınlaşmıştır. Yapay açıklıklı radarlar yeryüzeyindeki geniş alanların görüntülenmesinde kullanılır. Yapay Açıklıklı Radar verilerinin işlenmesiyle birlikte yerleşim alanlarının planlaması, tarımda ürün miktarının tesbiti, doğal afetlerin belirlenmesi ve analizi, çevre kirliliği etkilerinin belirlenmesi, maden tesbiti gibi bir çok konuda uygulamaları vardır. Yapay açıklık, mevcut fiziksel bir açıklığın sabit bir hızla hareket eden bir platformda bulunmasıyla elde edilir. Bu fiziksel platform uydu veya uçak olabilmektedir. Sensörden iletilen darbeler hedefe çarpar ve yansır. Hedeften saçılan işaret hedefin saçıcılık özelliği konusunda bilgi taşır. [8,12]

2.1 Yapay Açıklıklı Radar Görüntüleme Geometrisi

Yapay açıklıklı radar sisteminde iki ayrı nesne birbirlerinden, menzil bilgisini oluşturan zaman gecikmeleri arasındaki fark ve azimuth bilgisini oluşturan doppler geçmişleriyle ayrılırlar. Şekil 2.1’de yandan bakışlı radar için süpürme genişliğini ve menzil ayırıcılığını gösteren temel geometri gösterilmiştir.



Şekil 2.1 Menzil yönünde radar görüntüleme geometrisi

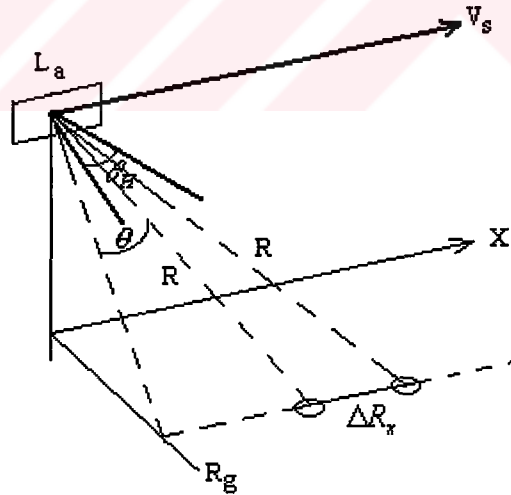
Bu geometri için bakış açısı γ çoğu kez radar hüzmesi ile görüntülenen yüzeyin normali arasındaki açı olan η ile eşit kabul edilebilir. Yapay açıklıklı radar için önemli özelliklerden biri de yüzeydeki iki noktayı birbirinden ayırdedebilme kabiliyetidir. Bu ayırdedebilme hem menzil hem de azimuth yönünde olmaktadır. Radar anteninin dikey yöndeki hüzme genişliği $\theta_v = \lambda/W_a$ ile bulunur. Böylece süpürme genişliği

$$W_g \approx \frac{\lambda R_m}{W_a \cos \eta} \quad (2.1)$$

olarak verilir. Radarın menzil doğrultusundaki ayırıcılığı ise Şekil 2.1' de ΔR_g ile gösterilmiştir. Eğer radar darbesinin süresi τ_p ise iki noktanın belirlenmesi için minimum ayırıcılık

$$\Delta R_g = \frac{\Delta R_s}{\sin \eta} = \frac{c \tau_p}{2 \sin \eta} \quad (2.2)$$

şeklindedir. Burada ΔR_s eğim menzil ayırıcılığı, c ise ışık hızıdır. Şekil 2.2'de ise antenin hareket yönüne paralel iki noktanın birbirinden ayırılmasına ilişkin geometri görülmektedir.



Şekil 2.2 Azimuth yönünde YAR Görüntüleme Geometrisi

Antenin uçuş doğrultusundaki uzunluğu L_a ile gösterilirse anten hüzmesinin enine genişliği $\theta_H = \lambda/L_a$ ile bulunur. Bu enine hüzme genişliği gerçek açıklık için azimuth yani enine ayırıcılığını belirler.

$$\Delta R_x = R\theta_H = \frac{R\lambda}{L_a} \quad (2.3)$$

Yapay Açıklıklı Radar için ise sensörün hareketinden kaynaklanan Doppler kayması gönderilen dalganın frekansına ve hareket doğrultusundaki nokta hedefin konumuna

$$f_D = \frac{2(V_{st} \sin \theta)}{\lambda} \approx \frac{2V_{st}x}{\lambda \cdot R} \quad (2.4)$$

olarak bağlıdır. (2.4) ifadesinde V_{st} sensörün hedefe göre hızı, θ hüzmelin maksimum doğrultusunda hedefi görmesi durumunda referansa göre açığı göstermektedir. Enine doğrultudaki ayırıcılığı belirlemek için kullanılan ilk hedefin uçuş yönündeki koordinatı referansa göre x_1 dopler kaymasına bağlı olarak bulunur.

$$x_1 = \frac{\lambda R f_{D_1}}{2V_{st}} \quad (2.5)$$

Benzer şekilde aynı menzilde bulunan diğer hedefin koordinatı (x_2) için dopler kayması f_{D_2} olur. Böylelikle eşit menzilde ve aynı hüzmeye içerisinde aynı anda bulunan hedefler, dönen işaretlerin dopler frekans spektrumunu analizi yardımıyla birbirinden ayırdedilebilir. Burdan hareketle yapay açıklıklı radar için azimuth ayırıcılığı;

$$\Delta R_x = \left(\frac{R\lambda}{2V_{st}} \right) \delta f_D \quad (2.6)$$

olarak belirlenir. Frekans domenindeki ayırıcılık zaman domeninde hedeflerin hüzmeye içinde kalma süresinin (T) tersi olarak alınabilir. $\delta f_D = 1/T$. Bu süre hüzmeye içindeki belirli bir hedefin hüzmeye kalma süresidir.

$$T = \frac{R\theta_H}{V_{st}} = \frac{R\lambda}{L_a V_{st}} \quad (2.7)$$

(2.7) ifadesi (2.6)' da yerine konursa;

$$\Delta R_x = \left(\frac{\lambda R}{2V_{st}} \right) \left(\frac{L_a V_{st}}{R\lambda} \right) = \frac{L_a}{2} \quad (2.8)$$

(2.8) ifadesiyle YAR için azimuth ayırıcılığı bulunmuş olur.[8] Aslında bu değer azimuth ayırıcılığı için yaklaşılabilecek en iyi değerdir.

Sürekli dalga darbesi gönderen bir YAR sensörünün ilettiği işaret

$$s(t) = m(t) \cos(2\pi f_0 t) \quad (2.9)$$

olarak verilir. Bu ifadede $m(t)$ τ_m sürelidir ve $s(t)$ 'nin zarfıdır.

$$m(t) = \frac{1}{2} \left(1 + \cos\left(\frac{2\pi}{\tau_m} t\right) \right) \quad (2.10)$$

şeklindedir. Radar sensörü tarafından Şekil 2.1'de gösterilen bir nokta hedef için alınacak işaret

$$S_r(t) = K \sigma \cos\left(2\pi f_0 \left(t - \frac{2R_m}{c} - \frac{2w_s}{c}\right)\right) \cdot m\left(t - \frac{2R_m}{c} - \frac{2w_s}{c}\right) \quad (2.11)$$

eşitliği ile belirlenir. Burada cos lü terim üstel olarak yazılırsa

$$S_r(t) = K \cdot \text{Re} \left\{ \sigma \cdot \exp\left(j2\pi f_0 \left(t - \frac{2R_m}{c} - \frac{2w_s}{c}\right)\right) \cdot m\left(t - \frac{2R_m}{c} - \frac{2w_s}{c}\right) \right\} \quad (2.12)$$

elde edilir. YAR sensörünün aydınlattığı alan içindeki saçıcılardan dönen toplam işaret ise

$$S_r(t) = K \cdot \text{Re} \left\{ \int_{-w_{s1}}^{w_{s1}} \sigma \cdot \exp\left(j2\pi f_0 \left(t - \frac{2R_m}{c} - \frac{2w_s}{c}\right)\right) \cdot m\left(t - \frac{2R_m}{c} - \frac{2w_s}{c}\right) dw_s \right\} \quad (2.13)$$

şeklinde hesaplanır. Bu denklemde σ , görüntülenen bölgedeki kompleks yansıtıcılık fonksiyonudur. Sensöre gelen toplam işaret için taşıyıcı frekansının etkilerini gidermek ve temelband işaretini almak için (2.13) ifadesi f_0 frekanslı işaretin eş fazlı I ve 90 derece faz farklı Q bileşeni ile çarpılır ve çarpıcı çıkışında alçak geçiren filtreden geçirilir. Böylece karıştırılmış işaretin eş fazlı, I bileşeni

$$S_r(t) = \frac{K}{2} \cdot \text{Re} \left\{ \int_{-w_{s1}}^{w_{s1}} \sigma \cdot \exp\left(j2\pi f_0 \left(2t - \frac{4R_m}{c} - \frac{2w_s}{c}\right)\right) m\left(t - \frac{2R_m}{c} - \frac{2w_s}{c}\right) dw_s \right\} +$$

$$\frac{K}{2} \cdot \text{Re} \left\{ \int_{-w_{s1}}^{w_{s1}} \sigma \cdot \exp\left(-j\frac{4\pi f_0 w_s}{c}\right) m\left(t - \frac{2R_m}{c} - \frac{2w_s}{c}\right) dw_s \right\} \quad (2.14)$$

olarak elde edilir. Bu işaret temel bantdaki işarettir ve alçak geçiren filtreden geçirildiğinde

$$S_I(t) = \frac{K}{2} \cdot \text{Re} \left\{ \int_{-w_{s1}}^{w_{s1}} \sigma \cdot \exp(-j \frac{4\pi f_0 w_s}{c}) \cdot m \left(t - \frac{2R_m}{c} - \frac{2w_s}{c} \right) dw_s \right\} \quad (2.15)$$

olarak bulunur. Benzer şekilde (2.14) denklemi 90 derece faz farklı ile çarpılıp alçak geçiren filtreden geçirilirse

$$S_Q(t) = \frac{K}{2} \cdot \text{Im} \left\{ \int_{-w_{s1}}^{w_{s1}} \sigma \cdot \exp(-j \frac{4\pi f_0 w_s}{c}) \cdot m \left(t - \frac{2R_m}{c} - \frac{2w_s}{c} \right) dw_s \right\} \quad (2.16)$$

ifadesi bulunur. Uydu platformunda demodülatör çıkışında ise kompleks işaret

$$S = S_I + jS_Q \quad (2.17)$$

biçiminde gösterilebilir.[12]

YAR'da sensör aydınlatılmış bölgeden yansıyan darbeleri alır. Alınan işaret, analog-sayısal dönüştürücüden geçirilerek örneklenir. İşaretin cosinüs bileşeni I, sinüs bileşeni Q olarak adlandırılır ve aşağı link uzaktan algılama yer istasyonlarına gönderilir. ERS 1 sistemi için I ve Q bileşenleri 5 bit/örnek ile kodlanır ve 104Mbit/sn hızıyla uzaktan algılama yer istasyonuna gönderilir. Uzaktan algılama sistemlerini sınırlayan en önemli konuların başında işlenmemiş YAR verisinin gönderilmesi için gereken bu yüksek veri hızı gelmektedir. Bu yüksek veri hızını azaltmak için İşlenmemiş YAR verilerinin sıkıştırılmasına ihtiyaç vardır. Aşağı link veri hızı YAR sistemlerinin performansını sınırlandırır.

YAR için eğim menzıl genişliği

$$W_s \approx W_g \cdot \sin \eta \quad (2.18)$$

olarak verilir. Veri örnekleme pencere süresi

$$\tau_w \approx \frac{2W_s}{c} \quad (2.19)$$

şeklindedir. Ani data hızı

$$r_i = n_b f_{sr} \quad (2.20)$$

olarak hesaplanır. Burada n_b örneklerin kodlanacağı kod sözcüklerinin uzunluğu Doppler f_{sr} band genişliği

$$B_D = \frac{2V_{st}}{L_a} \quad (2.21)$$

şeklindedir. Darbeler arası süre

$$T_p = \frac{1}{f_p} \quad (2.22)$$

olarak verilir. YAR için gerçek zamanlı aşağı link veri (İşlenmemiş YAR verisi) hızı

$$r_{DL} = \frac{\tau_w \cdot r_l}{T_p} \quad (2.23)$$

eşitliği ile hesaplanır.[1,8]

2.2 Yapay Açıklıklı Radar Datası İstatistiksel Özellikleri

Yapay açıklıklı radar verisini belirleyen, yüzeyin saçıcılığı ve yüzeydeki saçıcı yoğunluğu, sadece istatistiksel parametrelerle açıklanabilir. Bundan dolayı saçılan alan (İşlenmemiş yapay açıklıklı radar işareti) bir rasgele süreçtir.

YAR ayırıcılık hücresi elektromanyetik dalganın dalga boyu ile karşılaştırıldığı zaman oldukça geniştir. Bununla birlikte her ayırıcılık hücresinde yüzeyin yapısı ve saçıcılık yoğunluğunun dağılımına bağlı olarak çok sayıda saçıcı bulunmaktadır. Yansıyan işaret her bir saçıcı yansımalarının tutarlı toplamının sonucudur. Tek bir saçıcıdan dönen işaretin fazı, sensör ile saçıcı arasındaki mesafe, karşılıklı yönelme, ve saçıcının elektromanyetik özellikleriyle ilgilidir. Hareketli sistemler için dönen işaretin fazındaki değişiklik zamana ve alınan işarettaki farklılıklara bağlı olarak değişir.

Eğer ayırıcılık hücresinde N tane saçıcının olduğunu ve baskın bir hedefin olmadığını kabul edersek, toplam saçılan işaret

$$V_1 + jV_2 = \sum_{i=1}^N V_i \exp(j\phi_i) = \sum_{i=1}^N V_i \cos \phi_i + j \sum_{i=1}^N V_i \sin \phi_i \quad (2.24)$$

şeklinde ifade edilebilir. Burada $V_i \exp(j\phi_i)$ i. saçıcının katkısıdır. V_1 ve V_2 ise sırasıyla toplam işaretin reel ve imajiner bölümüdür. Eğer saçıcı sayısı N büyükse

merkezi limit teoremi uygulanabilir ve bu durumda V_1 ve V_2 normal dağılımlıdır. Buna bağlı olarak V_1 ve V_2 ' ye ilişkin olasılık dağılım fonksiyonları sıfır ortalamalı ve varyansı σ^2 olmak üzere

$$\begin{aligned} p(V_1) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{V_1^2}{2\sigma^2}\right) \\ p(V_2) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{V_2^2}{2\sigma^2}\right) \end{aligned} \quad (2.25)$$

şeklinde belirtilebilir.[17]. V_1 ve V_2 'yi ilişkisiz kabul edersek

$$E[V_1 V_2] = E[V_1] E[V_2] = 0 \quad (2.26)$$

olarak bulunur. Birleşik olasılık yoğunluk fonksiyonu

$$p(V_1, V_2) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{V_1^2 + V_2^2}{2\sigma^2}\right) \quad (2.27)$$

şeklindedir. Polar koordinatlar için;

$$p(V_1, V_2) dV_1 dV_2 = p(V_1, V_2) V dV d\phi = p(V, \phi) dV d\phi \quad (2.28)$$

(2.26) ve (2.27) ifadeleri karşılaştırılarak ve $p(V, \phi)$ için çözülerek (2.2.6) ifadesi

$$p(V, \phi) = \frac{V}{2\pi\sigma^2} \exp\left(-\frac{V^2}{2\sigma^2}\right) \quad (2.29)$$

olarak bulunur. $V_1^2 + V_2^2 = V^2$ alınır

$$p(\phi) = \int_0^\infty dV p(V, \phi) = \frac{1}{2\pi} \quad (2.30)$$

$$p(V) = \int_0^{2\pi} d\phi p(V, \phi) = \frac{V}{\sigma^2} \exp\left(-\frac{V^2}{2\sigma^2}\right), V \geq 0 \quad (2.31)$$

eşitlikleri elde edilir. (2.30) ve (2.31) ifadelerinde gösterildiği üzere işaretin genliği, V , $(0, \infty)$ aralığında Rayleigh dağılımlı ve fazı, ϕ , ise $(0, 2\pi)$ arasında uniform dağılımlıdır.[4,12]

$$E[V] = \int_0^{\infty} dV p(V) V = \sqrt{\frac{\pi}{2}} \sigma \quad (2.32)$$

$$E[V^2] = \int_0^{\infty} dV p(V) V^2 = 2\sigma^2 \quad (2.33)$$

İşaretin genliğinin standart sapması ise

$$E\{[V - E[V]]^2\} = E[V^2] - \{E[V]\}^2 = \frac{4 - \pi}{2} \sigma^2 \quad (2.34)$$

(2.34) varyans ifadesinin karakökü alınarak elde edilir. Bu ifade işaretin ortalama civarındaki değişimini tahmin etmemizi sağlar. Güç tesbiti için $W=V^2$ alınır. Böylece

$$p(V)dV = p(W)dW = p(W)2VdV \quad (2.35)$$

ifadesi bulunur. Bu durumda güç için olasılık yoğunluk fonksiyonu

$$p(W) = \frac{1}{2\sigma^2} \exp\left(-\frac{W}{2\sigma^2}\right) \quad (2.36)$$

şeklinde bulunur. Buna bağlı olarak üstel dağılımlı W için beklenen değer ve karesel beklenen değer

$$E[W] = 2\sigma^2 \quad (2.37)$$

$$E[W^2] = 8\sigma^4 \quad (2.38)$$

olarak ve varyans ise

$$E\{[W - E[W]]^2\} = E[W^2] - \{E[W]\}^2 = 4\sigma^2 \quad (2.39)$$

ifadesiyle belirlenir. Bu eşitlik işaret gücünün ortalama civarındaki değişimini gösterir.[12]

2.3 İşlenmemiş YAR Verisi Sıkıştırma Modeli

YAR sisteminde uydu ile yer istasyonu arasındaki aşağı link veri hızı sistemi sınırlayan önemli faktörlerden biridir. Veri sıkıştırmanın amacı gereksiz veya az

önemli verileri atarak gerekli bilginin mümkün olan en az sayıda bitle kodlanarak iletilmesidir.

YAR sensörünün bulunduğu platformda veri sıkıştırma uygulamaları genel olarak

- Geçerli band genişliği ile mümkün olduğunca çok verinin iletimi,
- Uydu ile yer istasyonu arasındaki gerekli veri hızının düşürülmesi,
- Veri kayıt ihtiyacının, sensörün bulunduğu platformda ve yer istasyonunda azaltılması

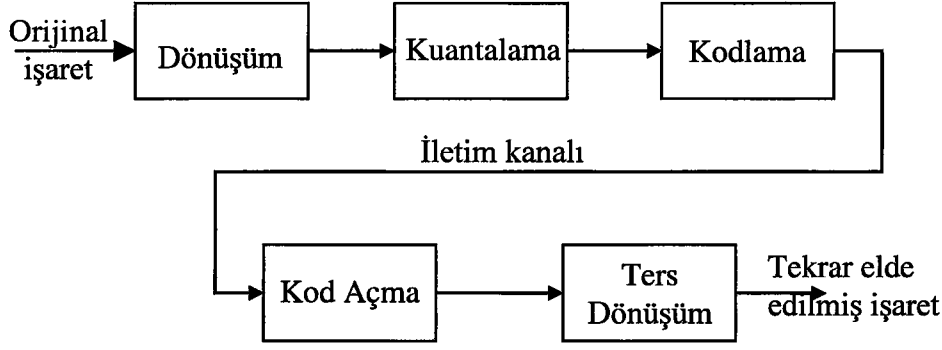
amacıyla yapılır.

Eğer tüm sistemin verimliliğini arttıracaksa aşağı link veri hızını düşüren veri sıkıştırma algoritmaları yer istasyonu ile son kullanıcılar arasındaki link yükünü azaltmak için de kullanılabilir. Aşağı link veri hızını azaltan algoritmalar sensörün bulunduğu platformda gerçekleştirileceği için bu platformdaki uydu kaynaklarıyla sınırlıdır. Bu yüzden sıkıştırma algoritmalarını gerçekleştirecek donanımın olabildiğince basit olması gerekir. Bununla birlikte yer istasyonu ile son kullanıcılar arasındaki link yükünü azaltacak sıkıştırma algoritmaları ihtiyacı karşılayacak ölçüde karmaşık olabilir.

Veri sıkıştırma genel olarak kayıplı ve kayısız sıkıştırma olarak ayrılır. Kayıpsız sıkıştırma tekniklerinden yeniden elde edilen sıkıştırılmış veri orijinal veriye göre herhangi bir kayıp olmadan elde edilebilir. Kayıplı sıkıştırma teknikleri ise bilgide bazı kayıplara neden olur.

Kayıpsız veri sıkıştırma, veri kümesinin ilişkisizlendirilmesi (genellikle DPCM); kayıpsız entropi kodlama (Koşu Uzunluğu Kodlaması, Öngörü Kodlaması, Huffman kodlaması) ile sağlanır. Kayıplı veri sıkıştırması ise veri kümesinin ilişkisizlendirilmesi (Genellikle DCT ve dalgacık dönüşümü), kuantalama ve kayıpsız entropi kodlamayı içerir. Şekil 2.3' de kayıplı sıkıştırma gösterilmiştir.[13]

Kayıplı veri sıkıştırılmasıyla işaretin daha az etkin olan bileşenleri atılır. Az önemli işaret bileşenlerinin atılması, kaynak bilgisinin kalitesini, örnekleme hızını, kuantalama seviyeleri sayısını etkiler .



Şekil 2.3 Kayıplı Sıkıştırma Blok Diyagramı

Veri sıkıştırma algoritmasının performansı, veri sıkıştırma oranına bağlı tekrar oluşturulan işaret için işaret gürültü oranı (SNR), kullanılan algoritmanın karmaşıklığı ve algoritmanın çalışma süresiyle belirlenir.

Veri sıkıştırma algoritmalarında ilişkisizlendirme, genellikle bir uzaydaki verinin başka bir uzayda daha küçük bir bölgede yoğunlaşması için dönüşüm yardımıyla gerçekleştirilir. Etkin bir veri sıkıştırma işlemi için dönüşüm, işaretin ortogonal serilere açılmasını sağlamalı ve böylece işaret dönüşüm uzayında katsayıları birbiriyle ilişkisiz serilerle ifade edilmelidir. Dönüşüm katsayılarının küçük bir kısmında enerjinin yoğunlaşması sağlanmalı ve hızlı hesaplama algoritmalarıyla gerçekleştirilmelidir.

2.3.1 Dönüşüm

Karhunen-Loeve (KL) dönüşümü herhangi bir görüntü ve herhangi bir katsayı kümesi için ortalama karesel hatayı minimize eden en iyi dönüşümdür. Giriş işareti için özilişki matrisini kullanarak ilişkisizlendirme işlemini optimum yapar. Bununla birlikte KL dönüşümü giriş işaretine bağlı bir dönüşümdür, bunun sonucu olarak işlem yoğunluğu fazla ve hızlı uygulamalar için uygun değildir. Sinüsoidal dönüşümler KL dönüşümüne yaklaştığı için daha sık kullanılırlar.

Hızlı Fourier Dönüşümü (FFT) iyi enerji yoğunlaşması sağlar fakat hızlı bir dönüşüm olmasına rağmen kompleks işlemler gerektirir.

Ayrık Kosinüs Dönüşümü (DCT) hızlı dönüşümdür, yüksek ilişkili görüntülerin KL dönüşümü için iyi yaklaşıklık sağlar. Katsayılarının hesaplanmasındaki kolaylık ve dönüşüm uzayına iyi enerji yoğunlaşması sağladığı için Bölüm 3.2.2' de anlatılan İşlenmemiş YAR verilerinin dönüşüm domeninde sıkıştırılması işleminde, dönüşüm algoritması olarak ayrık kosinüs dönüşümü (DCT) kullanılmıştır. Ayrık kosinüs dönüşümü katsayıları (2.40) ifadesiyle gösterilmiştir.

$$c(i, j) = \begin{cases} \sqrt{\frac{1}{N}} \cos \frac{(2j+1)i\pi}{2N} & i = 0, j = 0, 1, \dots, N-1 \\ \sqrt{\frac{1}{N}} \cos \frac{(2j+1)i\pi}{2N} & i = 1, 2, \dots, N-1, j = 0, 1, \dots, N-1 \end{cases} \quad (2.40)$$

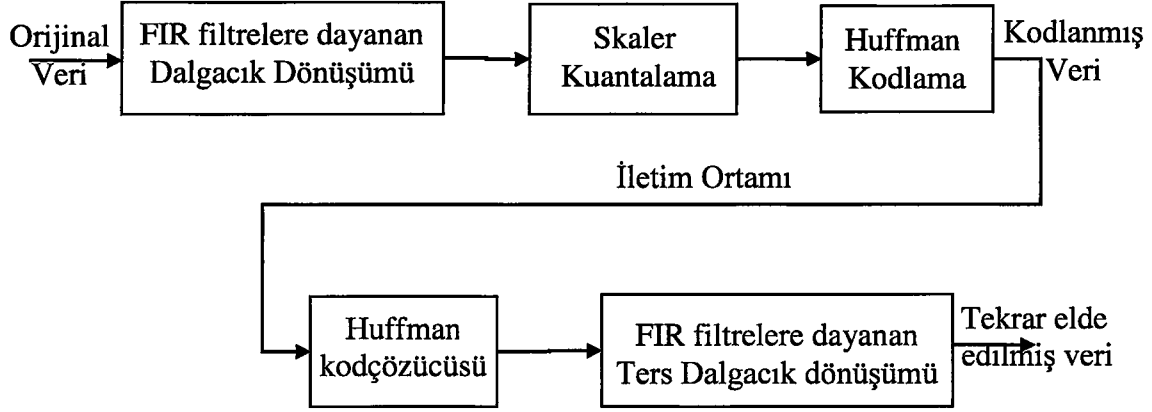
Ayrık Sinüs Dönüşümü (DST) ayrık kosinüs dönüşümüne göre iki kat daha hızlıdır. Sinüs dönüşümü reel, simetrik ve ortogonaldır.

Ayrık Walsh-Hadamard Dönüşümü (DWT) Sinüsoidal dönüşümlerden daha hızlıdır. Görüntü işleme algoritmalarının sayısal donanım uygulamalarında gereklidir. Enerji yoğunlaşması oldukça iyi olmasına karşın DCT kadar iyi değildir.

Düşük veri hızlarında her bir bloğun bağımsız işleminin doğal sonucu olan blok etkisidir yukarıda tanıtılan dönüşümler için en önemli problemdir. Süreksizlikler blok sınırlarında belirebilir. Zaman domeninde Walsh-Hadamard dönüşümünün konumsal ayırıcılığı iyidir, sinüsoidal dönüşümler için ise frekans ayırıcılığı iyidir. Veri sıkıştırma için hem konumda hem de frekansda iyi ayırıcılık sağlayabilecek dönüşüm ise Dalgacık Dönüşümüdür. [13,14] Bu amaçla **Dalgacık Dönüşümü** hem konumsal hem de frekans ayırıcılığı arasındaki verimli dengeyi sağlar. Dalgacık Dönüşümü'nde dönüşüm katsayıları oldukça ilişkisizdir. Dalgacık Dönüşümünü gerçekleyen piramit yapılı ve konvolüsyon içeren algoritmalar için katsayılar hızlı hesaplanabilir. Dalgacık Dönüşümüne dayanan algoritmalar için istenmeyen blok etkisi oluşmaz ve diğer dönüşümlerdeki gibi dönüşümü gerçekleştirecek sadece bir dalgacık ailesi yoktur, bu ise istenen uygulama için uygun dalgacık ailesinin seçilmesine olanak sağlar.

Bununla birlikte veri sıkıştırma sadece dönüşümden ibaret bir işlem değildir. Dönüşüm sonrası kuantalama ve kodsözcüğü atama adımları dönüşümle birlikte

sıkıştırma algoritmasını oluşturur. Şekil 2.4’de Dalgacık bazlı veri sıkıştırma ve sıkıştırılmış verinin tekrar elde edilmesini gösteren blok yapısı görülmektedir.



Şekil 2.4 Dalgacık dönüşümü temelli sıkıştırma sistemi blok diyagramı

2.3.2 Kayıpsız Kodlama

Bir veri dizisinde her bir sembolün taşıdığı bilgi miktarı, sembollerin veri içinde bulunma olasılıklarına bağlıdır. Bir sembolün bulunma olasılığı ile taşıdığı bilgi miktarı ters orantılıdır. Bir sembolün bir veri dizisinde bulunma olasılığı ne kadar fazla ise taşıdığı bilgi miktarı o derece azdır. Bir veri dizisinde sembollerin taşıdığı bilgi miktarı (2.41) ifadesiyle belirlenir.

$$I = -\log_2(p_i) \text{ bit} \quad (2.41)$$

(2.41) ifadesinde p_i i.sembolün gerçekleşme olasılığıdır. Entropi bir veri dizisinde bulunan tüm sembollerin ortalama bilgi miktarıdır.

$$H = -\sum_{i=1}^M p_i \cdot \log_2(p_i) \quad (2.42)$$

(2.42) ifadesinde M veri dizisindeki sembol sayısıdır. Kayıpsız kodlama sistemlerinde, bütün sembolleri eşit sayıda bitle göstermek yerine, sık kullanılan sembollere daha kısa uzunlukta kod sözcükleri, daha az kullanılanlara ise daha uzun kod sözcükleri atanmasına dayanır.

Kodama sonrası her bir sembolün kaç bitle temsil edilmesi gerektiğini, ortalama kod kelimesi uzunluğu gösterir

$$E[n] = \sum_{i=1}^M n_i \cdot p_i \quad (2.43)$$

Bu ifadede n_i i . sembole atanmış kod kelimesinin uzunluğudur. Entropi sembollerin ortalama olarak kaç bitle ifade edilip kodlanabileceğini verir.

Huffman Kodlaması ise optimum kod kuralını oluşturan bir yöntemdir. Huffman kodlaması veri dizisindeki sembollerin olasılıkları bilinmesi durumunda kullanılabilir. Huffman kodlamasında olasılıkları bilinen semboller büyük olasılıktan küçük olasılığa doğru sıralanır. Daha sonra en küçük olasılıkdaki iki sembol birleştirilerek ve olasılıkları toplanarak, yeni semboller yeniden olasılıklarına göre sıralanır. Bu sembollerin birleştirilmesi ve olasılıklara göre sıralama işlemi iki sembol kalıncaya kadar devam eder. İki sembol kaldığında ise sembollerin birine 1 diğerine 0 biti atanarak birleştirilen sembollerden geriye doğru ayrıştırırken herbirine yeni bitler ilave ederek baştaki sembollere ulaşılır. Bir veri dizisindeki sembollerin dizide bulunma olasılıkları $P(S_1)=0.25$, $P(S_2)=0.25$, $P(S_3)=0.20$, $P(S_4)=0.15$, $P(S_5)=0.10$, $P(S_6)=0.05$ olsun. Bu sembollerin olasılıklarına göre nasıl sıralanıp birleştirilebileceği yukarıda belirtilen örnek veri dizisi için Tablo 2.1’ de gösterilmiştir. Tablo 2.2’ de ise sembollerin ayrıştırılması ve kod kelimelerinin atanması gösterilmiştir. [10]

Tablo 2.1 Huffman kodlaması için sembollerin birleştirilmesi

$\underline{S_i}$	$\underline{P(S_i)}$	$\underline{S_i}$	$\underline{P(S_i)}$	$\underline{S_i}$	$\underline{P(S_i)}$	$\underline{S_i}$	$\underline{P(S_i)}$	$\underline{S_i}$	$\underline{P(S_i)}$
S1	0.25	S ₁	0.25	S _{4,5,6}	0.30	S _{2,3}	0.45	S _{1,4,5,6}	0.55
S2	0.25	S ₂	0.25	S ₁	0.25	S _{4,5,6}	0.30	S _{2,3}	0.45
S3	0.20	S ₃	0.20	S ₂	0.25	S ₁	0.25		
S4	0.15	S ₄	0.15	S ₃	0.20				
S5	0.10	S _{5,6}	0.15						
S6	0.05								

Tablo 2.2 Huffman kodlaması için sembollerin ayrıştırılması ve kod kelimelerinin atanması

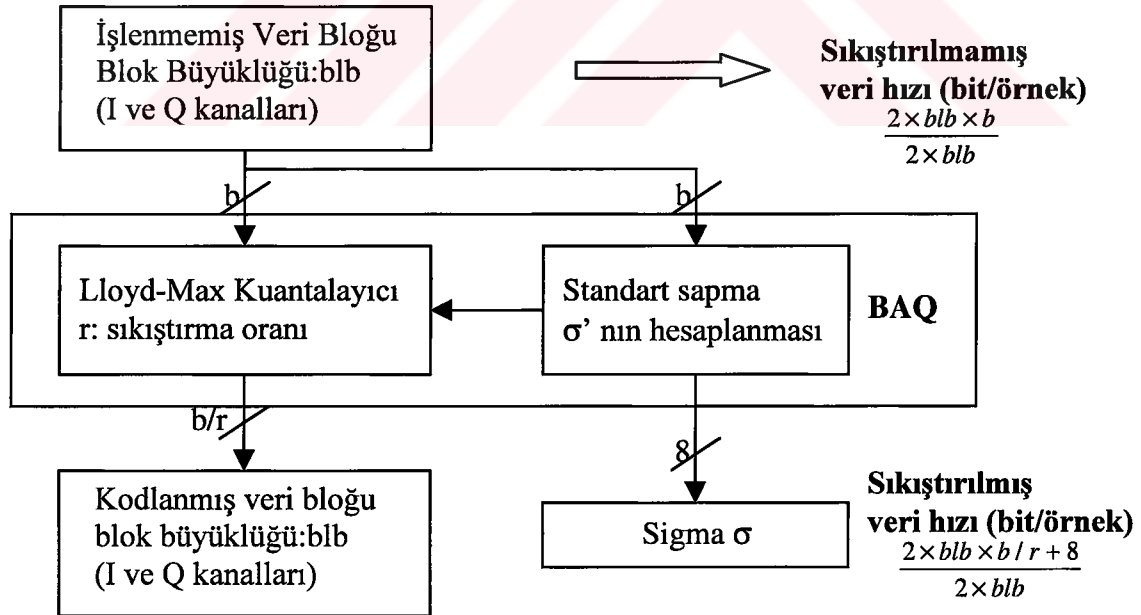
S_i	n_i	S_i	n_i	S_i	n_i	S_i	n_i	S_i	n_i
$S_{1,4,5,6}$	0	$S_{1,4,5,6}$	0	$S_{4,5,6}$	01	S_1	00	S_1	00
$S_{2,3}$	1	S_2	11	S_1	00	S_2	11	S_2	11
		S_3	10	S_2	11	S_3	10	S_3	10
				S_3	10	S_4	011	S_4	011
						$S_{5,6}$	010	S_5	0101
								S_6	0100

Tablo 2.1 ve Tablo 2.2' de verilen örnek için veri dizisinin entropisi 2,42 dir. Kodlama sonucu ortalama kod kelimesi uzunluğu ise 2,45 dir. Sonuç entropiye yakın çıkmıştır.

3. İŞLENMEMİŞ YAR VERİSİ İÇİN SIKIŞTIRMA YÖNTEMLERİ

3.1 Blok Adaptif Kuantalama (BAQ) Algoritması

Blok adaptif kuantalama algoritması zaman domeninde işlenmemiş YAR datalarının sıkıştırılmasında kullanılır. BAQ, sensörün bulunduğu platformda, sıkıştırma işleminin gerçek zamanda yapılmasına olanak sağlar. Saçıcılardan yansıyan işaretin I ve Q bileşenlerinin istatistiksel özelliklerinden yararlanır. I ve Q bileşenleri Gauss dağılımlı, sıfır ortalamalı ve bilinmeyen varyanslıdır. Algoritmanın adaptif olması, gelen data bloğunun uygun alt bloklara ayrılması ve her ayrılan blok için varyans bilgisinin hesaplanarak kuantalamada kullanılacak eşik bilgisinin varyansa bağlı bulunmasıyla sağlanır. İşlenmemiş veri bloğunu oluşturan alt bloklar, Gauss Dağılımını sağlayacak kadar büyük, blok içindeki enerjinin sabit olmasını garanti edecek kadar küçük boyutta seçilmelidirler. Şekil 3.1’ de BAQ için kullanılan algoritmanın blok diagramı gösterilmiştir.[4,9,10]



Şekil 3.1 BAQ blok diagramı

BAQ algoritmasında işlenmemiş veri temelde bir kaç bitle kodlanır. Bit sayısının az olması blok içindeki işaret gücünün değişim aralığının, tüm veri içindeki gücün değişim aralığına göre çok küçük olmasına dayanır. BAQ algoritmasında kullanılan Lloyd-Max kuantalayıcı, kuantalanacak işaretin olasılık yoğunluk fonksiyonunun bilinmesi durumunda eşğin kuantalama hatalarını minimize edecek şekilde seçilmesini sağlar.

Karar eşiklerini $b_i, i = 0..M$, yeniden oluşturma seviyelerini $y_i, i = 1..M$ ve kuantalama operatörünü $Q(.)$ ile gösterilirse

$$Q(x) = y_i, \text{ eğer } b_{i-1} < x < b_i$$

olarak verilir. Ortalama karesel kuantalama hatası;

$$\sigma_q^2 = \int_{-\infty}^{\infty} (x - Q(x))^2 \cdot p_x(x) dx \quad (3.1)$$

$$= \sum_{i=1}^M \int_{b_{i-1}}^{b_i} (x - y_i)^2 \cdot p_x(x) dx \quad (3.2)$$

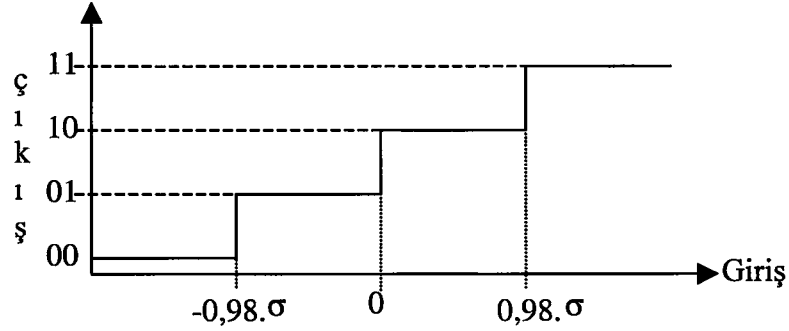
olarak belirlenebilir. Kaynağın istatistiksel modeli bilindiğine göre b_i karar eşikleri ve y_i yeniden oluşturma seviyeleri (3.2) ifadesi minimize edilerek bulunmaktadır. Karesel kuantalama hatasının minimize edilmesi için (3.2) ifadesinin y_j ' e göre türevi alınıp sıfıra eşitlenirse

$$y_j = \frac{\int_{b_{j-1}}^{b_j} x p_x(x) dx}{\int_{b_{j-1}}^{b_j} p_x(x) dx} \quad (3.3)$$

elde edilir.(3.3) ifadesinde b_j ' ye göre türev alıp sonuç sıfıra eşitlenirse;

$$b_j = \frac{y_{j+1} + y_j}{2} \quad (3.4)$$

bulunur. Karar eşiği, iki komşu tekrar oluşturma eşiğinin orta noktasıdır. Bu iki ifade çözüldüğünde ortalama karesel kuantalama hatasını (MSE) minimize eden tekrar oluşturma seviyeleri ve karar eşiği bulunabilir. Bununla birlikte y_j 'yi çözmek için b_j ve b_{j-1} değerlerine ve b_j 'yi çözmek için ise y_{j+1} ve y_j değerlerine ihtiyaç duyulur. 1960 yılında Joel Max bu iki denklemin iterasyon yoluyla nasıl çözüleceğini göstermiştir. Şekil 3.2' de Lloyd-Max kuantalayıcıya ait 2-bitlik kodlama karakteristliği gösterilmiştir.[10]



Şekil 3.2 İki bitlik BAQ algoritması için kodlama karakteristliği

Tablo 3.1' de ise kaynağın olasılık yoğunluk fonksiyonunun bilinmesi durumunda kuantalama hatalarını optimize eden kuantalayıcı için kuantalayıcı sınırları ve tekrar oluşturma seviyeleri uniform olmayan Gauss kuantalayıcı için sadece pozitif değerler için gösterilmiştir.

Tablo 3.1 Uniform olmayan Gauss (BAQ) Kuantalayıcı sınırları ve tekrar oluşturma seviyeleri

Seviyeler	b_i	BAQ y_i	SNR
4	0,0 0,9816σ	0,4528σ 1,510σ	9,3 dB
6	0,0 0,6589σ 1,447σ	0,3177σ 1,0σ 1,894σ	9,3 dB
8	0,0 0,7560σ 1,050σ 1,748σ	0,2451σ 0,6813σ 1,3440σ 2,1520σ	14,62 dB

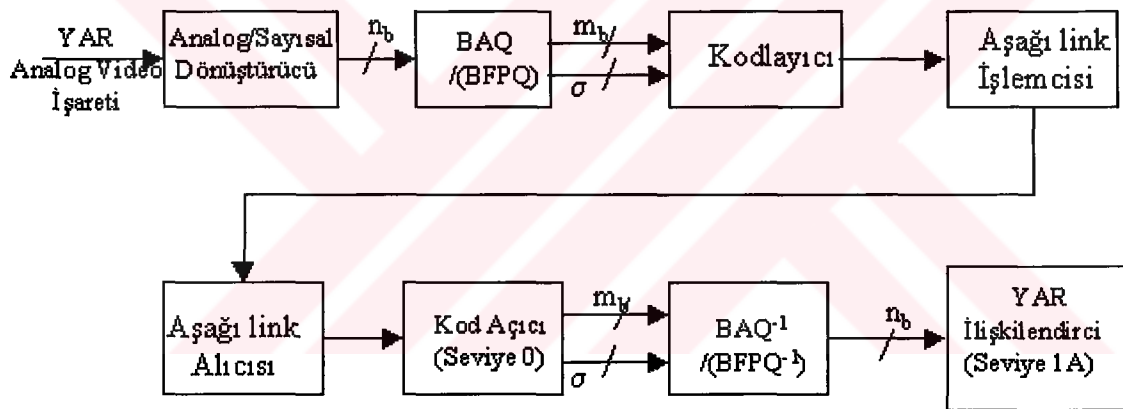
BAQ algoritması Magellan' da başarıyla uygulanmıştır. SIR-C sisteminde ise Blok Değişen Nokta Kuantalayıcısı (Block Floating Point Quantizer, BFPQ)

kullanılmıştır. BFPQ algoritmasında BAQ dan farklı olarak uniform kuantalayıcı kullanılmıştır. Tablo 3.2’ de Uniform Gauss Kuantalayıcı sınırlarının ve tekrar oluşturma seviyelerinin pozitif değerleri 2 Bitlik kuantalayıcı için gösterilmiştir.

Tablo 3.2 Uniform olmayan Gauss (BAQ) Kuantalayıcı sınırları ve tekrar oluşturma seviyeleri

Seviyeri	b_i	BFPQ y_i
4	0,0 0,9816 σ	0,52 σ 1,72 σ

Şekil 3.3’de YAR sistemi için BAQ ve BFPQ algoritmalarının uygulaması görülmektedir. Şekilde görüldüğü gibi Analog sayısal dönüştürücü çıkışı n_b bit olan işaret BAQ/BFPQ çıkışında m_b bitle kodlanmakta ve ele alınan her bir blok için standart sapma da işaretle birlikte gönderilmektedir.



Şekil 3.3 YAR sistemi için BAQ ve BFPQ algoritmaları

Her bir data bloğu için standart sapma her örneğin mutlak değerinin ortalamasının hesaplanması yardımıyla bulunur. (3.5) ifadesinde standart sapmanın her bir örneğin mutlak değerinin ortalamasıyla olan ilişkisi görülmektedir. (3.5) ifadesinde L_v seviyesi sayısını, x_i ise kuantalama sınırlarını göstermektedir. [9]

$$\overline{|x|} = \sum_{i=1}^{L_v} \frac{|x_i| + 0,5}{\sqrt{2\pi}\sigma} \int_{x_i}^{x_{i+1}} \exp(-x^2 / 2\sigma^2) dx \quad (3.5)$$

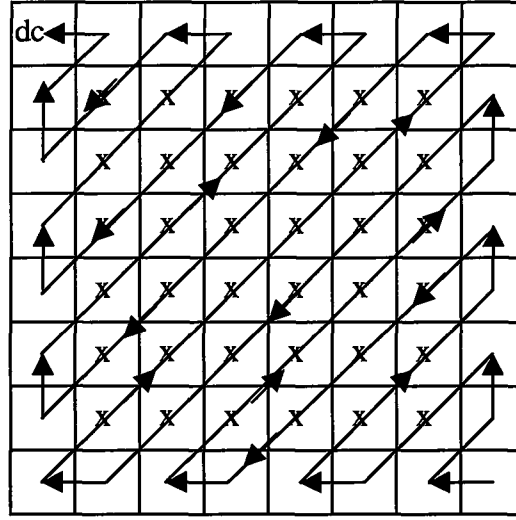
Alıcı tarafta standart sapma yardımıyla kuantalayıcı tekrar oluşturma seviyeleri belirlenmektedir. Standart sapmanın 8 bitle kodlanabileceğini düşünmek uygun olur.

3.2 Dönüşüm Domeninde Sıkıştırma

Dönüşüm domeninde sıkıştırma işlemi verinin dönüşüm domenindeki gösteriminde enerjinin belli bir bölgeye yoğunlaşacağı varsayımı üzerine yapılır. Enerjinin yoğunlaştığı bölgeler korunur, enerjinin sıfıra yakın olduğu değerlere sıfır atanır ve böylelikle gereksiz veya az önemli veri atılarak yüksek sıkıştırma oranları elde edilebilir. Dönüşüm domeninde sıkıştırma işleminde en çok kullanılan standart JPEG standartıdır. JPEG, DCT dönüşümüne dayanır.

3.2.1 JPEG ile Görüntü Kodlama

JPEG görüntü kodlama için kullanılan yaygın bir standarttır. Görüntü 8x8' lik bloklara ayrılır ve herbir bloğa DCT dönüşümü uygulanır. Dönüşüm sonrası oluşan her bir bloğun benzer indeksli katsayıları orijinal görüntüdeki yerleri değişmemek koşuluyla biraraya getirilir. Buradaki amaç herbir bloğun benzer indeksli katsayılarına aynı anda sıfır atama işlemini yapmaktır. JPEG ile görüntü kodlamada insanın görme özellikleri dikkate alınır. İnsan gözü görüntünün yüksek frekanslı bileşenlerine karşı daha az duyarlıdır. Yüksek frekanslı bileşenler görüntüde ani değişimleri gösterirler. Doğal görüntülerde bu tür geçişler baskın değildir. Bu yüzden DCT sonrası işaretin DC ve alçak frekanslı bileşenlerinde, yüksek frekanslı bileşenlerine göre daha fazla enerji yoğunlaşır. Bu dönüşüm sonrasında yüksek frekanslı bileşenlerin atılabileceği sonucuna ulaşılır. Yüksek frekanslı bileşenlerin atılma sırası, enerjisi az olan bileşenlerden enerjisi daha fazla olan bileşenlere doğru olmalıdır. Bunu her zaman sağlamasa bile JPEG zig-zag tarama yapısını kullanır. Zig-Zag tarama yapısı Şekil 3.4' de gösterilmiştir. Şekil 3.4' de gösterildiği düzende sıfır atama işlemi gerçekleştirilir. Doğal resimler için bu tür bir tarama yapısı oldukça iyi sonuç vermektedir. Tarama matrisinin sadece ilk 32 değerinin korunması durumunda bile görüntüde belirgin bozulma olmamaktadır. [10] Bu durum, doğal bir görüntünün herhangi bir 8x8' lik bloğunun dönüşüm katsayılarına bakılarak anlaşılabilir. Aynı şekilde işlenmemiş YAR için herhangi bir 8x8' lik bloğun dönüşüm değerleri gözönüne alınarak işlenmemiş YAR için dönüşüm temelli sıkıştırma tekniklerinin geliştirilmesi amaçlanmıştır.



Şekil 3.4 8x8 dönüşüm için zig zag tarama yapısı

Şekil 3.5 'de YAR işlenmemiş datasından alınan 8x8 lik bir veri bloğunun DCT sonrası katsayıları görülmektedir. Şekilde de görüldüğü üzere katsayıların tüm blok içindeki dağılımı JPEG de kullanılan zigzag tarama için uygun değildir.

1071	-10,6	39,55	0,3	0	39	51	0
36,6	0,18	0	56	25	0	60	55
84	13,14	31,8	71	0,4	0,7	0,3	55
-28	0,15	43,81	29	0	0,5	0,3	61
35	0,019	37	0,1	135	0	103,5	0
0,35	34,99	54	63	0	103	0,16	0
0,026	0,12	77,9	0	103	0,17	0,17	0,11
0,16	0,24	0	97,76	0,5	0,04	0	0,24

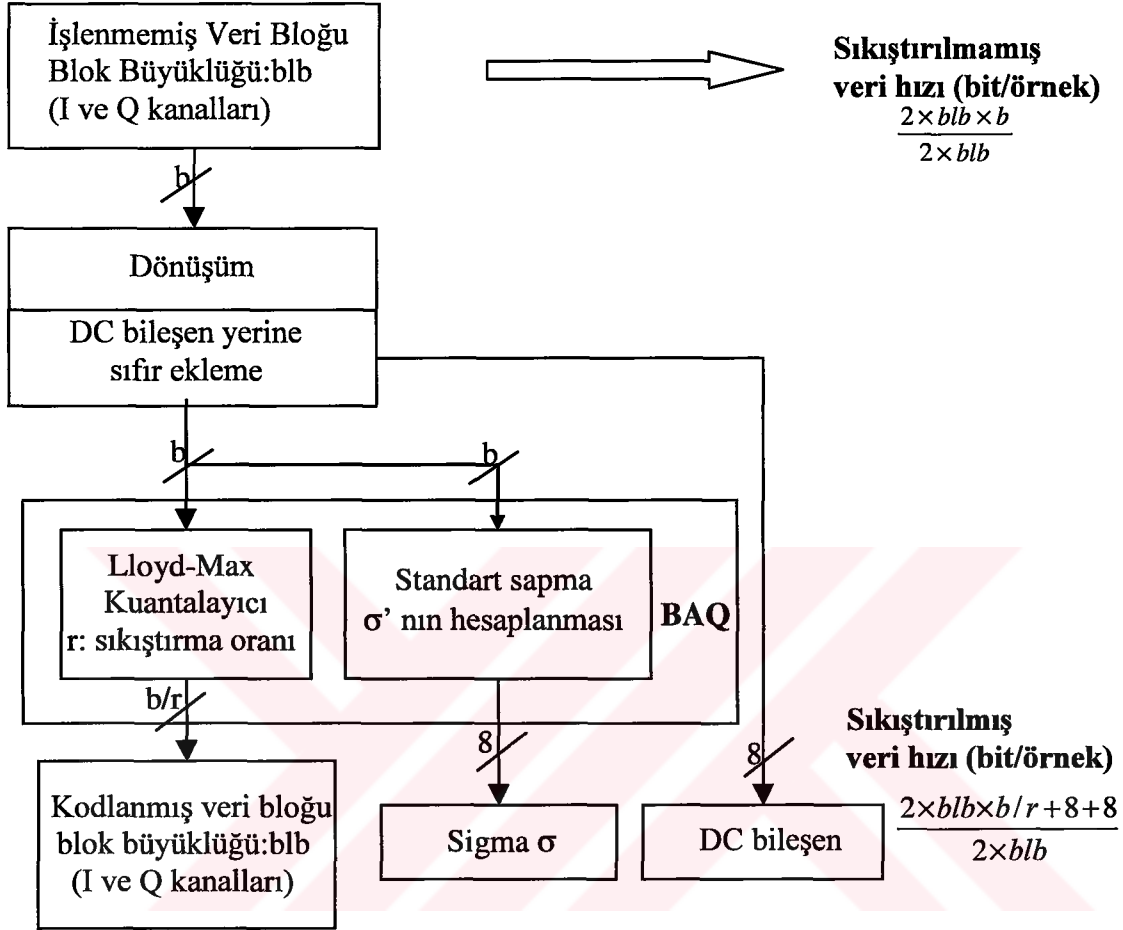
Şekil 3.5 İşlenmemiş YAR 8x8 veri bloğunun DCT katsayıları

Katsayıların belli bir düzen içinde olmadığı, işlenmemiş YAR datasının düşük frekans bileşenlerinin yanında, yüksek frekans bileşenlerinin de etkin olduğu görülmektedir. Bu durum işlenmemiş YAR verilerinin zaman domeninde dinamik aralığının fazla olmasıyla açıklanabilir. Veriler arasındaki sert geçişler yüksek frekans bileşenlerinin de etkin olmasının kanıtıdır.

3.2.2 İşlenmemiş YAR verilerinin Dönüşüm domeninde sıkıştırılması

Dönüşüm domeninde sıkıştırma, zaman domeninde çok sayıda olan birbiriyle ilişkili verilerin daha az sayıda birbiriyle ilişkisiz dönüşüm katsayılarıyla ifade edilmesiyle sağlanır. Dönüşüm sonrası daha az anlamlı katsayılar atılır veya daha az bitle kodlanır. İşlenmemiş YAR verisi dönüşüm domeninde Gauss dağılımına uygun ise Blok Adaptif Kuantalama dönüşüm domeninde de uygulanabilir. Bu şart oldukça geniş blok boyutları için sağlanır. İşlenmemiş YAR verisi hem azimuth hem de menzil doğrultusunda frekans modülasyonu içerir. Frekans modülasyonu enerjinin dönüşüm domeninde az bir bölgeye yoğunlaşması sonucunu doğurur. Bunun sonucu olarak işlenmemiş YAR verisinin dönüşüm domenindeki dinamik aralığı artar. Dinamik aralığın artması sıkıştırma performansını olumsuz etkiler, Benz, U.,Strodl, K. ve Moreira A. dönüşüm domeninde işlenmemiş YAR verilerinin sıkıştırılması için bir yöntem önermiştir.[3] Bu yöntemde işlenmemiş YAR verileri, en büyük bilgisayar duyarlılığı olan 32 bitle BAQ kullanılarak kodlanır. Böylelikle işlenmemiş YAR verilerinin dinamik aralığının düşürülmesi amaçlanmıştır. Burada BAQ ile kodlanan veri bloklarının boyutu, bls_1 , daha önceki bölümde anlatıldığı gibi Gauss dağılımını sağlayacak ölçüde büyük, veri bloğu içindeki enerji seviyesinin sabit tutulabilmesi amacıyla da yeterince küçük seçilmelidir. BAQ sonrası veri blokları biriktirilerek bls_2 boyutunda $bls_2 \gg bls_1$ şartını sağlayan bir veri bloğu oluşturulur. Dönüşüm bu biriktirilmiş veri bloğuna uygulanır. Dönüşüm sonrası kodlama işlemi yine BAQ kodlayıcı ile yapılır. Dönüşüm sonrası BAQ kod şemasının bulunması Benz, U.,Strodl, K. ve Moreira tarafından açıklanmıştır. [3] Bu tezde dönüşüm temelli sıkıştırma işleminde, dinamik aralığın azaltılma işlemi dönüşüm sonrası yapılmıştır. Dönüşüm sonrası BAQ uygulanabilmesi için dönüşüm sonrası veri bloğunun Gauss dağılımlı olması gerekir. Dönüşüm sonrası katsayılar Şekil 3.5' de gösterilmiştir. Burada DC katsayısının diğer katsayılarla göre oldukça büyük olduğu görülmektedir. Bu nedenle veri bloğunun standart sapması oldukça küçülmekte ve DC bileşene göre küçük olan katsayılar dönüşüm sonrası aynı BAQ kuantalama seviyesinde kalmakta ve bu durum kuantalayıcı seviyelerinin hepsinin kullanılabilmesini engellemektedir. Eğer dönüşüm sonrası elde edilen veri bloğunda DC bileşen ayrılır ve DC bileşenin yerine sıfır eklenirse elde edilen yeni data bloğunun BAQ ile daha verimli kodlandığı görülmüştür. Bu DC bileşen çıkarıldığında oluşan yeni data bloğunun Gauss dağılımına yaklaştığı görülür. Varyansla birlikte DC bileşenin de 8 bitle kodlanarak

gönderilebileceği düşünülürse bu durum sıkıştırma performansını biraz da olsa olumsuz etkiler fakat daha büyük data blokları için sıkıştırma oranındaki olumsuz etkisi az olur. Şekil 3.6' de işlenmemiş YAR datasının dönüşüm domeninde sıkıştırılmasına ait algoritma görülmektedir.



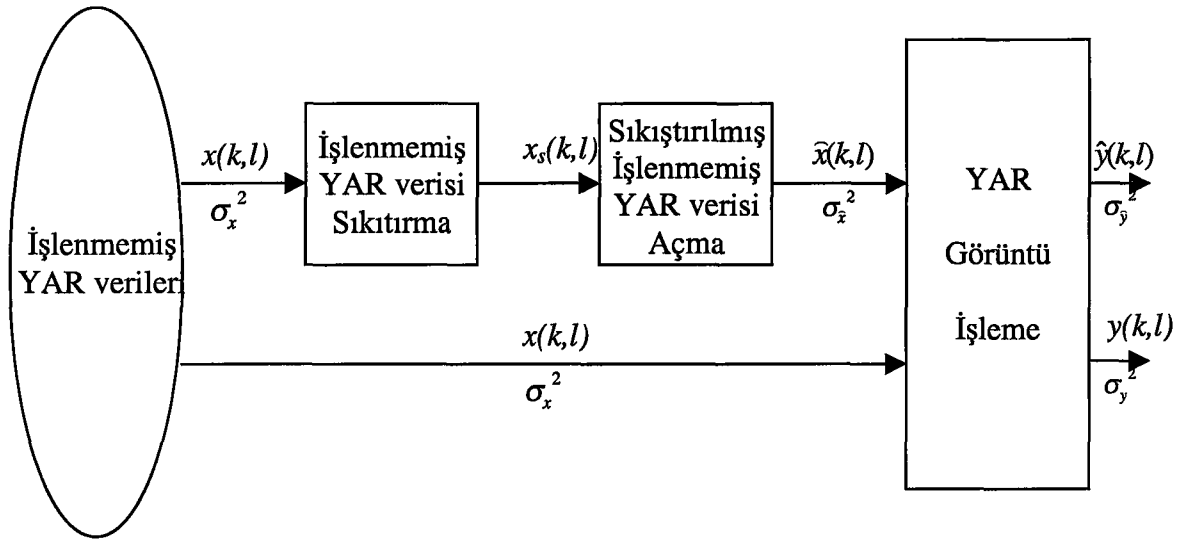
Şekil 3.6 İşlenmemiş YAR verilerinin dönüşüm domeninde sıkıştırılması

4. İŞLENMEMİŞ YAR VERİSİ SIKIŞTIRMA TEKNİKLERİ PERFORMANS ANALİZİ

Bu bölümde işlenmemiş YAR Datası için tanıtılan sıkıştırma algoritmaları, belirli başarımlarına göre karşılaştırılmaktadır. Yapılan karşılaştırmalar sonunda en iyi performans sonucunu veren teknikler belirlenecektir. Karşılaştırmalarda performans kriterlerinin yanında, görüntüler de verilmiştir. Sıkıştırma algoritmalarının gerçekleştirilmesi için MATLAB programı kullanılmıştır.

4.1 İşlenmemiş YAR Datası Sıkıştırma Teknikleri Performans Kriterleri.

Sıkıştırma işlemi sonucunda orijinal veri ile sıkıştırılıp tekrar elde edilmiş verilerin karşılaştırılması için çeşitli kriterler vardır. Bu kriterlerin belirlenmesinde sıkıştırılan verinin hangi amaçla kullanılacağı önem taşır. Kimi zaman insanın görme yapısını da kriterlerin belirlenmesinde dikkate almak gerekebilir. Ama sıkıştırılan veriler bir uydudan elde edilmiş işlenmemiş veriler ise, orijinal verilerle sıkıştırılıp tekrar elde edilmiş verilerin birebir karşılaştırılmasının yanında, sıkıştırmanın işlenmiş veri üzerindeki etkisinin de belirlenmesi gerekir. Bu çalışmada bu gereklilik bilinmesine rağmen sadece işlenmemiş veriler üzerinde başarımlar incelenmiş, sıkıştırmanın işlenmiş veriler üzerindeki etkisi incelenmemiştir. Şekil 4.1 de temel bir sıkıştırma sistemi YAR verileri için gösterilmiştir. Şekil 4.1' de $x(k,l)$ ile işlenmemiş orijinal YAR verileri, σ_x^2 ile işlenmemiş orijinal YAR işaret gücü, $\hat{x}(k,l)$ ile sıkıştırılıp açılmış işlenmemiş YAR verisi, $\sigma_{\hat{x}}^2$ ile ise sıkıştırılıp açılmış YAR işaret gücü gösterilmiştir. Karşılaştırmalarda burada belirtilen bileşenler dikkate alınmıştır. Sıkıştırma tekniklerinin başarımların analizi, ortalama karesel hata, normalize ortalama karesel hata, işaret gürültü oranı, tepeden tepeye işaret gürültü oranı, sıkıştırma oranı olmak üzere beş kriterle ortaya konmuştur.



Şekil 4.1 İşlenmemiş YAR verileri performans karşılaştırılması için sıkıştırma sistemi

Ortalama karesel hata gürültü gücünün belirlenmesini, işaret gürültü oranı ve tepeden tepeye işaret gürültü oranı ise işaret gücüyle gürültü gücünün karşılaştırılmasını sağlar. Ortalama karesel hata, MSE;

$$MSE = \frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N (x(k,l) - \hat{x}(k,l))^2 \quad (4.1)$$

MSE genellikle orijinal işaret gücüyle normalize edilir. Böylece normalize ortalama karesel hata

$$NMSE = \frac{\frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N (x(k,l) - \hat{x}(k,l))^2}{\frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N x(k,l)^2} \quad (4.2)$$

olarak bulunur. İşaret gürültü oranı, SNR

$$SNR(dB) = 10 \log_{10} \frac{\sigma_x^2}{MSE} \quad (4.3)$$

şeklindedir. (4.3)' de σ_x^2 orijinal işaret gücü

$$\sigma_x^2 = \frac{1}{N^2} \sum_{k=1}^N \sum_{l=1}^N x(k,l)^2 \quad (4.4)$$

olarak verilir. Tepeden tepeye işaret gürültü oranı, PSNR

$$PSNR(dB) = 10 \log_{10} \frac{x_{tepe}^2}{MSE} \quad (4.4)$$

olarak tanımlanır. Sıkıştırma Oranı, CR

$$CR = \frac{\text{sıkıştırılmış veri}}{\text{orijinal veri}} \quad (4.5)$$

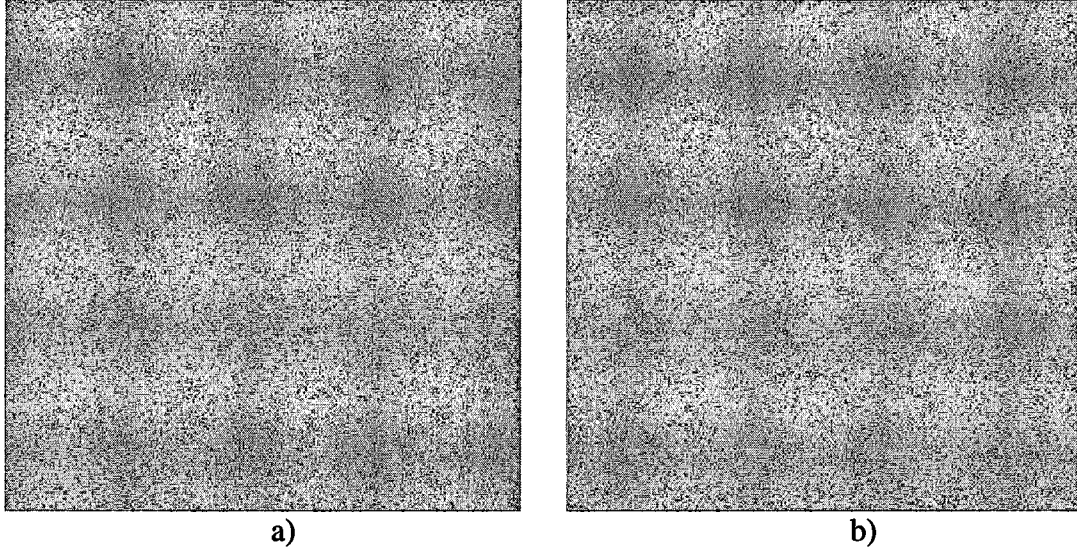
olarak belirtilir.

4.2 Performans Analizinde Kullanılan Örnek Veriler.

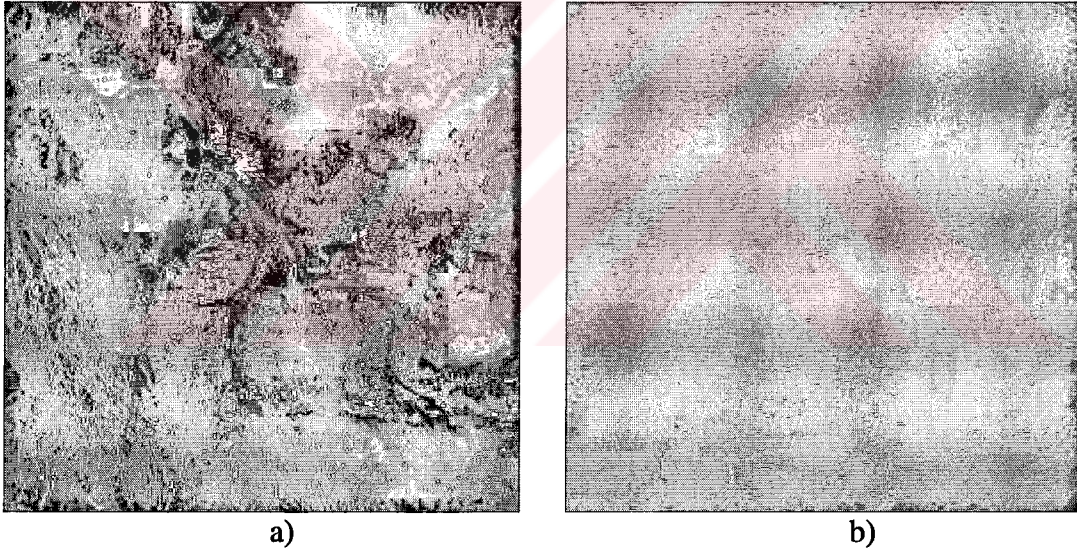
Tezde tanıtılan işlenmemiş YAR verilerinin sıkıştırılmasına yönelik kullanılan örnek veriler 256x256 piksel buyutunda 0-255 gri seviyede, piksel başına 8 bitle gösterilen görüntüler olarak alınmıştır. Örnek verilerin seçiminde işlenmemiş YAR verilerinin yanında işlenmiş YAR verileri de alınmış, İşlenmemiş YAR verileri için geliştirilen algoritmalarının İşlenmiş YAR verilerine etkileri de incelenmiştir. İşlenmemiş YAR verilerinin doğal görüntülerden farkını ortaya koymak için pikseller arası yumuşak geçişlere sahip doğal görüntü de performans analizinde örnek veri olarak kullanılmıştır. Böylelikle pikseller arasında ilişkisi az olan işlenmemiş YAR verileriyle, pikseller arasında ilişkiye sahip doğal görüntülerinin karşılaştırılması ve doğal görüntüler için geliştirilen tekniklerin neden işlenmemiş YAR verileri için kullanılamayacağını ortaya çıkarılabilir.

İşlenmemiş YAR verisi komplekstir , reel (I) ve imajiner (Q) olmak üzere iki farklı bileşenden oluşur. Şekil 4.2' de İşlenmemiş YAR verisinin reel (I) ve imajiner (Q) bileşeleri görülmektedir.

Şekil 4.2' de görülen veriler işlenmemiş ERS 1 data benzetimlerinden elde edilmiş verilerdir.[11] Şekil 4.3'a' da uydudan elde edilmiş İşlenmiş YAR görüntüsü Şekil 4.3 b' de ise noktasal gürültülü İşlenmiş YAR görüntüsü görülmektedir.

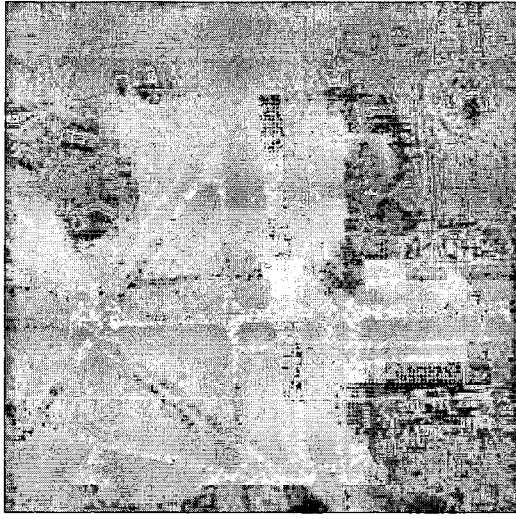


Şekil 4.2 İşlenmemiş YAR verisi a) reel (I) bileşeni ve b) imajiner (Q) bileşeni

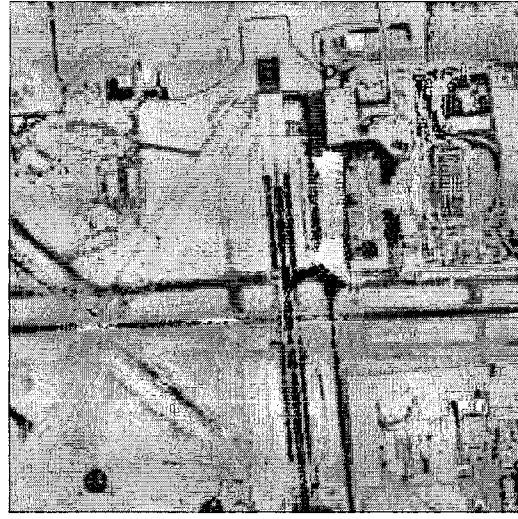


Şekil 4.3 Uydudan elde edilmiş İşlenmiş YAR görüntüleri a)Noktasal gürültüden arındırılmış L.A. görüntüsü b)Noktasal gürültülü Hollanda görüntüsü

Optik sensörlerden elde edilen görüntülerle, YAR sensöründen elde edilen görüntü için sıkıştırma performansının farklılığını ortaya koymak için ise Şekil 4.4' de aynı yere ait, uçaktan elde edilmiş YAR sensörü ve optik sensör görüntüleri kullanılmıştır



a)



b)

Şekil 4.4 Uçaktan elde edilen Havaalanı a)YAR görüntüsü b)Optik görüntü

İşlenmemiş YAR verilerinin pikseller arası yumuşak geçişlere sahip doğal görüntülerden farkını ortaya koyabilmek için ise Lena görüntüsü kullanılmıştır.

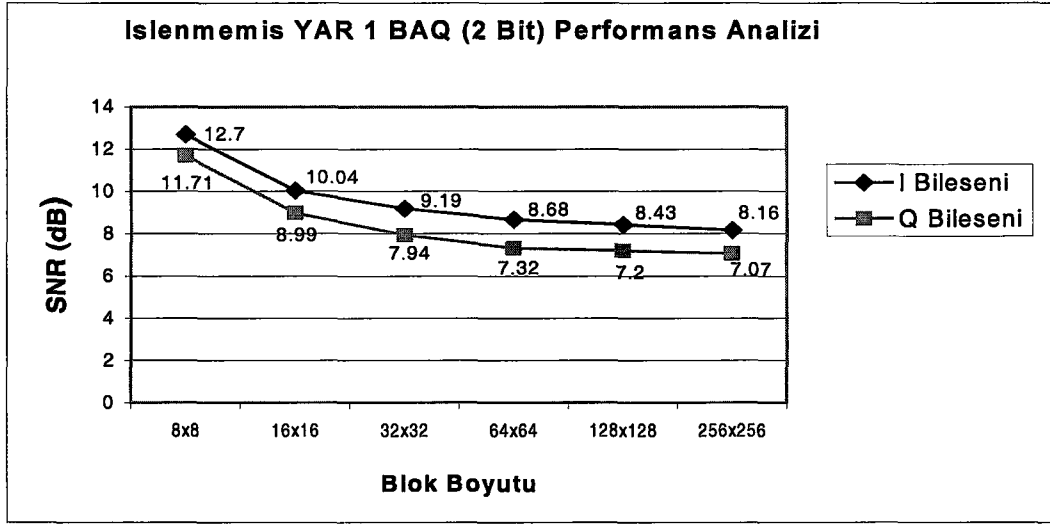


Şekil 4.5 Lena

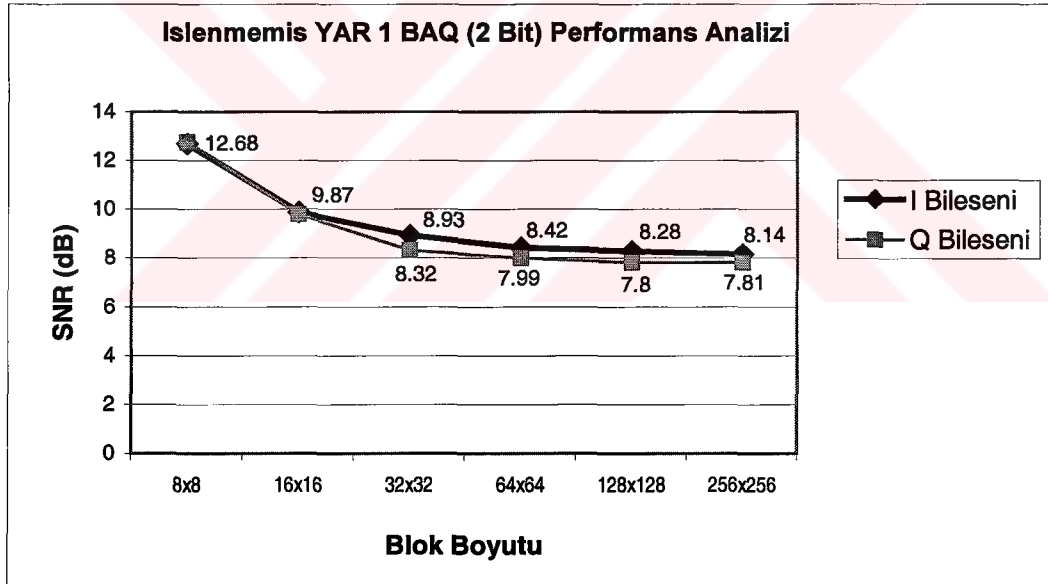
4.3 Blok Adaptif Kuantalama (BAQ) Algoritması Performans Analizi

Blok Adaptif Kuantalama Algoritması Performans Analizi için iki farklı İşlenmemiş YAR verisi kullanılmıştır. İşlenmemiş YAR verileri komplekstir ve reel (I) ve imajiner (Q) bileşenlerinden oluşur. İki farklı İşlenmemiş YAR verisinde yapılan incelemede işlenmemiş verilere ait I ve Q bileşenlerin benzer performans sergilediği

görülmüştür. Şekil 4.6' de İşlenmemiş YAR 1 verisine ait, Şekil 4.7' de ise İşlenmemiş YAR 2 verisine ait 2 Bitlik BAQ' da Blok Boyutuna bağlı işaret gürültü oranının değişimi görülmektedir.



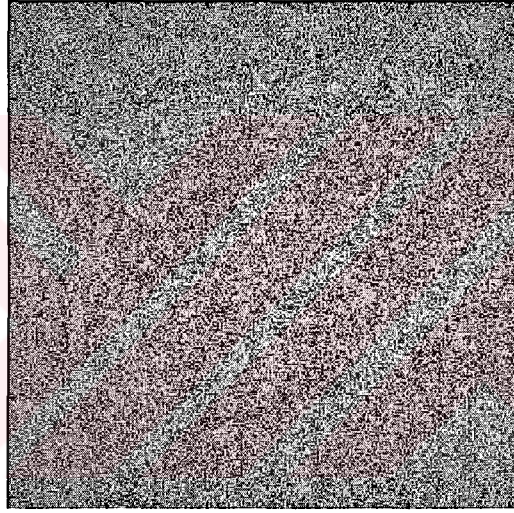
Şekil 4.6 İşlenmemiş YAR 1 verisi 2 Bit BAQ Blok Boyutuna Bağlı Analiz



Şekil 4.7 İşlenmemiş YAR 2 verisi 2 Bit BAQ Blok Boyutuna Bağlı Analiz

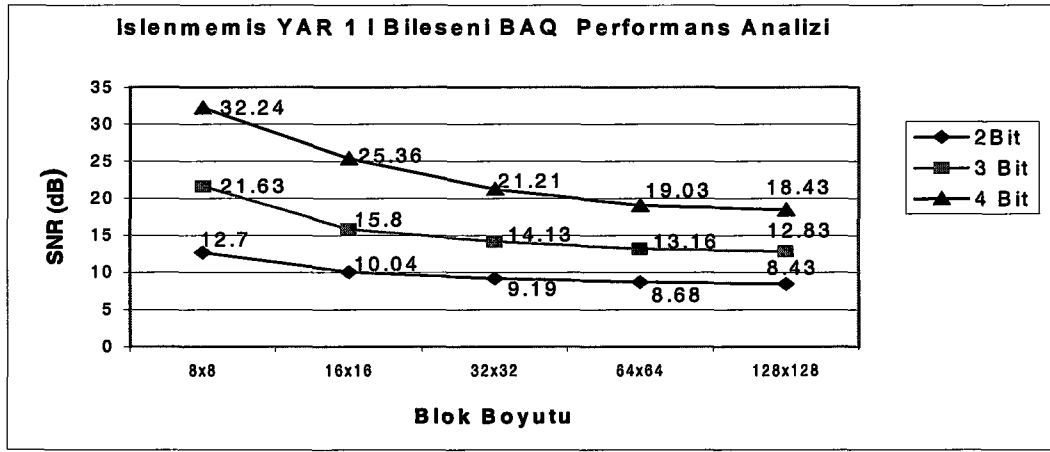
BAQ için blok boyutunun seçimi de önemli bir kriterdir. Blok boyutu, seçilen bloğun Gauss dağılımını sağlaması için yeterince büyük, blok içindeki enerjinin sabit kalması için ise yeterince küçük seçilmelidir. Bununla birlikte Gauss dağılımı için 50 örnek alınmasının yeterli olduğu bilinmektedir. BAQ algoritmasında her bir bloğa ait varyans bilgiside 8 Bitle kodlandığından blok boyutu büyüdüğünde sıkıştırma

oranının artacağı açıktır. Şekil 4.8’ de Blok boyutuna bağlı olarak sıkıştırma oranının değişimi herbir örneğin 2,3 ve 4 Bitle kodlanması durumunda gösterilmiştir. Ayrıca alıcı tarafta kod çözülürken her bir bloğa ait farklı varyans değerlerine bağlı kod çözme şemaları oluşacağından tüm veriler varyanslar dışında 2 Bitle yani 4 farklı seviyede kodlansa da sonuçta her bir blok için eğer varyans değerleri farklı ise dört farklı seviye oluşacak ,fakat tüm veri için blok sayısının 4 katı kadar farklı seviye oluşacaktır. Eğer her bir blok m bitle kodlanırsa, herbir bloğa ait varyansların farklı olma durumunda, $\text{bloksayısı} \cdot 2^m$ farklı seviye oluşacaktır. Buradan da anlaşılabacağı üzere blok sayısının artması işlenmemiş YAR verileri için işaret gürültü oranının artması sonucunu doğurur. Şekil 4.8’ de İşlenmemiş YAR 1 I bileşeninin 8x8 blok boyutunda 2 bit BAQ ile kodlandıktan sonra tekrar elde edilen görüntüsü görülmektedir.

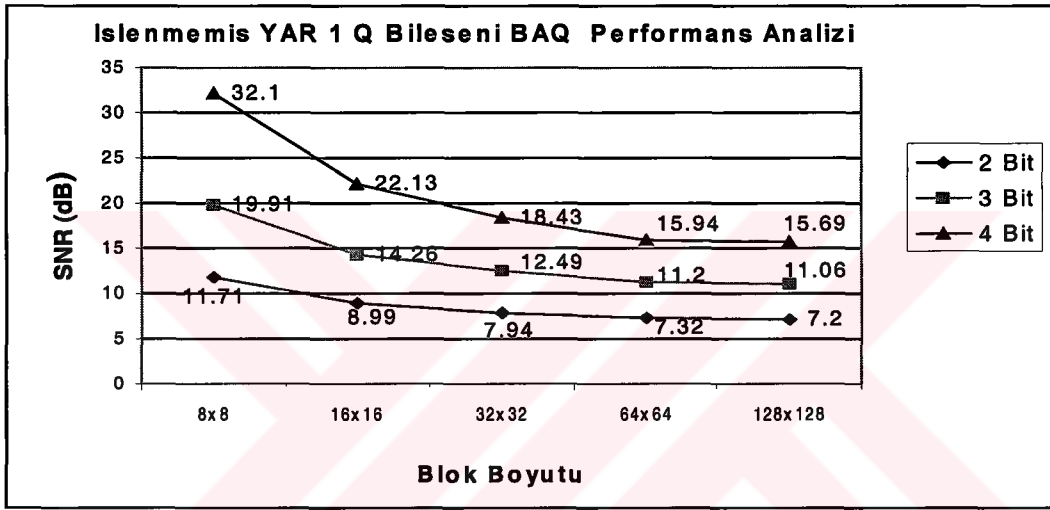


Şekil 4.8 İşlenmemiş YAR 1 verisi I Bileşeni 2 Bit BAQ SNR=12.7 dB

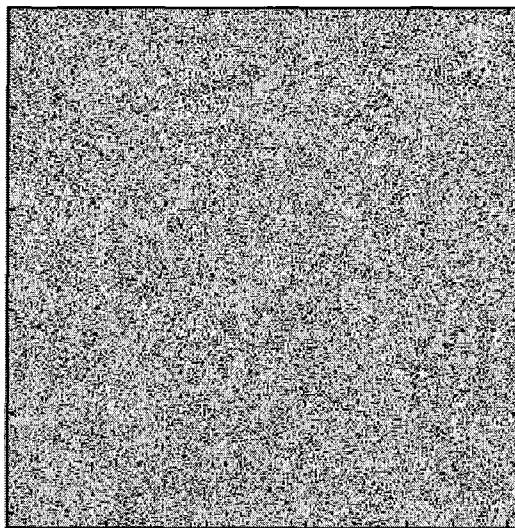
Şekil 4.9, Şekil 4.10, Şekil 4.12 ve Şekil 4.13’ de ise sırasıyla de İşlenmemiş YAR 1 I ,Q ve İşlenmemiş YAR 2 I ve Q bileşeninin 2,3 ve 4 Bitle kodlanması durumunda blok boyutuna bağlı işaret gürültü oranının değişimi görülmektedir.Şekil 4.11 a’ da ise İşlenmemiş YAR 1 verisi I Bileşeni 3 Bit BAQ SNR=20.14 dB ve b’de 4 Bit BAQ SNR=32.24 dB için gösterilmiştir.



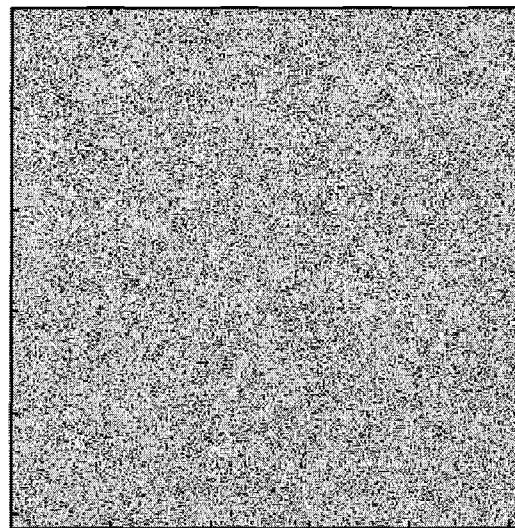
Şekil 4.9 İşlenmemiş YAR 1 I Bileşeni 2,3 ve 4 Bitle Kodlama durumunda SNR



Şekil 4.10 İşlenmemiş YAR1 Q Bileşeni 2,3 ve 4 Bitle Kodlama durumunda SNR

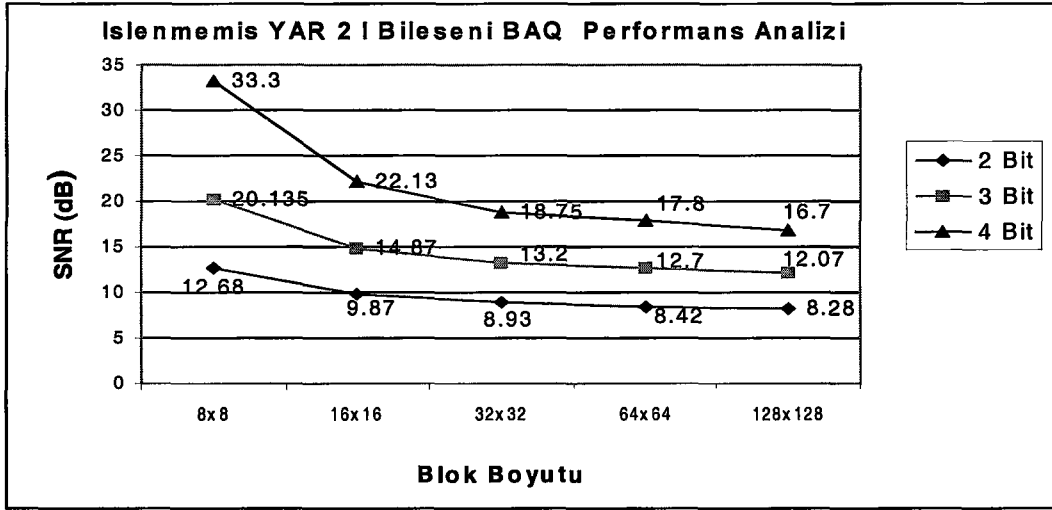


a)

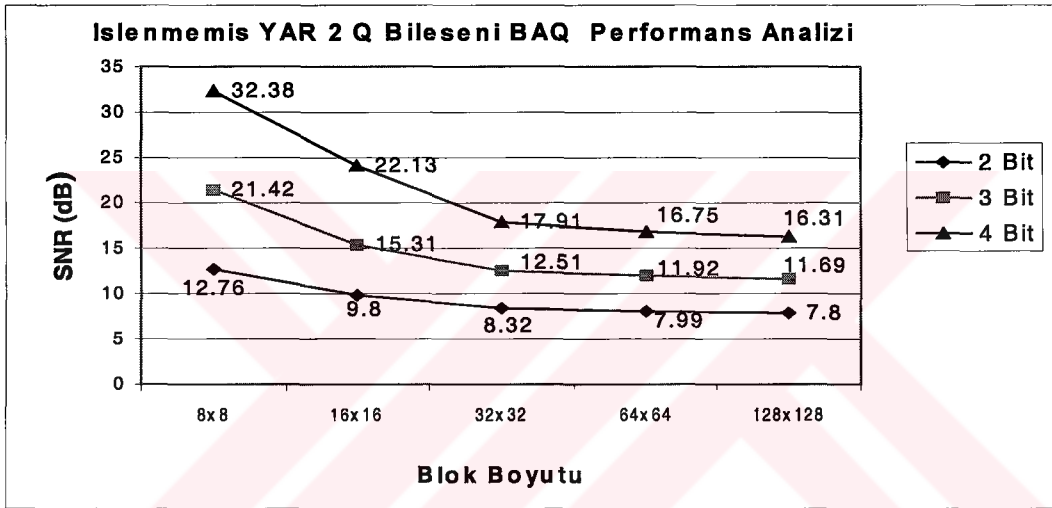


b)

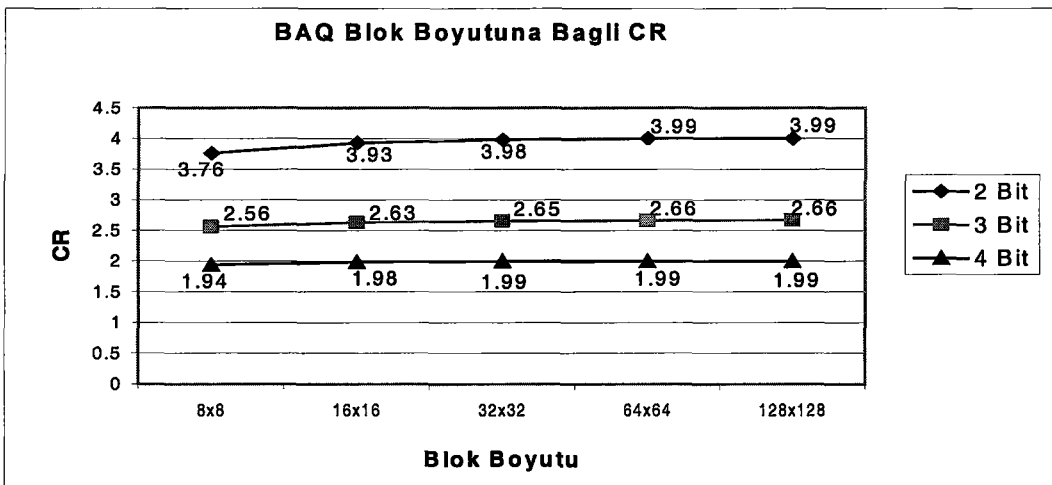
Şekil 4.11 İşlenmemiş YAR verisi I Bileşeni a)3 Bit BAQ SNR=20.14 dB b)4 Bit BAQ SNR=32.24 d B



Şekil 4.12 İşlenmemiş YAR 2 I Bileşeni 2,3 ve 4 Bitle Kodlama durumunda SNR

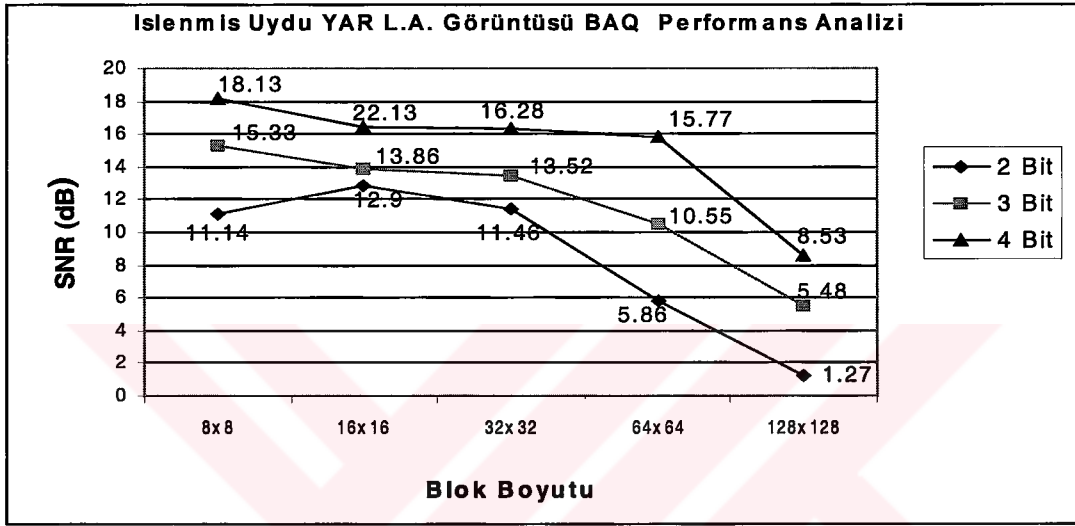


Şekil 4.13 İşlenmemiş YAR2 Q Bileşeni 2,3 ve 4 Bitle Kodlama durumunda SNR

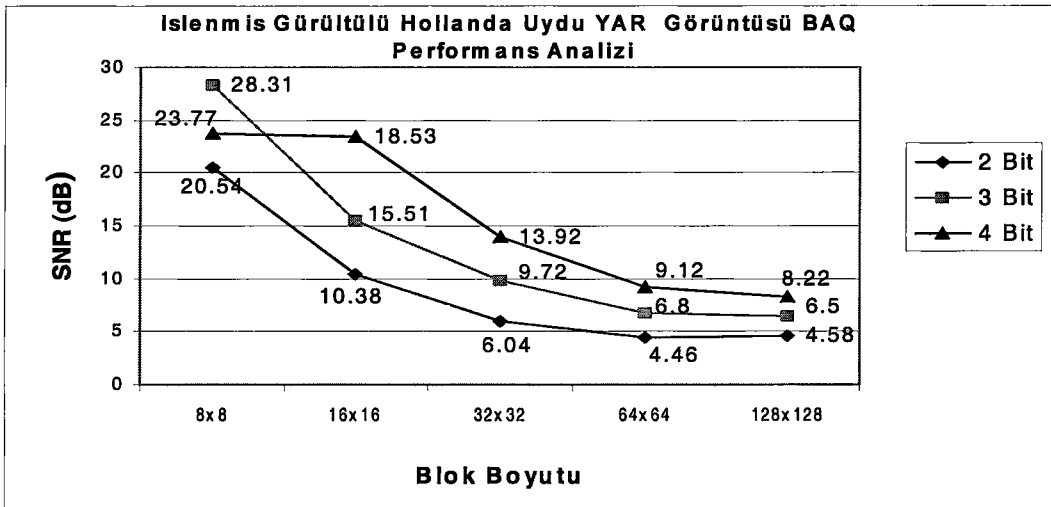


Şekil 4.14 BAQ 2,3 ve 4 Bit için Blok boyutuna bağlı sıkıştırma oranı

Şekil 4.14'da Blok Boyutuna bağlı sıkıştırma oranının değişimi 2,3 ve 4 Bit BAQ için gösterilmiştir. Blok Boyutu arttıkça sıkıştırma oranının bir miktar arttığı yani her bir blok için kodlanan varyansın sıkıştırma oranına olan olumsuz etkisinin azaldığı görülmektedir. Şekil 4.15' de ise BAQ 'nun işlenmiş Uydu YAR verisi üzerindeki etkisi görülmektedir. Şekilde görüldüğü gibi 2 Bit BAQ için 8x8' lik blok boyutunda SNR değerinin beklenenden daha düşük olduğu görülmektedir bu da bu boyut için varyansın küçük olduğunu ve daha fazla kuantalama seviyesine ihtiyaç duyulduğunu anlatır.



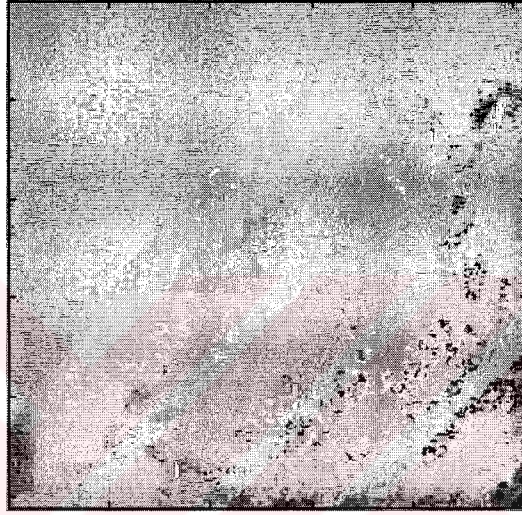
Şekil 4.15 İşlenmiş Uydu YAR L.A. Görüntüsü BAQ Blok boyutuna bağlı SNR



Şekil 4.16 İşlenmiş Gürültülü Hollanda YAR Görüntüsü BAQ
Blok boyutuna bağlı SNR

Blok boyutunun artmasıyla SNR' ın hızla düşmesi ise büyük blok boyutları için uygun istatistiği gösteremediğini açıklıyor.

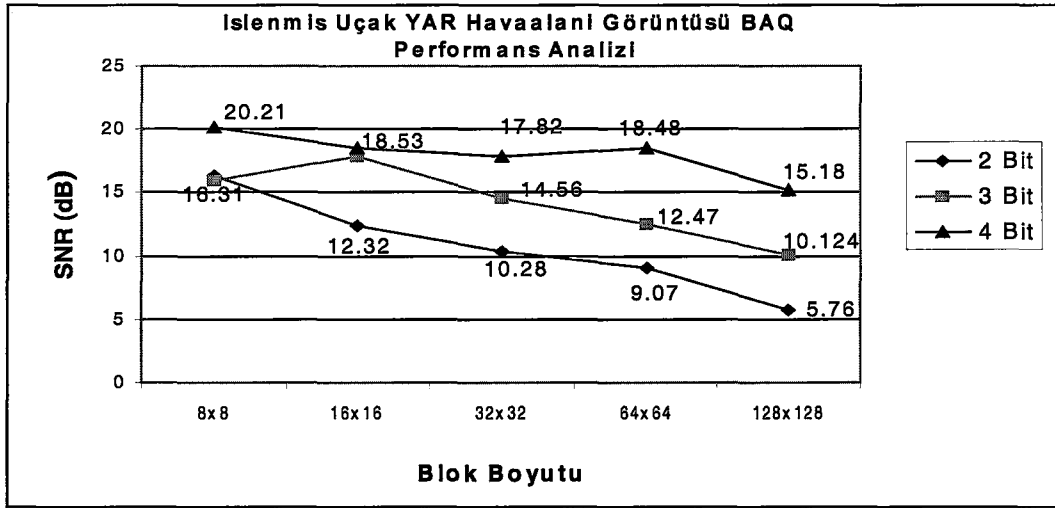
Şekil 4.17' de İşlenmiş Gürültülü YAR Hollanda verisi için BAQ Blok boyutuna bağlı SNR' ın değişimi görülmektedir. İşlenmiş YAR görüntüsü noktasal gürültüye sahiptir. İşlenmiş YAR görüntülerini noktasal gürültüden arındırmak için ortalama değer süzgeci ve sigma süzgeci kullanılabilir.[12] Şekil 4.3.11' de en dikkat çekici husus 2 bit BAQ için 8x8 boyutunda çok iyi SNR' a sahip olduğudur. Bu da işaretin noktasal gürültüye sahip olmasıyla bağdaştırılabilir.



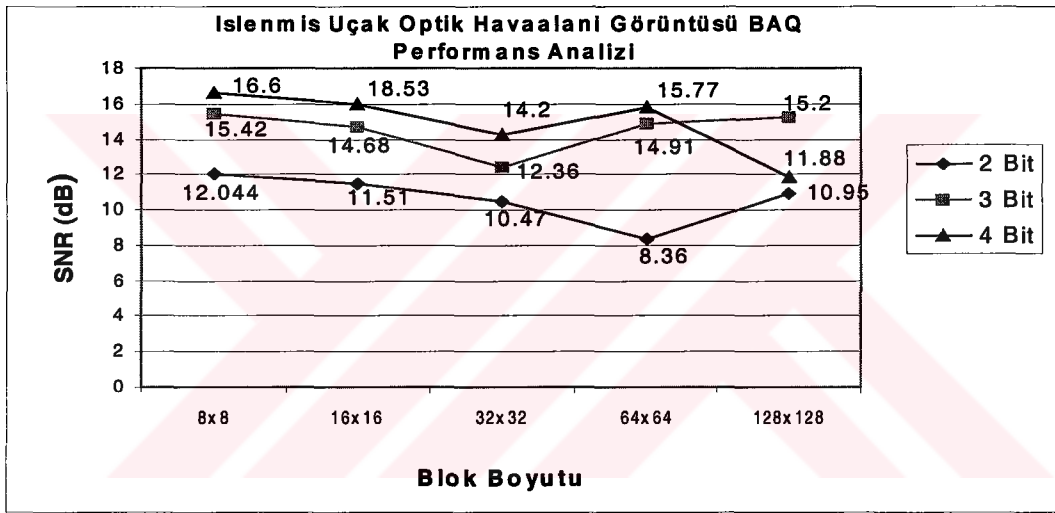
Şekil 4.17 İşlenmiş Gürültülü YAR Hollanda görüntüsü 2 Bit BAQ SNR=20.54 dB

Şekil 4.18' de İşlenmiş Uçak YAR Havaalanı görüntüsü için BAQ Performansı Blok Boyutuna bağlı SNR değişimi görülmektedir. Şekil 4.19' de ise aynı yere ait optik sensörden elde edilmiş görüntünün Performans analizi gösterilmektedir.

Aynı yere ait YAR sensöründen ve optik sensörden elde edilen görüntülerin BAQ sıkıştırma performansı incelendiğinde YAR sensöründen elde edilen işlenmiş görüntülerin daha iyi sonuç verdiği görülmektedir. Bu ise YAR sensöründen elde edilen görüntünün hedefin (görüntülenecek bölge) saçılım özelliklerini belirtmesiyle ve radar görüntülerinin noktasal gürültüye sahip olmasıyla açıklanabilir.

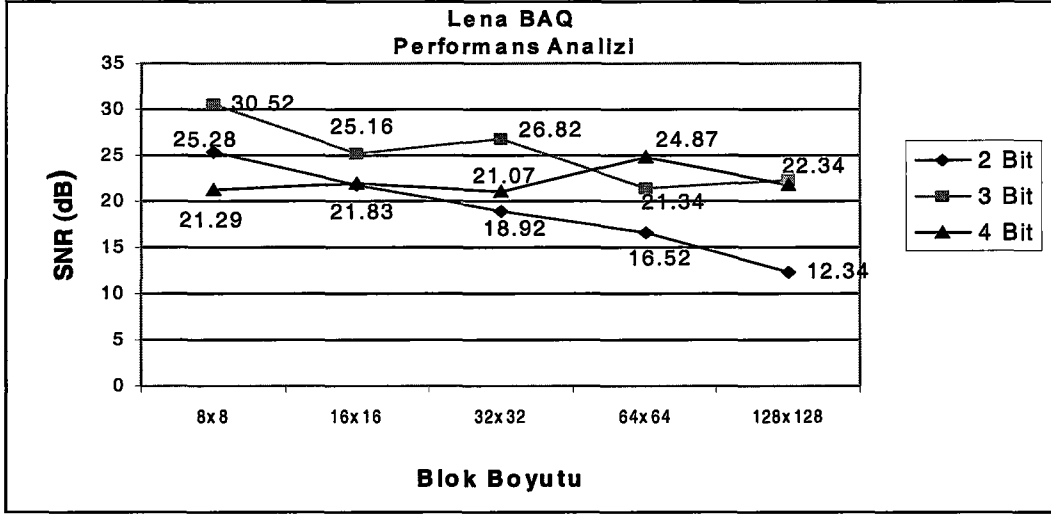


Şekil 4.18 İşlenmiş Uçak YAR Havaalanı Görüntüsü Blok Boyutuna Bağlı SNR



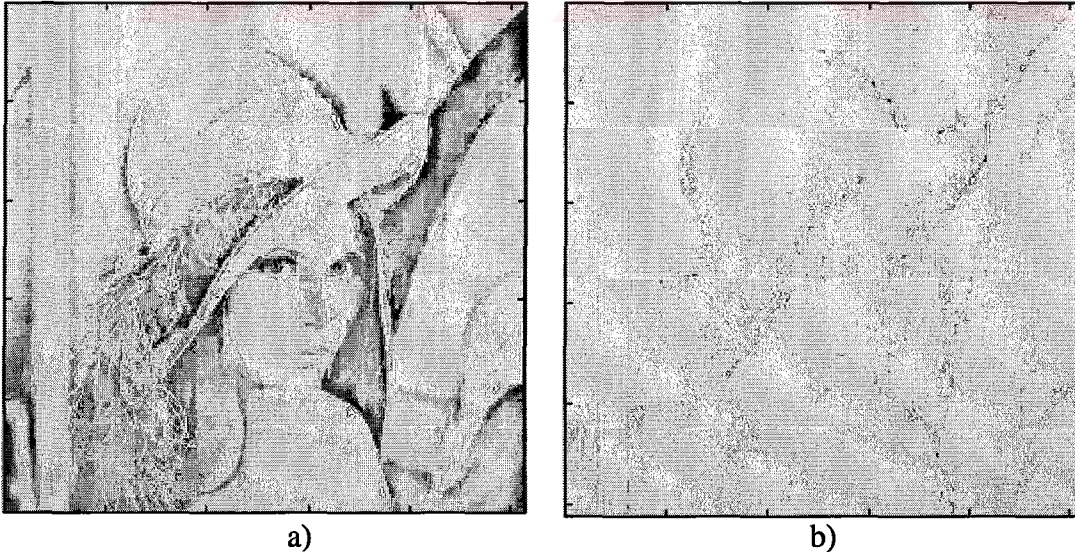
Şekil 4.19 İşlenmiş Uçak Optik sensörden elde edilmiş Havaalanı Görüntüsü BAQ Blok boyutuna bağlı SNR

Şekil 4.20 de ise pikseller arası yumuşak geçişleri olan yani komşu piksellerin birbiriyle ilişkili olduğu Lena görüntüsüne ilişkin BAQ Blok boyutuna bağlı SNR değişimi gösterilmektedir.



Şekil 4.20 Lena için BAQ blok dağılımına bağlı SNR

Lena gibi pikseller arası ilişkili doğal görüntülerde sıkıştırma performansının piksellerin konumlarına bağlı olduğu ve farklı bir doğal görüntü için farklı sonuçlar vereceği görülmektedir. Diğer ilgi çekici sonuç ise 8x8 Blok boyutlu 2 Bit BAQ için gerçekleşen SNR'nın kabul edilebilir bozulmaya sahip olmasıdır. Görüntüler üzerinde yapılan incelemelerde bu tür doğal görüntülerde blok boyutu büyüdüğünde yeni oluşan görüntüdeki blok etkisinin belirgin olmasıdır. Şekil 4.21 a' da 2 Bit BAQ SNR= 25.28 dB için elde edilmiş Lena görüntüsü, b' de ise yapılan hata gösterilmiştir.



Şekil 4.21 8x8 Blok boyutlu 2 Bit BAQ için a)Elde edilmiş Lena SNR=25.28 dB, b)Yapılan Hata (Orijinal görüntüden farkı)

Tablo 4.1 BAQ 2 Bit Algoritması Performans Analizi

Örnek Veri 256x256	Blok Boyutu	MSE	NMSE	SNR (dB)	PSNR(dB)	CR
İşlenmemiş YAR 1 I Bileşeni	8x8	1001	0.053	12.7	18.1	3.76
	16x16	1848	0.099	10.04	15.46	3.93
	32x32	2249	0.12	9.19	14.61	3.984
	64x64	2527	0.135	8.68	14.1	3.996
	128x128	2682	0.143	8.43	13.84	3.999
	256x256	2853	0.152	8.16	13.57	3.999
İşlenmemiş YAR 1 Q Bileşeni	8x8	1169	0.067	11.71	17.45	3.76
	16x16	2186	0.13	8.99	14.73	3.93
	32x32	2785	0.16	7.94	13.68	3.98
	64x64	3211	0.19	7.32	13.06	3.996
	128x128	3300	0.19	7.2	12.94	3.999
	256x256	3397	0.195	7.07	12.98	3.999
İşlenmemiş YAR 2 I Bileşeni	8x8	1070	0.053	12.68	17.83	3.76
	16x16	2044	0.102	9.87	15.03	3.938
	32x32	2541	0.127	8.93	14.08	3.984
	64x64	2681	0.143	8.42	13.84	3.996
	128x128	2948	0.148	8.28	13.43	3.999
	256x256	3046	0.153	8.14	13.29	3.999
İşlenmemiş YAR 2 Q Bileşeni	8x8	993	0.052	12.76	18.16	3.76
	16x16	1963	0.104	9.8	15.2	3.93
	32x32	2758	0.146	8.32	13.72	3.984
	64x64	2978	0.158	7.99	13.39	3.996
	128x128	3106	0.165	7.8	13.2	3.999
	256x256	3102	0.165	7.81	13.21	3.999
İşlenmiş Uydu YAR L.A	8x8	469	0.077	11.14	21.41	3.76
	16x16	311	0.051	12.9	23.17	3.93
	32x32	434	0.071	11.47	21.75	3.984
	64x64	1575	0.25	5.86	16.16	3.999
	128x128	4535	0.74	1.27	11.56	3.999
İşlenmiş Uydu YAR Gürültülü Hollanda	8x8	121	0.0088	20.54	27.27	3.464
	16x16	1265	0.091	10.38	17.1	3.93
	32x32	3435	0.248	6.04	12.77	3.98
	64x64	4944	0.357	4.46	11.18	3.996
	128x128	4803	0.347	4.58	11.31	3.999
İşlenmiş Uçak YAR Havaalanı	8x8	186	0.023	16.31	25.43	3.76
	16x16	466	0.058	12.32	21.44	3.93
	32x32	745	0.093	10.28	19.4	3.98
	64x64	986	0.123	9.07	18.19	3.996
	128x128	2110	0.265	5.76	14.88	3.999
İşlenmiş Uçak Optik Havaalanı	8x8	990	0.06	12.04	18.17	3.764
	16x16	1119	0.07	11.51	17.63	3.938
	32x32	1425	0.089	10.47	16.59	3.984
	64x64	2313	0.145	8.36	14.48	3.996
	128x128	1272	0.08	10.95	17.09	3.999
Lena	8x8	57	0.002	25.28	30.54	3.464
	16x16	127	0.0065	21.83	27.09	3.93
	32x32	248	0.012	18.92	24.18	3.984
	64x64	432	0.022	16.52	21.77	3.996
	128x128	1128	0.058	12.34	17.6	3.999

Tablo 4.2 BAQ 3 Bit Algoritması Performans Analizi

Örnek Veri 256x256	Blok Boyutu	MSE	NMSE	SNR (dB)	PSNR(dB)	CR
İşlenmemiş YAR 1 I Bileşeni	8x8	128	0.0068	21.63	27.05	2.56
	16x16	490	0.026	15.8	21.22	2.63
	32x32	720	0.038	14.13	19.55	2.65
	64x64	901	0.048	13.16	18.58	2.66
	128x128	971	0.052	12.83	18.256	2.666
İşlenmemiş YAR 1 Q Bileşeni	8x8	177	0.01	19.91	25.65	2.56
	16x16	650	0.037	14.26	20.003	2.63
	32x32	977	0.056	12.49	18.23	2.65
	64x64	1314	0.075	11.2	16.94	2.66
	128x128	1355	0.078	11.06	16.8	2.666
İşlenmemiş YAR 2 I Bileşeni	8x8	193	0.0009	20.16	25.28	2.56
	16x16	646	0.032	14.87	20.02	2.63
	32x32	949	0.047	13.2	18.35	2.65
	64x64	1066	0.053	12.7	17.85	2.66
	128x128	1232	0.062	12.07	17.22	2.666
İşlenmemiş YAR 2 Q Bileşeni	8x8	135	0.007	21.42	26.82	2.56
	16x16	552	0.029	15.31	20.71	2.63
	32x32	1050	0.055	12.51	17.91	2.65
	64x64	1204	0.064	11.92	17.32	2.66
	128x128	1270	0.067	11.69	17.09	2.666
İşlenmiş Uydu YAR L.A	8x8	178	0.029	15.33	25.62	2.56
	16x16	249	0.041	13.86	24.152	2.63
	32x32	270	0.044	13.52	23.81	2.65
	64x64	535	0.087	10.55	20.84	2.66
	128x128	1720	0.28	5.48	15.77	2.666
İşlenmiş Uydu YAR Gürültülü Hollanda	8x8	20	0.0014	28.31	35.04	2.56
	16x16	388	0.028	15.51	22.24	2.63
	32x32	1470	0.106	9.72	16.45	2.65
	64x64	2881	0.208	6.8	13.53	2.66
	128x128	3092	0.223	6.5	13.22	2.66
İşlenmiş Uçak YAR Havaalanı	8x8	205	0.025	15.89	25.01	2.56
	16x16	129	0.016	17.89	17.89	2.63
	32x32	278	0.034	14.56	23.68	2.65
	64x64	450	0.056	12.47	21.59	2.66
	128x128	773	0.097	10.12	19.24	2.666
İşlenmiş Uçak Optik Havaalanı	8x8	455	0.028	15.42	21.54	2.56
	16x16	539	0.034	14.68	20.81	2.63
	32x32	919	0.057	12.36	18.49	2.65
	64x64	511	0.032	14.91	21.04	2.66
	128x128	478	0.03	15.2	21.33	2.666
Lena	8x8	17	8.86E-04	30.52	35.77	2.56
	16x16	59	0.003	25.16	30.42	2.63
	32x32	40	0.002	26.82	32.07	2.65
	64x64	142	0.007	21.34	26.59	2.66
	128x128	113	0.0058	22.34	27.6	2.666

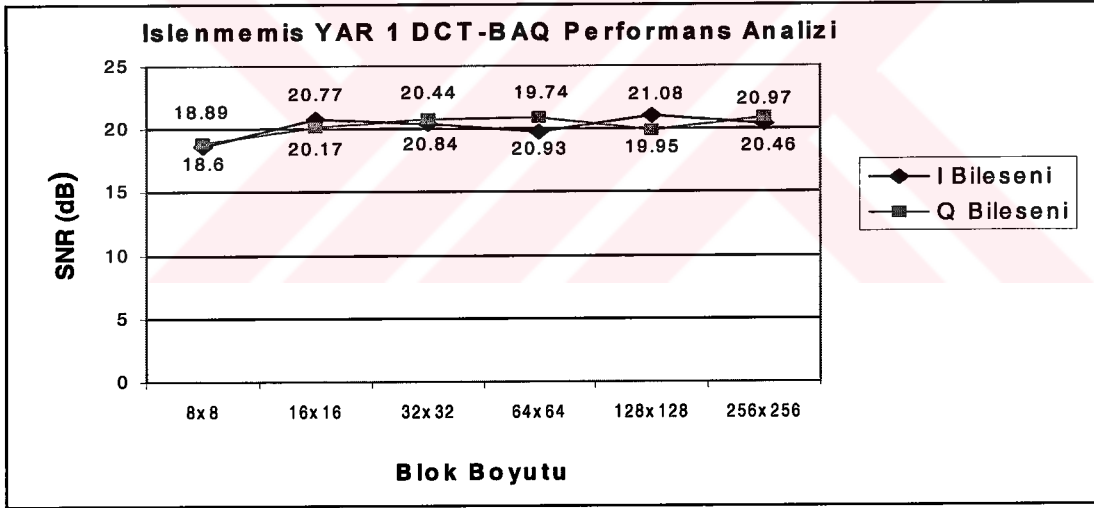
Tablo 4.3 BAQ 4 Bit Algoritması Performans Analizi

Örnek Veri 256x256	Blok Boyutu	MSE	NMSE	SNR (dB)	PSNR(dB)	CR
İşlenmemiş YAR 1 I Bileşeni	8x8	11	5.95E-04	32.24	37.66	1.93
	16x16	54	0.0029	25.36	30.78	1.98
	32x32	141	0.0075	21.21	26.63	1.99
	64x64	233	0.012	19.03	24.45	1.999
	128x128	267	0.014	18.43	23.85	1.999
İşlenmemiş YAR 1 Q Bileşeni	8x8	11	6.15E-04	32.1	37.84	1.93
	16x16	106	0.006	22.13	27.86	1.98
	32x32	249	0.014	18.43	24.17	1.996
	64x64	441	0.025	15.94	21.68	1.999
	128x128	468	0.026	15.69	21.43	1.999
İşlenmemiş YAR 2 I Bileşeni	8x8	9	4.67E-04	33.3	38.44	1.93
	16x16	120	0.006	22.2	27.35	1.98
	32x32	264	0.013	18.75	23.9	1.996
	64x64	329	0.016	17.8	22.95	1.999
	128x128	424	0.021	16.7	21.85	1.999
İşlenmemiş YAR 2 Q Bileşeni	8x8	11	5.77E-04	32.38	37.78	1.93
	16x16	73	0.0038	24.12	29.51	1.98
	32x32	304	0.016	17.91	23.3	1.996
	64x64	397	0.021	16.75	22.14	1.999
	128x128	439	0.023	16.31	21.7	1.999
İşlenmiş Uydu YAR L.A	8x8	94	0.015	18.13	28.41	1.93
	16x16	136	0.022	16.47	26.76	1.98
	32x32	143	0.023	16.28	26.57	1.996
	64x64	161	0.026	15.77	26.07	1.999
	128x128	854	0.14	8.53	18.81	1.999
İşlenmiş Uydu YAR Gürültülü Hollanda	8x8	58	0.004	23.77	30.5	1.93
	16x16	63	0.004	23.41	30.14	1.98
	32x32	559	0.04	13.92	20.65	1.996
	64x64	1691	0.122	9.12	15.84	1.999
	128x128	2079	0.15	8.22	14.95	1.999
İşlenmiş Uçak YAR Havaalanı	8x8	76	0.0095	20.21	29.33	1.93
	16x16	112	0.014	18.53	27.65	1.98
	32x32	131	0.016	17.82	26.95	1.996
	64x64	113	0.014	18.48	27.6	1.999
	128x128	241	0.03	15.18	24.3	1.999
İşlenmiş Uçak Optik Havaalanı	8x8	347	0.022	16.6	22.7	1.93
	16x16	396	0.024	16.02	22.15	1.98
	32x32	602	0.037	14.2	20.33	1.996
	64x64	419	0.026	15.77	21.9	1.999
	128x128	1028	0.064	11.88	18	1.999
Lena	8x8	143	0.0074	21.29	26.54	1.93
	16x16	121	0.0062	22.02	27.28	1.98
	32x32	151	0.0078	21.07	26.32	1.996
	64x64	63	0.0032	24.87	30.13	1.999
	128x128	126	0.065	21.86	27.12	1.999

Tablo 4.1, Tablo 4.2 ve Tablo 4.3’ de sırasıyla 2, 3 ve 4 Bit BAQ algoritması için SNR ile birlikte Bölüm 4.3.1’ de açıklanan diğer performans kriterlerine göre başarımlar gösterilmiştir.

4.4 Dönüşüm Domeninde DCT-BAQ Algoritmasının Performans Analizi

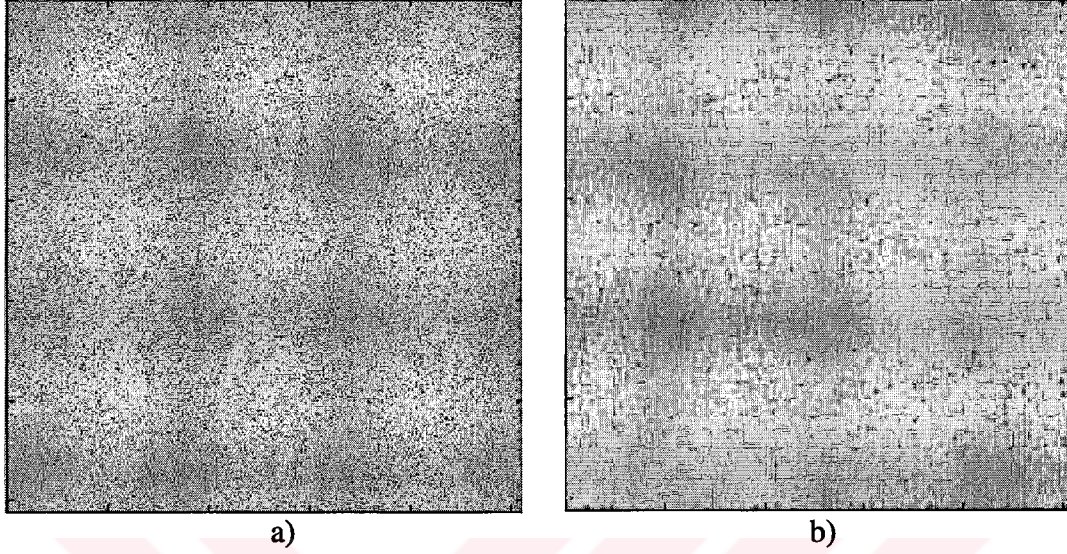
Dönüşüm domeninde katsayıların dağılımı Gauss Dağılımına uyuyorsa, dönüşüm katsayıları BAQ ile kodlanabilir. Bu çalışmada dönüşüm domeni için Ayrık Kosinüs Dönüşümü kullanılmıştır. DCT seçilmesinin nedeni, işlem kolaylığı sağlaması ve dönüşüm domeninde Karhunen-Leove Dönüşümünden sonra en iyi enerji yoğunlaşmasına sahip olmasıdır. İşlenmemiş YAR verilerinin dönüşüm sonrası katsayıların dağılımı 3.2.1’ de anlatıldığı gibi yumuşak geçişli doğal görüntülere göre farklılık gösterir. Dönüşüm domeni katsayıları için yüksek frekanslı bileşenlerde de önemli ölçüde enerjinin olduğu görülmektedir. Bu yüzden JPEG gibi doğal görüntüler için geliştirilmiş sıkıştırma algoritmalarında kullanılan standart tarama sisteminin kullanılması mümkün değildir. Bunun için, 3.2.2’ de anlatılan DC bileşenin ayrı kodlanmasıyla oluşturulan DCT-BAQ algoritması kullanılmıştır. Algoritmaya ait Blok diagramı Şekil 3.6’ te verilmiştir. Şekil 4.22’ de İşlenmemiş YAR 1 verisine ilişkin I ve Q bileşenlerinin DCT-BAQ algoritması için blok boyutuna bağlı SNR’ ın değişimi gösterilmektedir.



Şekil 4.22 İşlenmemiş YAR 1 verisi I ve Q bileşenleri DCT-BAQ için Blok boyutuna bağlı SNR’ ın değişimi

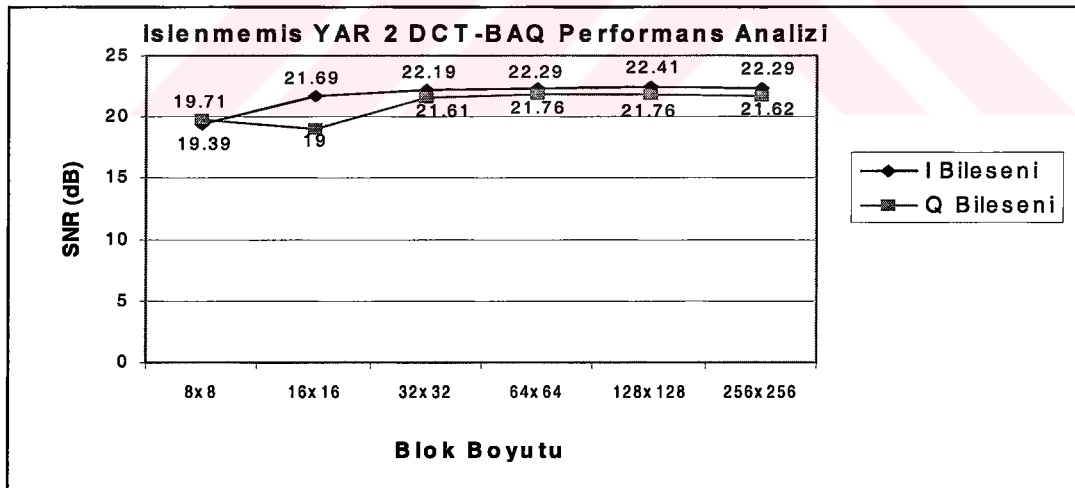
Şekil 4.22’ den de görüldüğü gibi İşlenmemiş YAR 1 verisi için I ve Q bileşenleri benzer başarımlar göstermiştir. SNR 1,5 dB’ lik bir alan içinde değişir. SNR’ ın değişimi blok boyutuna bağlı değildir. Bunun sonucu olarak her bir blok boyutu için DC bileşen dışındaki dönüşüm katsayılarının benzer istatistiksel özelliklere sahip olduğu düşünülebilir. Sıkıştırma oranının yüksek olması için büyük boyutlu veri blokları tercih edilebilir. Büyük boyutların tercih edilebilmesi için ise, eşzamanlı

uygulamalarda uzun veri bloğunun saklanabilmesi gerekir. Şekil 4.23 a'da İşlenmemiş YAR 1 I bileşeninin 128x128 blok boyutunda DCT-BAQ, Şekil 4.23 b' de ise JPEG kodlanıp tekrar elde edilmiş görüntüleri görülmektedir. JPEG İşlenmemiş YAR verileri için uygun değildir



Şekil 4.23 İşlenmemiş YAR 1 verisi I Bileşeni CR=4 a) DCT-BAQ SNR=20.08 dB
b) JPEG SNR=13.4 dB

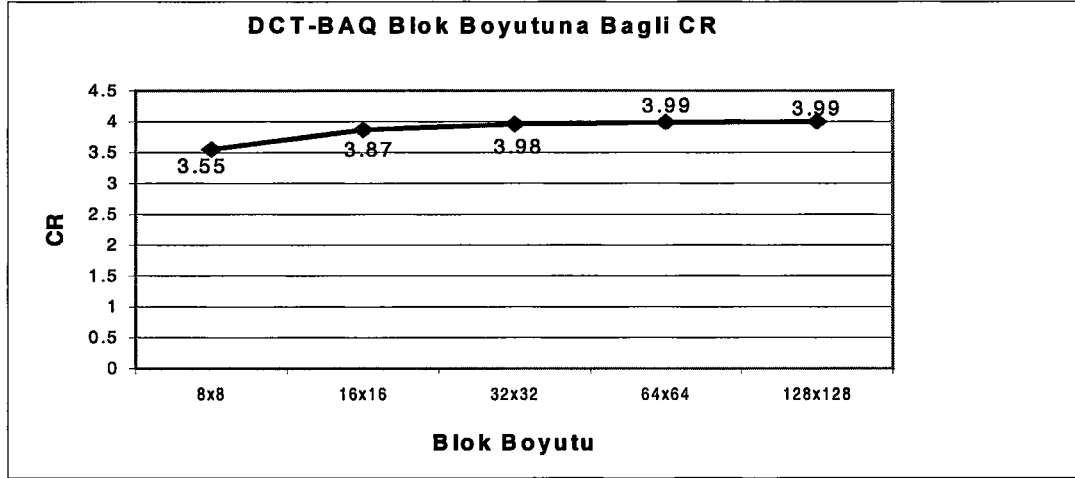
Şekil 4.24' de ise İşlenmemiş YAR 2 verisinin DCT-BAQ performans sonuçları görülmektedir.



Şekil 4.24 İşlenmemiş YAR 2 verisi I ve Q bileşenleri DCT-BAQ için SNR' ın blok boyutuna bağlı değişimi

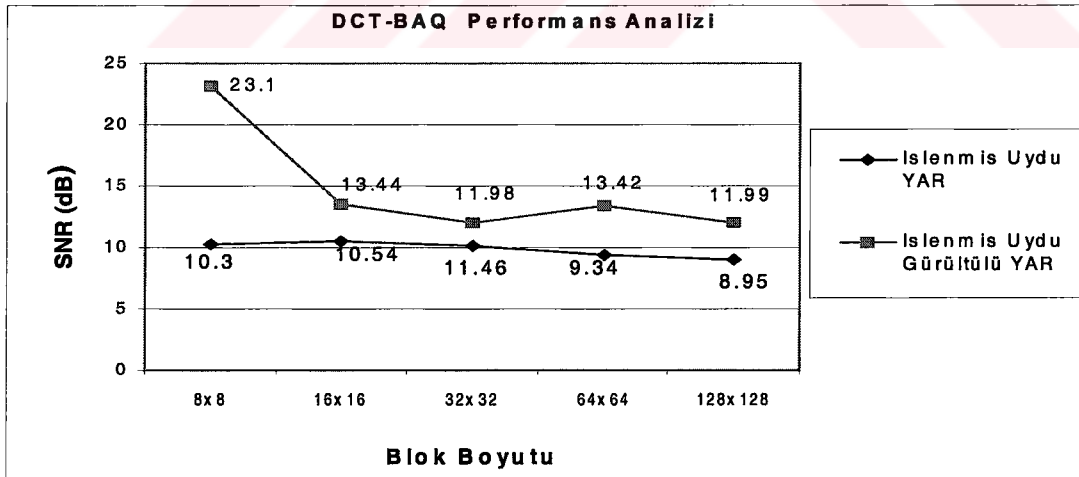
Şekil 4.24'te de görüldüğü gibi DCT-BAQ ile sıkıştırılan İşlenmemiş YAR 2 verisi için SNR, İşlenmemiş YAR 1 verisinde olduğu gibi yaklaşık 1.5 dB' lik bir değişim gösterir. 32x32' lik blok boyutundan sonra fazlaca bir değişim görülmemektedir.

Şekil 4.25 'te ise DCT-BAQ algoritmasında sıkıştırma oranının değişimi görülmektedir. Her bir blok için dönüşüm katsayılarının DC bileşenin de standart sapmayla birlikte ayrı olarak kodlanması sıkıştırma oranını küçültmektedir. Fakat bu etki büyük blok boyutları için gözardı edilebilecek bir husustur. Özellikle 32x32 lik blok boyutundan itibaren bu etki gözardı edilebilir



Şekil 4.25 DCT-BAQ için sıkıştırma oranının blok boyutuna bağlı değişimi

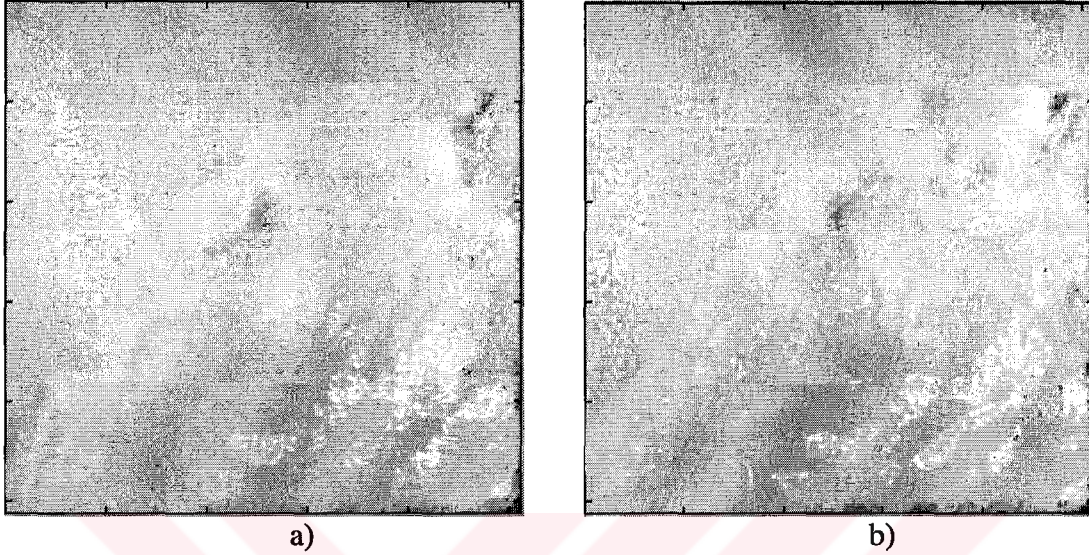
Şekil 4.25'te ise DCT-BAQ algoritması ile sıkıştırılan İşlenmiş Uydu L.A. ve Gürültülü Hollanda görüntüleri için SNR'ın blok boyutuna bağlı değişimi görülmektedir. Gürültülü verinin sıkıştırma performansının daha iyi olduğu gözükmedir.



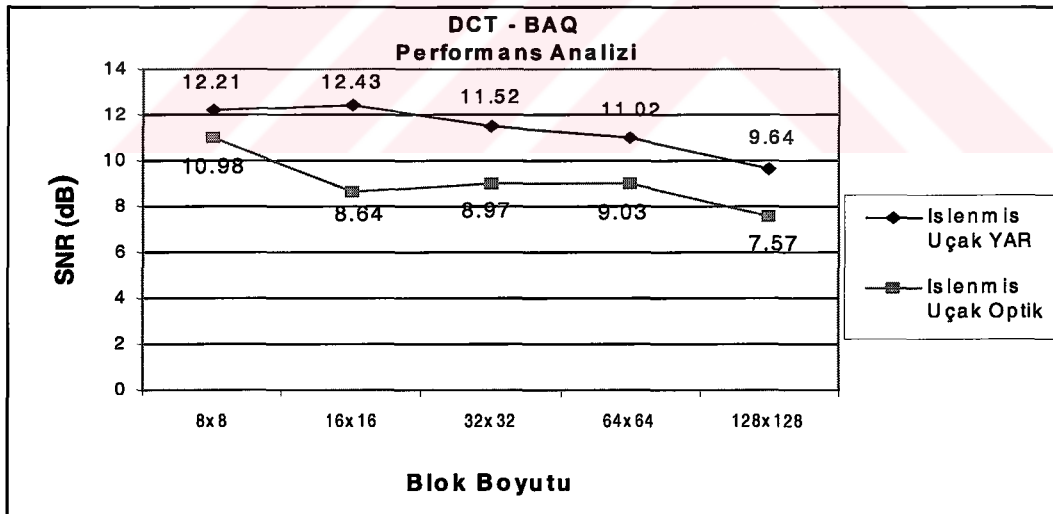
Şekil 4.26 İşlenmiş Uydu YAR L.A. ve İşlenmiş Uydu Gürültülü Hollanda verisinin DCT-BAQ performans Analizi

İşlenmiş Uydu Gürültülü YAR Hollanda verisinde blok boyutu 8x8 gibi küçük alındığında blok içindeki dağılım işlenmemiş YAR verisi gibi olmakta ve böylece DCT-BAQ için SNR 23,1 dB gibi yüksek çıkmaktadır. Şekil 4.4.6 a' da İşlenmiş

Gürültülü YAR Hollanda verisi 8x8 blok boyutu için DCT-BAQ, b' de ise JPEG ile kodlanıp tekrar oluşturulmuş görüntüler görülmektedir. Bu veri için JPEG ile SNR'ın biraz daha iyi ve 1.3 dB kadar yüksek olduğu görülmektedir. JPEG ile pikseller arası geçişin yumuşadığı yani yüksek frekanslı bileşenlerin daha çok kaybolduğu gözlenmektedir.

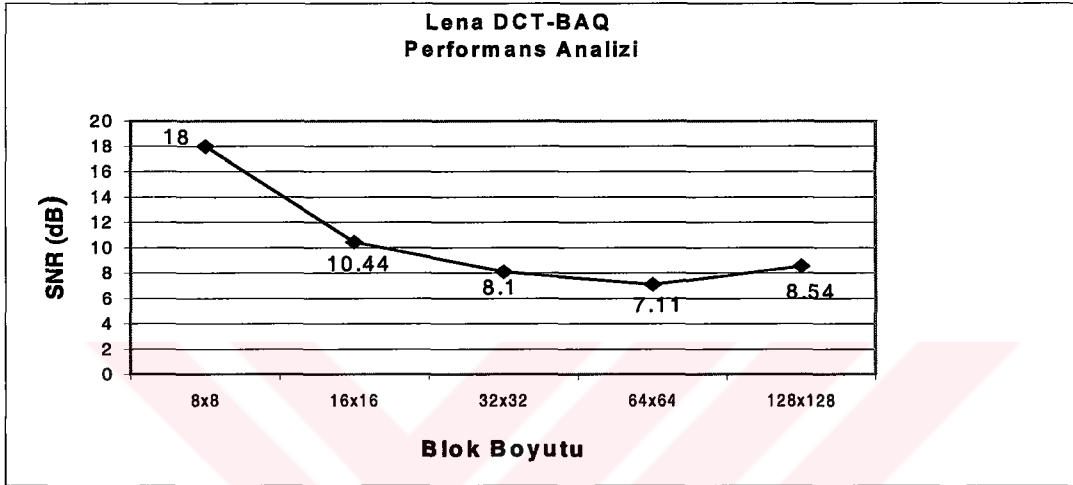


Şekil 4.27 İşlenmiş Gürültülü Uydu YAR Hollanda Görüntüsü CR=3.55 a)DCT-BAQ SNR=23.1 dB b)JPEG SNR=24.4



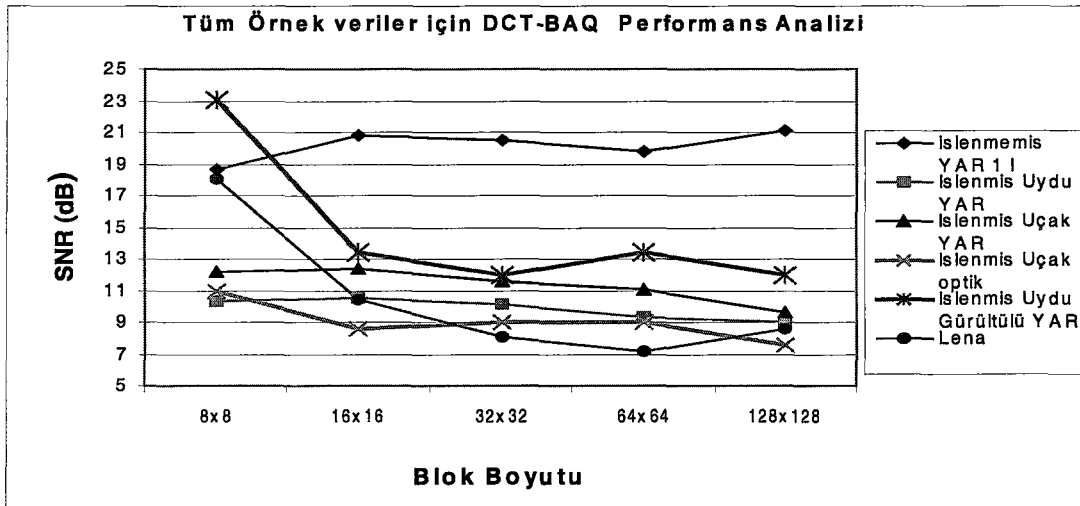
Şekil 4.28 İşlenmiş Uçak YAR ve Optik verilerinin DCT-BAQ Performans Analizi

Şekil 4.28’ de ise İşlenmiş Uçak YAR ve Optik sensörlere ait DCT-BAQ için SNR’ın blok boyutuna bağlı değişimi görülmektedir. Burada görüldüğü gibi İşlenmiş Uçak YAR verisinin aynı yere ait İşlenmiş Uçak Optik görüntüsüne göre daha yüksek SNR sağladığı görülmektedir. Yine İşlenmemiş YAR verisinin aksine blok boyutunun artmasıyla SNR’ ın azaldığı görülmektedir. Bu da işlenmiş data için beklenen bir durumdur.Şekil 4.29 ‘ de ise pikseller arası geçişleri yumuşak doğal bir görüntü olan Lena için DCT-BAQ performans analizi görülmektedir.



Şekil 4.29 Lena için DCT-BAQ Performans Analizi

Şekil 4.30’ da ise tüm örnek veriler için DCT-BAQ performans analizi gösterilmiştir. Şekilde de görüldüğü gibi DCT-BAQ için en iyi performans sonucu İşlenmemiş YAR verisi için elde edilmiştir. Bununla birlikte YAR verilerinin optik verilere göre daha iyi performans sergilediği görülebilir.



Şekil 4.30 Tüm örnek veriler için DCT-BAQ performans Analizi

Tablo 4.4 Dönüşüm domeninde DCT-BAQ Algoritması Performans Analizi

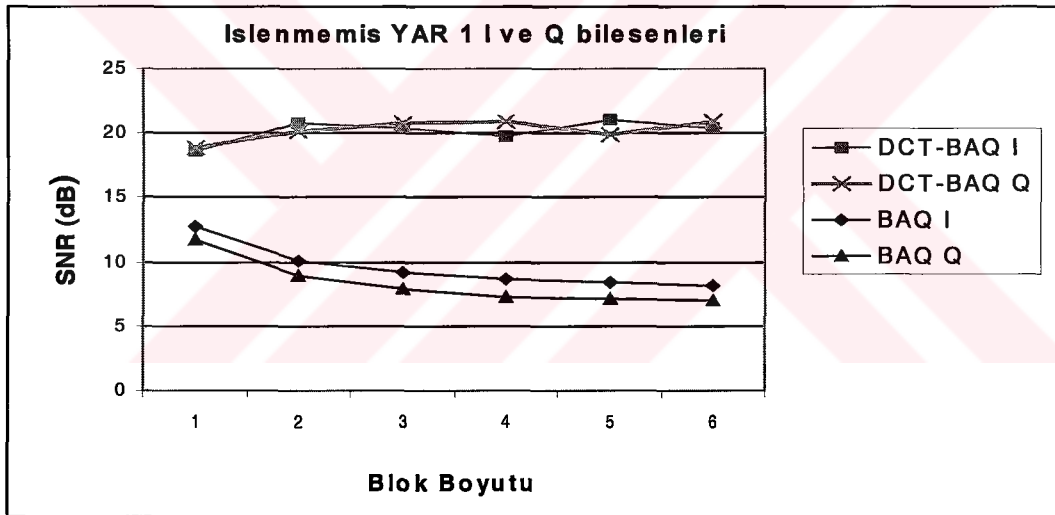
Örnek Veri 256x256	Blok Boyutu	MSE	NMSE	SNR (dB)	PSNR(dB)	CR
İşlenmemiş YAR 1 I Bileşeni	8x8	258	0.013	18.6	24.02	3.55
	16x16	156	0.0083	20.77	26.19	3.87
	32x32	168	0.009	20.44	25.86	3.96
	64x64	198	0.0106	19.74	25.16	3.99
	128x128	146	0.0077	21.08	26.5	3.998
	256x256	168	0.0089	20.46	25.88	3.999
İşlenmemiş YAR 1 Q Bileşeni	8x8	223	0.012	18.89	24.64	3.55
	16x16	166	0.0095	20.17	25.92	3.87
	32x32	143	0.0082	20.84	26.58	3.96
	64x64	139	0.008	20.93	26.67	3.992
	128x128	175	0.01	19.95	25.7	3.998
	256x256	138	0.0079	20.97	26.71	3.999
İşlenmemiş YAR 2 I Bileşeni	8x8	228	0.011	19.39	24.54	3.55
	16x16	135	0.0067	21.69	26.84	3.87
	32x32	120	0.006	22.19	27.34	3.968
	64x64	117	0.005	22.29	27.44	3.99
	128x128	113	0.0057	22.41	27.56	3.99
	256x256	117	0.0058	22.29	27.48	3.999
İşlenmemiş YAR 2 Q Bileşeni	8x8	200	0.01	19.71	25.11	3.55
	16x16	235	0.0125	19	24.4	3.87
	32x32	129	0.0069	21.61	27	3.96
	64x64	125	0.0066	21.76	27.16	3.99
	128x128	125	0.0066	21.76	27.16	3.998
	256x256	129	0.0068	21.62	27.02	3.999
İşlenmiş Uydu YAR L.A.	8x8	567	0.093	10.3	20.56	3.55
	16x16	537	0.088	10.54	20.82	3.87
	32x32	587	0.096	10.12	20.44	3.96
	64x64	704	0.115	9.33	19.65	3.99
	128x128	775	0.127	8.95	19.23	3.998
İşlenmiş Uydu YAR Gürültülü Hollanda	8x8	68	0.0048	23.098	29.83	3.555
	16x16	626	0.045	13.44	20.16	3.87
	32x32	873	0.063	11.98	18.71	3.96
	64x64	628	0.045	13.42	20.14	3.992
	128x128	872	0.063	11.99	18.72	3.998
İşlenmiş Uçak YAR Havaalanı	8x8	478	0.059	12.21	21.34	3.55
	16x16	454	0.057	12.43	21.56	3.87
	32x32	560	0.07	11.52	20.64	3.96
	64x64	628	0.078	11.02	20.14	3.992
	128x128	864	0.108	9.64	18.76	3.998
İşlenmiş Uçak Optik Havaalanı	8x8	1263	0.079	10.98	17.11	3.55
	16x16	2167	0.136	8.64	14.77	3.87
	32x32	2010	0.126	8.97	15.09	3.96
	64x64	1983	0.125	9.03	15.15	3.99
	128x128	2774	0.174	7.57	13.69	3.998
Lena	8x8	307	0.0158	18	23.25	3.55
	16x16	1749	0.09	10.44	15.7	3.87
	32x32	2996	0.154	8.10	13.36	3.96
	64x64	3767	0.194	7.11	12.36	3.992
	128x128	2711	0.139	8.54	13.79	3.998

Tablo 4.4’ de ise DCT-BAQ algoritması performans analizi SNR’ ın yanında Bölüm 4.1’ de anlatılan diğer performans kriterleriyle birlikte gösterilmiştir.

4.5 Zaman Domeninde BAQ ve Dönüşüm Domeninde DCT-BAQ algoritmalarının birbiriyle karşılaştırılması

BAQ algoritması YAR sistemlerinde pratikte uygulanan bir algoritmadır. İşlenmemiş YAR verilerinin sıkıştırılması için yeni bir algoritma önerildiğinde, İşlenmemiş YAR verilerinin sıkıştırılmasında standart kabul edilen, BAQ ile karşılaştırılmaları, yeni algoritmanın avantajlarını ortaya koymak için kullanılabilecek bir yoldur. Bu amaçla Bölüm 4.2’ de anlatılan örnek verilerin performansları BAQ ve DCT-BAQ algoritmaları için karşılaştırmalı olarak gösterilmiştir.

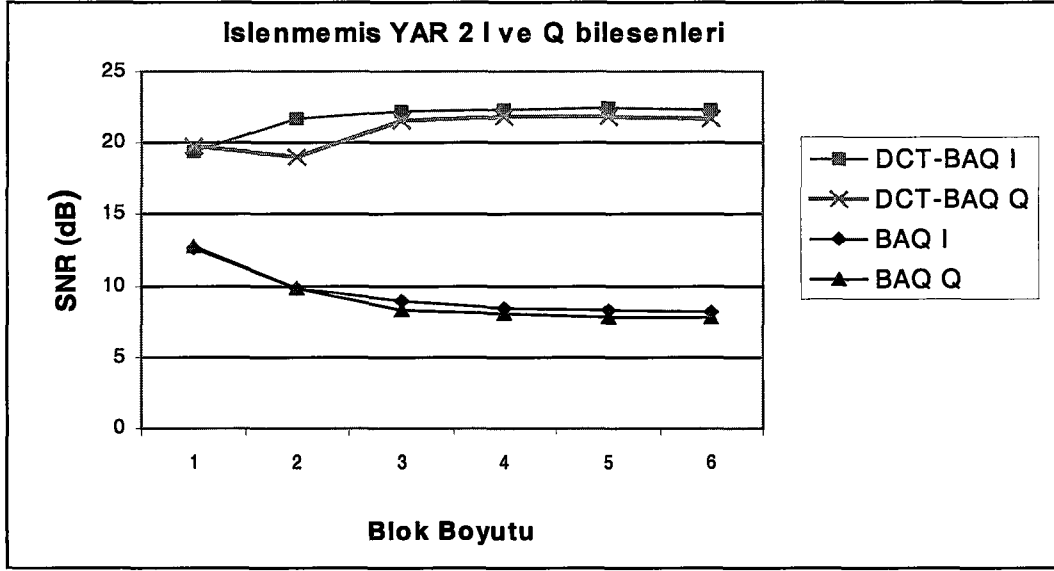
Şekil 4.31’ de İşlenmemiş YAR 1 I ve Q bileşenlerinin BAQ ve DCT-BAQ için performansı birlikte gösterilmiştir.



Şekil 4.31 İşlenmemiş YAR 1 I ve Q bileşenlerinin BAQ ve DCT-BAQ algoritmaları performans analizi

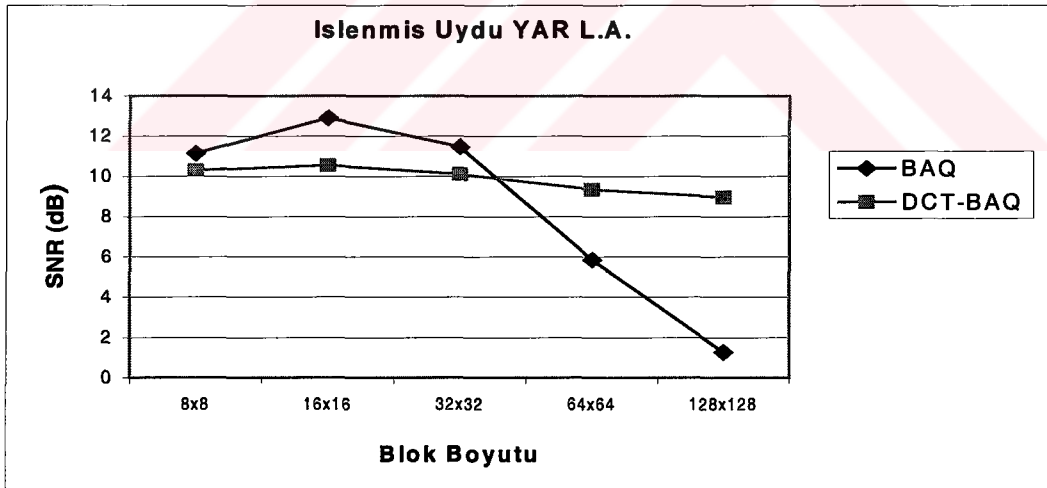
Şekil 4.31’ de görüldüğü gibi DCT-BAQ algoritması yaklaşık 10 dB daha iyi SNR sağlar. BAQ algoritması için blok boyutunun artmasına bağlı olarak SNR’ da genel bir düşüş görülürken DCT-BAQ için blok boyutunun artmasıyla genel bir yükseliş görülmektedir. Ayrıca DCT-BAQ algoritmasının başarımını Blok boyutuyla az ilişkili olduğu söylenebilir. Bu ise dönüşüm domenindeki DC bileşen dışındaki katsayıların dağılımının blok boyutuna bağlı olmaksızın Gauss dağılımlı olduğunu gösterir.

Şekil 4.32' de İşlenmemiş YAR 2 verisi için BAQ ve DCT-BAQ algoritmalarının performansı birlikte gösterilmiştir.



Şekil 4.32 İşlenmemiş YAR 2 verisi için BAQ ve DCT-BAQ algoritmaları performans analizi

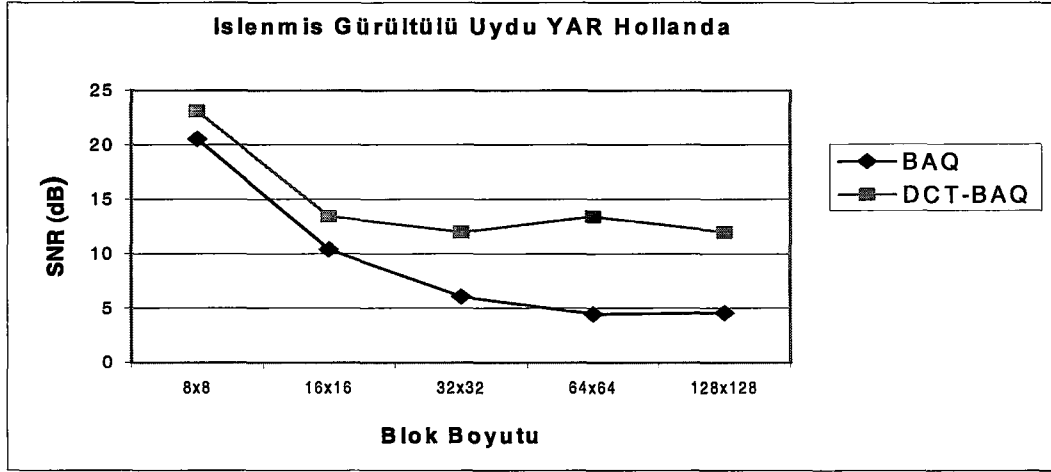
Şekil 4.32' de görülen, İşlenmemiş YAR 2 verisinin, İşlenmemiş YAR 1 verisiyle benzer bir performans gösterdiğiğidir. Şekil 4.33' de ise İşlenmiş Uydu YAR L.A. verisi için BAQ ve DCT-BAQ performans analizi birlikte gösterilmiştir.



Şekil 4.33 İşlenmiş Uydu YAR L.A için BAQ ve DCT-BAQ performans analizi

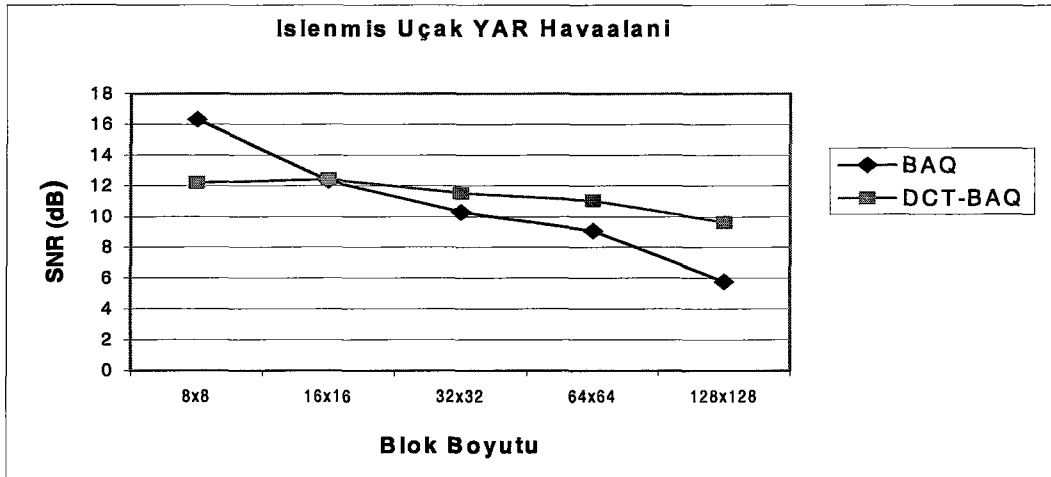
Şekil 4.33' de görülen İşlenmiş YAR L.A. görüntüsü için DCT-BAQ algoritmasının blok boyutuyla daha az ilişkili olduğudur. Bununla birlikte düşük blok boyutları için BAQ performansı DCT-BAQ performansına göre daha iyidir. Bu ise İşlenmiş YAR verisinin birbiriyle daha ilişkili piksellerden oluşması gerçeğinin bir sonucudur.

Şekil 4.34' de ise İşlenmiş Gürültülü YAR Hollanda görüntüsüne ilişkin performans analizi görülmektedir. Düşük blok boyutları için BAQ ve DCT-BAQ birbirlerine yakın sonuçlar vermiştir. Bu da görüntünün gürültülü yapısıyla ilişkilendirilebilir. Yüksek blok boyutları için ise DCT-BAQ, BAQ algoritmasına göre daha iyi sonuç verir.



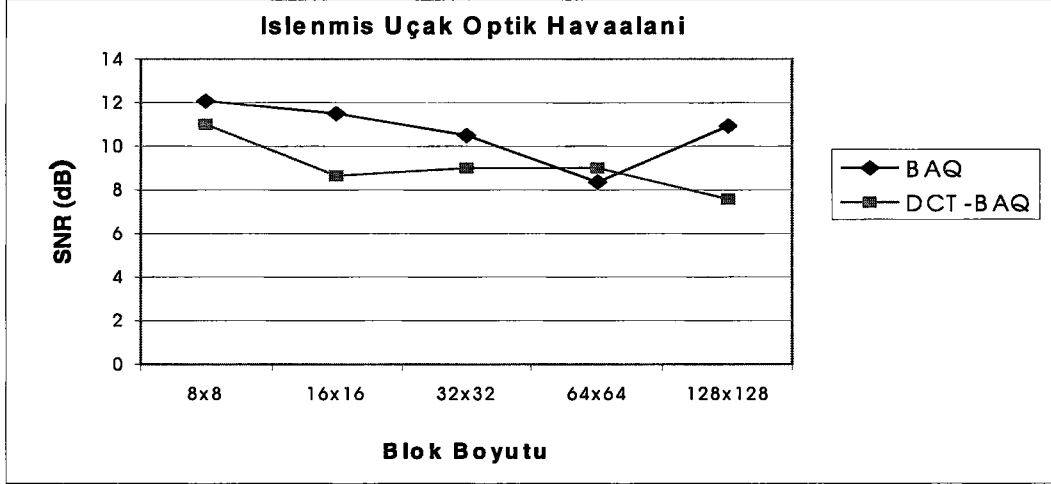
Şekil 4.34 İşlenmiş Gürültülü Uydu YAR Hollanda için BAQ ve DCT-BAQ performans analizi

Şekil 4.35'te ise Uçak YAR Havaalanı görüntüsü için BAQ ve DCT-BAQ performans analizi görülmektedir. DCT-BAQ için blok boyutuyla SNR'da yavaş bir azalma görülmekle birlikte, düşük blok boyutlarında BAQ'nun iyi sonuç verdiği görülmektedir. Bu sonuç düşük blok boyutlarında İşlenmiş Uçak YAR Havaalanı verisinin gauss dağılımına uyduğunu gösterir.



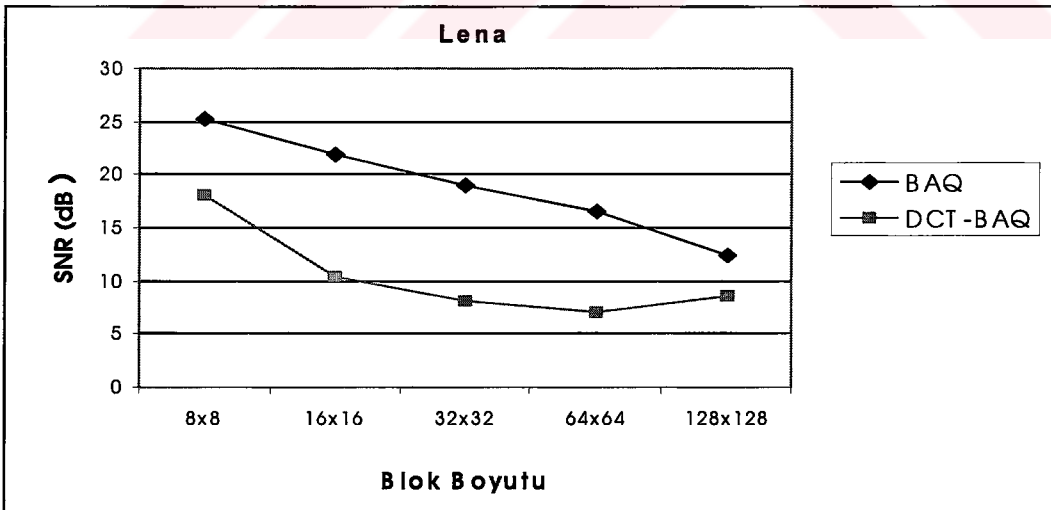
Şekil 4.35 İşlenmiş Uçak YAR Havaalanı için BAQ ve DCT-BAQ performans analizi

YAR sensörü ile optik sensörden elde edilen veriler arasında kullanılacak sıkıştırma algoritmalarının farkını ortaya koymak için aynı havaalanı verisinin optik sensörden elde edilmiş görüntüsü kullanılmıştır. Bu görüntü için BAQ ve DCT-BAQ algoritmaları performans analizi Şekil 4.36' da gösterilmiştir.



Şekil 4.36 İşlenmiş Uçak Optik Havaalanı görüntüsü için BAQ ve DCT-BAQ performans analizi

Optik sensörden elde edilen veriler için SNR BAQ için daha iyi sonuç vermiştir. Bu beklenen bir durumdur. Aynı bölgeye ait işlenmiş YAR verisinde DCT-BAQ nun büyük blok boyutlarında daha iyi sonuç vermesi ise YAR verisinde piksellerin optik verilere göre birbirleriyle daha ilişkisiz olmalarının bir sonucu olarak yorumlanabilir.

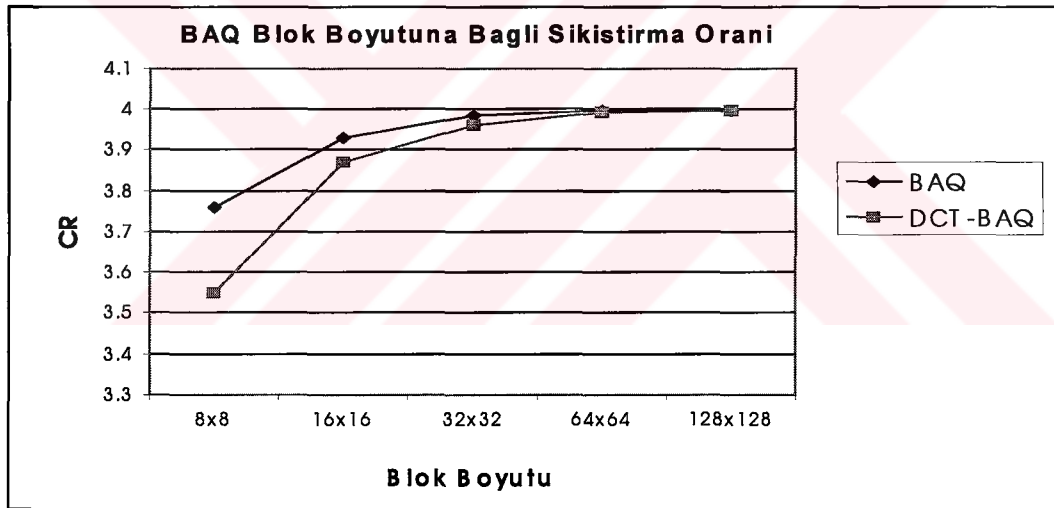


Şekil 4.37 Lena için BAQ ve DCT BAQ algoritmaları performans analizi

Şekil 4.37' de ise pikseller arası geçişleri ilişkili Lena görüntüsüne ait BAQ ve

DCT-BAQ için performans analizi görülmektedir. BAQ algoritması Lena için DCT-BAQ' ya göre daha iyi sonuç vermiştir. İki algoritma içinde blok boyutlarının artmasıyla SNR' da düşme görülür. Küçük blok boyutları için her iki algoritma da iyi sonuç verir.

Şekil 4.38 de ise BAQ ve DCT-BAQ algoritmalarının yukarıda açıklanan performans sonuçları için blok boyutuna bağlı sağladıkları sıkıştırma oranları gösterilmiştir. BAQ algoritmasında alınan her bloğa ilişkin standart sapma 8 bitle kodlanır. DCT-BAQ' da ise her bir bloğa ilişkin dönüşüm katsayılarının DC bileşeni de, standart sapmayla birlikte 8 Bitle kodlanarak gönderilir. Bu durum DCT-BAQ'da gerçekleşen sıkıştırma oranının BAQ' ya göre biraz daha az olması sonucunu doğurur. Bu etki yüksek blok boyutları için gözardı edilebilir. DCT-BAQ nun SNR performansının blok boyutundan bağımsız olduğu düşünülürse, DCT dönüşüm katsayılarının DC bileşeninin 8 Bitle kodlanarak gönderilmesi sonucu sıkıştırma oranına getirdiği olumsuz etki göz ardı edilebilir.



Şekil 4.38 BAQ ve DCT-BAQ algoritmaları için sıkıştırma oranının blok boyutuna bağlı değişimi

Şekil 4.38' de görüldüğü gibi 32x32' lik bloklarda sıkıştırma oranı birbirine çok yakın değerler almaktadır. Bu blok boyutundan büyük blok boyutları için sıkıştırma oranı BAQ ve DCT-BAQ algoritmaları için eşit kabul edilebilir.

5. SONUÇLAR VE ÖNERİLER

Tezde İşlenmemiş YAR verisi için dönüşüm domeninde sıkıştırma algoritması geliştirilmiştir. İşlenmemiş YAR verisi doğal resimlere göre birbiriyle ilişkisiz komşu verilerden oluşur. Dinamik aralığı çok yüksek olan İşlenmemiş YAR verisi dönüşüm domeni katsayılarının da dinamik aralığı yüksek olur. Dönüşüm domeni katsayılarının bu yüksek dinamik aralığı dönüşüm domeninde sıkıştırma performansını olumsuz yönde etkiler. İşlenmemiş YAR verileri için dönüşüm domeninde sıkıştırma, dönüşüm domeni katsayılarının dinamik aralığının azaltılması ile sağlanabilir.

Yapılan dönüşüm domeni sıkıştırma algoritması için bir sinüsoidal dönüşüm olan Ayrık Kosinüs Dönüşümü kullanılmıştır. Ayrık kosinüs dönüşümü katsayılarının dinamik aralığının azaltılması DC bileşenin bu katsayı kümesinden çıkarılarak yerine sıfır eklenilmesi yoluyla sağlanmıştır. Bu şekilde oluşan yeni dönüşüm katsayıları kümesi Gauss dağılımını uyan dinamik aralığı azaltılmış. Katsayılar kümesidir. Oluşturulan dönüşüm domeni katsayıları Blok Adaptif Kuantalama (BAQ) yardımıyla kuantalamıştır.. DCT-BAQ algoritmasıyla yapılan İşlenmemiş YAR verisi için sıkıştırma performansı bu tür veriler için standart kabul edilen BAQ ile karşılaştırılmış ve yaklaşık 10 dB'lik bir artış sağlanmıştır. Diğer yandan tezde dönüşüm domeni algoritmalarının doğal görüntülerle (Birbirleriyle ilişkili komşu pikseller) İşlenmemiş YAR verilerinin (Birbirleriyle ilişkisiz komşu pikseller) sıkıştırılmasında kullanılırken nasıl değişmesi gerektiği açıklanmıştır. Bununla birlikte optik sensörler ile YAR sensörlerinden elde edilen işlenmiş veriler için sıkıştırma algoritmasının farklı olması gerektiği açıklanmıştır. YAR sensöründen elde edilen verinin sahip olduğu noktasal gürültülü yapı, optik sensörlerle elde edilmiş bu tür görüntülerden farkını gösterir.

DCT-BAQ algoritması BAQ algoritmasına göre sıkıştırma oranı bakımından biraz daha kötü performans gösterir. Ancak 32x32 ve daha yüksek blok boyutları için BAQ ve DCT-BAQ' nun sağladığı sıkıştırma oranı eşit kabul edilebilir. DCT-BAQ

algoritmasında BAQ'nun aksine İşlenmemiş YAR verileri SNR'ı için blok boyutuna bağlı değildir,. BAQ için blok boyutu arttığında SNR azalır.

DCT-BAQ algoritması için kod sözcüklerinin atanması Huffman kodlayıcı yardımıyla yapılır. Huffman kodlayıcı veri dizisindeki farklı sembollerin dizide bulunma olasılığına göre uygun uzunlukta kod kelimelerinin atanmasını sağlar. Bunun amacı veri dizisinde sıkça tekrarlanan sembollerin diğer sembollere göre daha az bitle kodlanmasıdır. Böylece kayıpsız bir kodlama verimli şekilde yapılmış olur.ş olur.

Özet olarak İşlenmemiş YAR verisinin sıkıştırılmasında, verinin dinamik aralığının büyük, komşu veriler arasındaki ilişkinin ise küçük olması, doğal görüntülerde kullanılan sıkıştırma algoritmaların, İşlenmemiş YAR verisi için uygun olamayacağı anlamı taşır.

Geliştirilen DCT-BAQ algoritmasının SNR'ı BAQ ya göre İşlenmemiş YAR verileri için çok daha iyi performans vermiştir.

Sonuç olarak BAQ skaler kuantalayıcıdır. DCT-BAQ algoritmasında BAQ yerine vektörel kuantalayıcı kullanılırsa (VQ) performansın daha iyi gerçekleşmesi beklenir.

İşlenmemiş YAR verilerinin sıkıştırılması sensörün bulunduğu uydu platformundan uzaktan algılama yer istasyonu ile olan aşağı link hızını azaltmak için kullanılır. Bu durumda Uydu da kullanılabilecek sıkıştırma algoritmalarının gerçek zamanda çalışması gerekir. Bu ise Uydu platformunda yapılacak sıkıştırma işlemlerinin hızlı yani az hesaplama gerektiren algoritmalar olması gerektiğini doğurur. İşlenmemiş YAR verilerinin sıkıştırılması ihtiyacı sadece aşağı link data hızını azaltmak için gerekli değildir. Özellikle uzaktan algılama yer istasyonları için İşlenmemiş YAR verilerinin arşivlenmesi içinde etkin sıkıştırma algoritmalarına ihtiyaç duyulur. Arşivlemenin yanı sıra alınan İşlenmemiş YAR verisinin doğrudan son kullanıcılara iletilmesini sağlayan linkler içinde link yükünü azaltmak için sıkıştırma algoritmalarına ihtiyaç duyulur.

Uydu platformunda sıkıştırma algoritmaları uydunun kaynaklarıyla sınırlı olarak gerçekleştirileceği için basit hesaplanabilir olmalıdırlar. Ancak yer istasyonlarında kullanılacak sıkıştırma algoritmaları daha kompleks olabilir.

Çalışmada zaman domeninde BAQ, Dönüşüm domeninde bir sinüsoidal dönüşüm olan DCT kullanılmıştır. Bununla birlikte hem zamanda hemde frekans da ayırıcılığı değişebilen dalgacık dönüşümü yardımıyla bulunan filtre katsayılarıyla İşlenmemiş YAR verisini etkin biçimde uygun alt bandlara ayrılması ve her bir alt bandın sahip olduğu enformasyona bağlı olarak uygun kod sözcükleriyle skaler veya vektörel kodlanması sonucu daha etkin sıkıştırma performansı sağlanabileceği düşünülebilir. Özellikle bu tür yüksek yoğunluklu hesaplama içeren algoritmalar uzaktan algılama yer istasyonlarında arşivleme ve İşlenmemiş YAR verisinin doğrudan son kullanıcıya linkler yardımıyla iletilmesi konularında kullanılabilir.



KAYNAKLAR

- [1] **Zeoli, G. W.**, 1976. A lower bound on the data rate for synthetic aperture radar, *IEEE Trans. Inform. Theory*, 22, 708.
- [2] **Evert P. and Attema, W.**, 1991. The active microwave instrument on-board the ERS-1 satellite, *Procc. IEEE*, 79, 791.
- [3] **Steinberg, B.S.**, 1987. A theory on the effect of hard limiting and other distortions upon the quality of microwave images, *IEEE Trans. Acoustics, Speech, Signal Proc.*, 35, 1462
- [4] **Benz, U., Strodl, K. and Moreira, A.**, 1995. A Compression of Several Algorithms for SAR Raw Data Compression, *IEEE Trans On Geoscience and Remote Sensing*, Vol 33, No 5, September 1995.
- [5] **Franceschetti, G., Pascazio, V., and Schirinzi, G.**, 1991. Processing of signum coded SAR signal: theory and experiments, *IEE Proc.*, Part F, 138, 192.
- [6] **Franceschetti, G., Merello, S., and Tesauro, M.**, 1999. processing of phase quantized SAR signal: theory and experiments, *IEEE Trans. AEROSP. Electron. Syst.*, 31, 227.
- [7] **Lim, S, J.**, 1990 Two Dimentional Signal and Image Processing, Printice-Hall
- [8] **Curlander, J. and. Mcdonough, R.**, 1991. Synthetic Aperture Radar Systems and Signal Processing, John Wiley & Sons, Inc.
- [9] **Kwok, R. and Johnston, W.**, 1989 . Block Adaptive Quantization of Magellan SAR Data, *IEEE Trans. On Geoscience and Remote Sensing*. Vol. 27 No 4 July 1989.
- [10] **Syood, K.**, 1996 Introduction to Data Compression, Morgan Kaufmann Publishers, San Francisco.
- [11] **Rabbani, M. and Jones, P.**, 1991 Digital Image Compression Techniques, Spie Optical Engineering Press Washington, USA.
- [12] **Franceschetti, G and Lanari, P.**, 1999 Synthetic Aperture Radar Processing, CRS Press Washington, USA.
- [13] **Wijmans, W and Armbruster, P.**, 1996 Data Compression Techniques for Space Applications *DASIA' 96* Rome, Italy

- [14] **Ruliang, Y, Yanfei, W and Ping, B.**, 1998 Raw Data Compression Technology Study in Spaceborne SAR, *Eurosar' 98* , p 283, Germany.
- [15] **Odoux, B, Deschaux, M and Planes, J.**, 1998 SAR Raw Data On-Board Compression with Frequency Filtering, *Eurosar' 98* p 521, Germany
- [15]http://callisto.si.ushelb.ca/~cartel/rso/Applets/Applet_donnees_brutes/exec_eng.htm
- [17] **Ulaby, W., Moore, R. K., and Fung, A.K.**, 1982 Microwave Remote Sensing :Active and Passive, vol.2:Radar remote Sensing and Surface Scattering and Emission, Addison Wesley, Reading MA.



ÖZGEÇMİŞ

N. Gökhan Kasapoğlu 1973’de İstanbul’da doğdu.1995 yılında Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümünü bitirdi. 1996 yılında İTÜ Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Programı’nda yüksek lisans çalışmalarına başladı. Halen İTÜ Elektronik ve Haberleşme Mühendisliği Bölümü Elektromagnetik Alanlar ve Mikrodalga Tekniği Anabilim Dalında araştırma görevlisi olarak çalışmaktadır.

