

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI**

YÜKSEK LİSANS TEZİ

**DÜŞÜK KARMAŞIKLIĞA SAHİP HAREKET KESTİRİMİ DONANIM
MİMARİLERİ**

SEDA ÇETİN

KOCAELİ 2019

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

DÜŞÜK KARMAŞIKLIĞA SAHİP HAREKET KESTİRİMİ
DONANIM MİMARİLERİ

SEDA ÇETİN

Dr. Öğr. Üyesi Anıl ÇELEBİ
Danışman, Kocaeli Üniv.
Prof. Dr. Oğuzhan URHAN
Jüri Üyesi, Kocaeli Üniv.
Dr. Öğr. Üyesi Ramazan YENİÇERİ
Jüri Üyesi, İstanbul Teknik Üniv.

Tezin Savunulduğu Tarih: 10.07.2019



ÖNSÖZ VE TEŞEKKÜR

İlk olarak yüksek lisans eğitimim boyunca engin bilgisi ve motivasyonu ile beni destekleyen, sınırsız sabrı ile bu tezin araştırma ve yazımının her aşamasında bana yardımcı olan danışmanım Dr. Öğr. Üyesi Anıl ÇELEBİ'ye sonsuz teşekkürlerimi sunmak isterim.

Danışmanımın yanı sıra, teşvik edici ve anlayışlı yorumları için Dr. Öğr. Üyesi Aysun TAŞYAPI ÇELEBİ'ye teşekkür etmek isterim.

Bu tez çalışmasını EEEAG/115E921 numaralı araştırma projesi kapsamında finansal olarak destekleyen Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'na (TÜBİTAK) teşekkürlerimi sunarım.

Kocaeli Üniversitesi Tümüleşik Sistemler Araştırma Laboratuvarındaki hocam Ali Can KARACA'ya bilgilendirici yorumları ve samimi dostlukları için teşekkür ederim.

Son olarak, manevi desteklerini esirgemedikleri ve her ihtiyaç duyduğumda yanımda oldukları için kıymetli aileme ve eşim Yücel ÇETİN'e en derin duygularıyla teşekkür ederim.

Temmuz – 2019

Seda ÇETİN

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ	ii
TABLolar DİZİNİ	ii
SİMGELER VE KISALTMALAR DİZİNİ	v
ÖZET.....	vi
ABSTRACT	vii
GİRİŞ	1
1. VİDEO KODLAMA	2
2. HAREKET KESTİRİMİ.....	7
2.1. Düşük Bit Gösterimi Temelli Hareket Kestirimi Yöntemleri	8
2.1.1. Bir bit dönüşümü temelli HK yöntemi.....	9
2.1.2. Çarpmasız bir bit dönüşümü temelli HK yöntemi	10
2.1.3. Kısıtlanmış bir bit dönüşümü temelli HK yöntemi	11
2.1.4. Yerel ikili örüntü temelli HK yöntemleri.....	12
2.1.5. Seçimli Gray kodlama temelli HK yöntemi.....	14
3. DÜŞÜK KARMAŞIKLIĞA SAHİP HK YAKLAŞIMLARI İÇİN ÖNERİLEN DONANIM MİMARİLERİ	16
3.1. 1BT, MF-1BT ve C-1BT HK Yaklaşımları İçin Önerilen Donanım Mimarileri	16
3.2. LBP-1BT ve LBP-2BT HK Yaklaşımları İçin Önerilen Donanım Mimarileri	19
3.2.1. 1-bit derinliğinde LBP donanımı (LBP-1BT).....	19
3.2.2. 2-bit derinliğinde LBP donanımı (LBP-2BT).....	21
3.3. 1DF-2BT HK Yaklaşımı İçin Önerilen Donanım Mimarisi	22
3.4. SGC-2BT HK Yaklaşımı İçin Önerilen Donanım Mimarisi.....	25
4. BULGULAR VE TARTIŞMA	30
4.1. 1BT, MF-1BT, LBP-1BT ve LBP-2BT Yaklaşımları İçin Geliştirilen Donanım Mimarilerinin İncelenmesi	30
4.2. 1DF-2BT ve SGC-2BT Yaklaşımları İçin Geliştirilen Donanım Mimarilerinin İncelenmesi	34
5. SONUÇLAR VE ÖNERİLER	39
KAYNAKLAR	40
KİŞİSEL YAYIN VE ESERLER	44
ÖZGEÇMİŞ	45

ŞEKİLLER DİZİNİ

Şekil 1.1.	Video kodlama standartlarının tarihsel gelişimi [4].....	2
Şekil 1.2.	Genel bir H.264/AVC video kodlayıcı için blok diyagram [4].....	4
Şekil 1.3.	HEVC standardının genel kodlayıcı yapısı [6]	4
Şekil 1.4.	HEVC’de bölümlenme yapıları	5
Şekil 1.5.	Çerçeve içi tahminleme için PU boyutları	5
Şekil 1.6.	Çerçeveler arası tahminleme için kodlama ve tahmin birimi arasındaki ilişki	6
Şekil 2.1.	Foreman video dizisi örnek imgeleri a) Orijinal 8-bit derinlikli imge b) 1BT yaklaşımı c) MF-1BT yaklaşımı [21].....	11
Şekil 2.2.	Foreman video dizisi örnek imgeleri a) 8-bit derinlik b) Filtrelenen imge c) MF-1BT yaklaşımıyla elde edilen 1-bit imge d) CM [22].....	12
Şekil 2.3.	Örnek LBP konfigürasyonları LBP (P, R) a) LBP (4, 4) b) LBP (8, 4) c) LBP (8, 8) [28].....	13
Şekil 2.4.	3 bitlik düzlemlerin elde edilme süreci	15
Şekil 2.5.	SG temelli yöntemde bit düzlemlerinin seçilmesi	15
Şekil 3.1.	İkilileştirme ve uyumlama aşaması birlikte ele alındığında temel HK donanım mimarisi	16
Şekil 3.2.	Süzgeçleme aşamasında gerekli olan imge çerçevesi bölgeleri a) MF-1BT ve C-1BT yöntemleri için b) 1BT yöntemi için [36].....	17
Şekil 3.3.	Süzgeç donanım mimarileri a) MF-1BT ve C-1BT yöntemleri için b) 1BT yöntemi için [36].....	18
Şekil 3.4.	İkilileştirmenin son aşamasında kullanılan yapılar a) MF-1BT ve 1BT yöntemleri için, b) C-1BT yöntemi için [36]	19
Şekil 3.5.	LBP yöntemi için donanım mimarisi a) Donanımı gerçekleştirilen LBP (8, 8) konfigürasyonu b) İkilileştirme donanım mimarisi c) Uyumlama donanım mimarisi [28]	20
Şekil 3.6.	LBP-2BT yöntemi için donanım mimarisi a) Süzgeç donanım mimarisi b) İkilileştirmenin son aşamasında kullanılan yapı c) Uyumlama donanım mimarisi [37]	22
Şekil 3.7.	1DF-2BT yöntemi için ikilileştirme yapıları a) Süzgeçleme aşamasında gerekli olan imge çerçevesi bölgesi b) Süzgeç donanım mimarisi c) İkilileştirme son aşamada kullanılan yapı [21].....	24
Şekil 3.8.	1DF-2BT yönteminin uyumlama donanım mimarisi	25
Şekil 3.9.	8×8 imge bloğu için bit seçme örüntüsü a) [30]’da gösterilen Yöntemin bit seçme yapısı b) SGC-2BT yöntemi için 1.bit uzayı seçme örüntüsü c) SGC-2BT yöntemi için 2.bit uzayı [39].....	26
Şekil 3.10.	SGC-2BT için geliştirilen donanım mimarileri a) İkilileştirme birimi b) Uyumlama birimi [39].....	28
Şekil 4.1.	Farklı HK yaklaşımları için PSNR değerleri ile kaynak kullanımı arasındaki ilişki	32
Şekil 4.2.	Farklı HK yaklaşımları için PSNR değerleri ve enerji tüketimi arasındaki ilişki [37].....	33

TABLolar DİZİNİ

Tablo 4.1. 1BT, MF-1BT, LBP-1BT ve LBP-2BT yaklaşımları için HK performansları	31
Tablo 4.2. 1BT, MF-1BT, C-1BT, LBP-1BT ve LBP-2BT yaklaşımları için donanım kaynak kullanımı performansı.....	31
Tablo 4.3. 1BT, MF-1BT, LBP-1BT ve LBP-2BT yaklaşımları için güç ve enerji tüketimi sonuçları [37]	33
Tablo 4.4. 1DF-2BT ve SGC-2BT yaklaşımları için HK performansı incelemesi(dB) [39].....	34
Tablo 4.5. 1DF-2BT yöntemi için donanım kaynak kullanımı incelemesi [21].....	35
Tablo 4.6. 1DF-2BT yöntemi için güç ve enerji tüketimi incelemesi [21]	35
Tablo 4.7. SGC-2BT yöntemi için donanım kaynak kullanımı incelemesi [39].....	37
Tablo 4.8. SGC-2BT güç analizi sonuçları [39].....	38

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

ASIC	: Application Specific Integrated Circuits (Uygulamaya Özel Entegre Devre)
AVC	: Advanced Video Coding (Gelişmiş Video Kodlama)
BME	: Block Matching Estimation(Blok Eşlemeli Hareket Kestirimi)
BP	: Bit Plane (Bit Düzlemi)
BPM	: Bit Plane Matching (Bit Düzlem Uyumlama)
1BT	: One Bit Transform (Bir Bit Dönüşümü)
2BT	: Two Bit Transform (İki Bit Dönüşümü)
CB	: Current Block (Mevcut Blok)
CM	: Context Modeler (Bağlam Modelleyici)
CTU	: Coding Tree Unit (Kodlama Ağaç Ünitesi)
CU	: Coding Unit (Kodlama Ünitesi)
C-1BT	: Constrained One Bit Transform (Kısıtlanmış Bir Bit Dönüşümü)
1DF-2BT	: One Dimensional Filtering Based Two Bit Transform (Bir Boyutlu Filtreleme Temelli İki Bit Dönüşümü)
EXOR	: Exclusive OR (Özel Veya)
FF	: Flip Flop (Yaz-Boz)
FPGA	: Field Programmable Gate Array (Alanda Programlanabilir Kapı Dizisi)
HEVC	: High Efficiency Video Coding (Yüksek Verimli Video Kodlama)
LBP-1BT	: Local Binary Pattern Based One Bit Transform (Yerel İkili Örüntü Temelli Bir Bit Dönüşümü)
LBP-2BT	: Local Binary Pattern Based Two Bit Transform (Yerel İkili Örüntü Temelli İki Bit Dönüşümü)
LUT	: Look Up Table (Arama Tablosu)
MB	: Makro Blok
MC	: Motion Compansation(Hareket Dengeleme)
MF-1BT	: Multiplication Free One Bit Transform (Çarpmasız Bir Bit Dönüşümü)
MV	: Motion Vector (Hareket Vektörü)
MUX	: Multiplexer (Çoklayıcı)
NNMP	: Number of Non Matching Points (Eşleşmeyen Nokta Sayısı)
PB	: Prediction Block (Tahmin Bloğu)
PSNR	: Peak Signal to Noise Ratio (Tepe Sinyal Gürültü Oranı)
PU	: Prediction Unit (Tahmin Ünitesi)
SAD	: Sum of Absolute Differences (Mutlak Farklar Toplamı)
SGC-1BT	: Selective Gray Coded Based One Bit Transform (Seçimli Gri Kodlama Temelli Bir Bit Dönüşümü)
SGC-2BT	: Selective Gray Coded Based Two Bit Transform (Seçimli Gri Kodlama Temelli İki Bit Dönüşümü)

DÜŞÜK KARMAŞIKLIĞA SAHİP HAREKET KESTİRİMİ DONANIM MİMARİLERİ

ÖZET

Yüksek çözünürlüklü video uygulamalarının artması ile kodlama verimliliğine duyulan ihtiyaç da artmaktadır. Ultra yüksek çözünürlüklü televizyonlar (UHD TV), kameralar ve akıllı telefonlar gibi video kaydetme ve aktarma kabiliyeti olan cihazlar için güç tüketimi ve hafıza önemli kısıtlardır. Video kodlayıcılarda, en karmaşık görevi gerçekleştiren ve en fazla güç tüketen kısım hareket kestirimidir. Bu nedenle, verimli donanım mimarileri sunabilen düşük karmaşıklığa sahip hareket kestirimi yöntemleri geliştirilmiştir. Bu yöntemlerden biri de imgelerin 8 bit yerine daha düşük bit derinliğinde ifade edilmesidir. Literatürde bilinen düşük bit gösterimi temelli hareket kestirimi yaklaşımları için ikili imgelerin hazır olduğu kabul edilmiş ve sadece uyumlama aşamasını içeren donanım mimarileri önerilmiştir. Bu tez kapsamında hareket kestirimi yaklaşımlarına ait uyumlama aşamasına ek olarak ikilileştirme aşamasının donanım mimarileri gerçekleştirilmiştir. Elde edilen sonuçlardan anlaşılacağı üzere ikilileştirme donanım maliyeti uyumlamaya göre çok daha fazladır. Video kodlayıcının tamamı dikkate alındığında ikilileştirme aşamasının donanım maliyetini ihmal etmenin doğru bir kabul olmadığı ortaya çıkmaktadır.

Anahtar Kelimeler: Donanım Mimarisi, FPGA, Hareket Kestirimi, Video Kodlama.

EFFECTIVE HARDWARE ARCHITECTURES FOR LOW COMPLEXITY MOTION ESTIMATION

ABSTRACT

The need for coding efficiency is being increased with applications in which high resolution video processing is being performed. Power consumption and memory are important constraints for devices capable of recording and transmitting video, such as ultra-high definition televisions (UHD TVs), cameras and smartphones. In video encoders, motion estimation is the process which utilizes the most complex tasks and consumes most of the power. Therefore, low complexity motion estimation methods have been developed which can provide efficient hardware architectures. One of these methods is to represent images at a lower bit depth rather than 8 bits. It is accepted that binary images are ready for motion estimation approaches based on low bit representation, and hardware architectures including only the matching stage are proposed in the literature. Within the scope of this thesis, in addition to the matching stage of motion estimation approaches, the hardware architectures of the binarization stage were implemented. As can be seen from the results, the cost of binarization hardware is much higher than matching. It is revealed that ignoring the hardware cost of binarization stage is not an appropriate assumption when the whole video encoder architecture is taken into account.

Keywords: Hardware Architecture, FPGA, Motion Estimation, Video Coding.

GİRİŞ

Günümüzde, teknolojinin hızla ilerlemesi ile kamera, cep telefonu vb. elektronik cihazların videoları işleme ve yayınlama kabiliyetine ihtiyaç artmaktadır. Ham videolar, sınırlı bant genişliğine sahip kanallar üzerinden iletmenin neredeyse imkânsız olduğu muazzam büyüklüktedir [1]. Bu nedenle, etkin video sıkıştırma teknikleri ile çekilen videoları iletebilmek için boyutunu azaltmak gerekmektedir.

MPEG-4 ve H.264/AVC kodlama standartları, düşük bit hızlı video aktarımı uygulamalarından HDTV ye kadar geniş bir uygulama alanı sunmaktadır. H.264/AVC kodlama standartları yüksek çözünürlükte yeterli olsa bile 2K,4K,8K çözünürlükteki teknolojiler için gerekli performansı sağlayamamaktadır [2]. Bu sebeple H.264/AVC yerini H.265/HEVC ye bırakmaya başlamıştır.

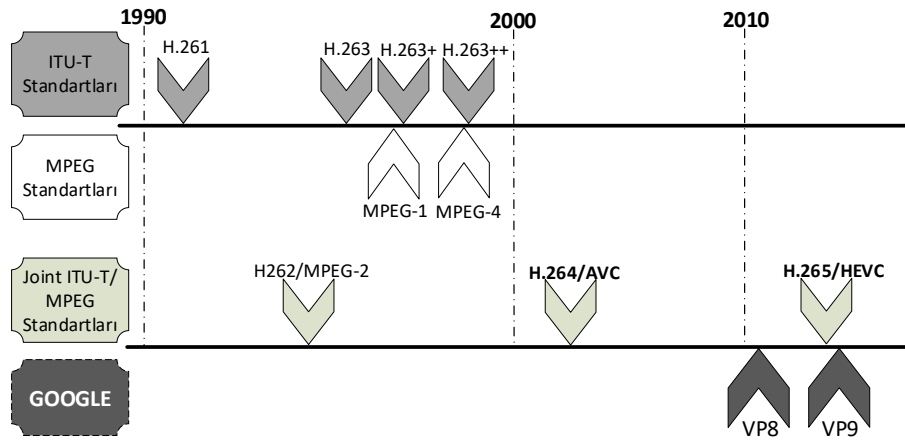
Benzer video sıkıştırma teknolojilerinde olduğu gibi H.264/AVC ve HEVC’de de hareket kestirimi (HK) video kodlamanın önemli bir bölümünü oluşturur. Çözünürlüğün de artması ile hareket kestiriminin sebep olduğu artan güç tüketimi ve işlem yükünü azaltmak amacıyla farklı hareket kestirimi yaklaşımları geliştirilmiştir.

Tez kapsamında ilk olarak video kodlama standartlarına ve en yeni video kodlama standardı olan HEVC’nin kodlama yapısına genel hatları ile değinilmiştir. Bölüm 2’de hareket kestirimi alanında yapılan eski çalışmalardan özellikle de tez kapsamında literatürde donanımı gerçekleştirilen düşük bit gösterimi temelli HK yaklaşımlarından bahsedilmiştir. Bölüm 3’te literatürde yaygın olarak bilinen düşük bit gösterimi temelli HK yaklaşımları için hem ikilileştirme hem de uyumlama aşamasını içeren donanım mimarileri gerçekleştirilmiştir. Elde edilen deneysel sonuçlar ile gerçekleştirilen donanım mimarileri HK performansı, donanım maliyeti, güç ve enerji tüketimi açısından ayrıntılı olarak karşılaştırılmıştır.

1. VİDEO KODLAMA

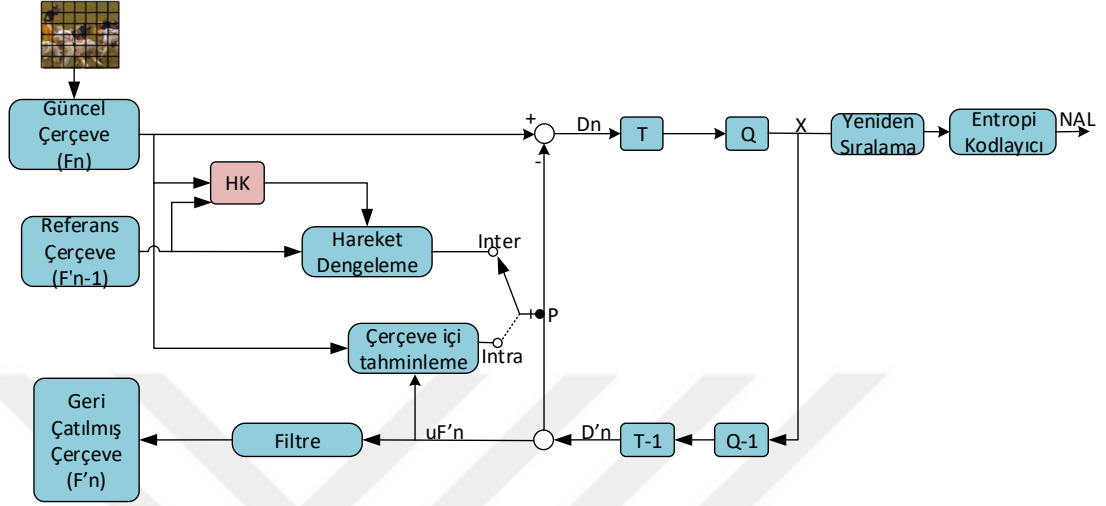
Günümüzde teknolojinin hızla ilerlemesi ile birlikte yüksek çözünürlüklü video uygulamaları, gerçek zamanlı dijital uygulamalar ve video aktarımı gibi işlemlere ihtiyaç artmaktadır [3]. Videolarda görüntü çözünürlüğünün artması ve dolayısıyla aktarımda artan bant genişliği ihtiyaçları beraberinde fazla güç tüketimi ve kaynak kullanımını getirmektedir. Saniyede 30 çerçevede 1280 ×720 (HD) piksel çerçeve boyutuna sahip ve her piksele 24 bit olacak şekilde bir video dizisi için 663,5 Mbps iletim hızına ihtiyaç duyulurken, güncel Wi-Fi(802.11g) standardında bile 54 Mbps iletim hızı mevcuttur. Daha yüksek çözünürlükteki bir video için bu değer daha da artmaktadır. Saniyede 30 çerçevede 3840×2160 (4K) piksel çerçeve boyutuna sahip ve her piksele 24 bit olacak şekilde bir video dizisi için 5972 Mbps iletim hızına ihtiyaç duyulmaktadır. Elde edilen değerlerden anlaşılacağı üzere çözünürlük artışı ile birlikte video verisinin etkin biçimde kodlanması kaçınılmaz hale gelmiştir.

Video kodlama standartlarının ve bu standartlara ait formatların tarihsel gelişimi Şekil 1.1’de gösterilmektedir. Video endüstrisinin gelişimi için Uluslararası Telekomünikasyon Birliği (ITU) ve Uluslararası Standartlar Organizasyonu (ISO) bir dizi standart geliştirmiştir. ITU H.26x standartlarını telekomünikasyon uygulamaları üzerine yoğunlaşarak hazırlamıştır. ISO ise MPEG-x serilerini ortaya koymuştur.



Şekil 1.1. Video kodlama standartlarının tarihsel gelişimi [4]

1990'ların sonunda MPEG-4 geliştirilmiştir ardından H.263 standart haline getirilmiştir. H.264/AVC, 2003 yılında yayınlanmıştır [5]. 2013 yılında Yüksek Verimli Video Kodlama (HEVC) standardının yayınlandığı görülmektedir [6].



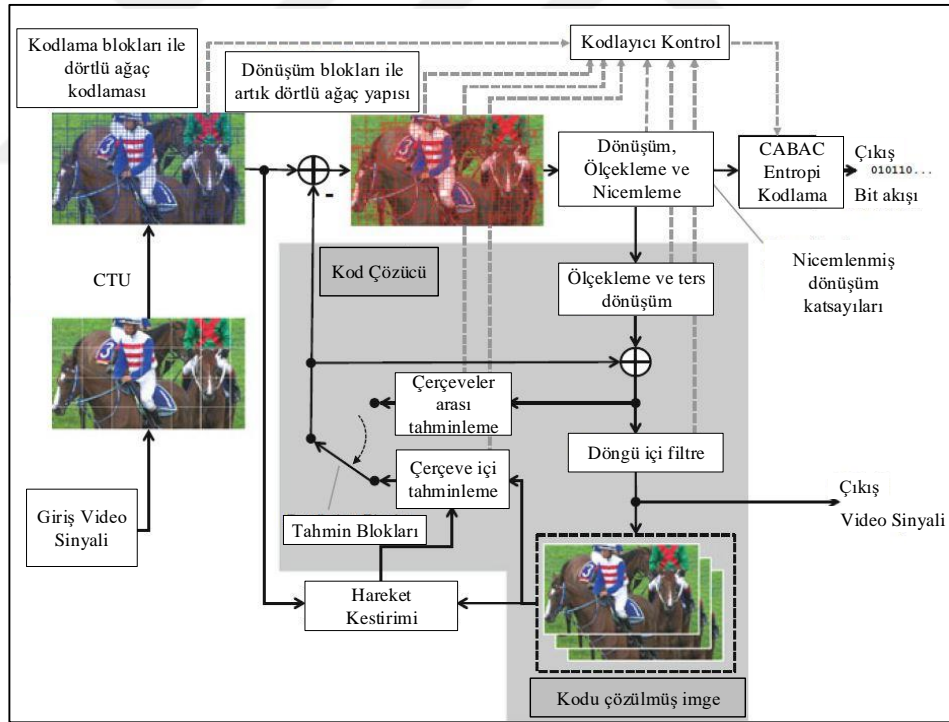
Şekil 1.2. Genel bir H.264/AVC video kodlayıcı için blok diyagramı [4]

Bir H.264/AVC video kodlayıcının genel blok diyagramı Şekil 1.2'de gösterilmektedir. Kodlayıcı yapısının başlangıcında giriş çerçevesi (F_n), makro bloklara (MB) ayrılmaktadır. Her MB çerçeve-içi ve çerçeveler-arası tahminlerde kullanılmaktadır. Geri çatılmış çerçeveleri oluşturmak için Şekil 1.2'de 'P' olarak gösterilen makro bloklar içindeki bloklar kullanılır. Çerçeve-içi tahminde P, o anki dilim (slice) içerisindeki kodlanmış, çözülmüş ve yeniden oluşturulmuş örnek değerlerinden oluşturulur. Çerçeveler-arası tahminde kullanılması halinde, tahmin bloğu uygun bir referans çerçeve (F_{n-1}) seçilerek HK (hareket kestirimi) ve hareket dengeleme blokları ile oluşturulur.

Artıklık bloğunu oluşturmak için tahmin bloğu (P), güncel bloktan çıkarılır. Bu blok uygun bir blok dönüşüm kodlaması (T) kullanılarak frekans düzlemine dönüştürülür. Dönüşüm kodlaması sonrası ilgili blok nicemlenir (Q). Nicemlenen dönüşüm katsayıları yeniden sıralanarak düzenlenir ve kayıpsız sıkıştırma özelliği olan entropi kodlama yapısı kullanılarak sıkıştırılır. Entropi kodlanmış katsayılar, makro-blok içindeki her bloğu çözmek için gerekli olan bilgilerle birlikte sıkıştırılmış bit dizinini oluşturur ve bu bit dizinleri iletim ve depolama amaçlı olarak ağ iletim katmanına gönderilir (Network Abstraction Layer - NAL).

Video kodlama sistemlerinde daha yüksek sıkıştırma başarımı için çerçevelerin referans olarak kullanılması gerekmektedir. Çerçevelerin referans olarak kullanılabilmesi için de makro-blok içindeki her bir bloğun çözülmesi ve yeniden oluşturulması gerekmektedir. Bundan dolayı, Şekil 1.2’de olarak gösterilen dönüşüm (X) katsayıları, fark bloğu D'n’i oluşturmak için ölçekleme işlemi (Q-1) yapılmaktadır. Ardından fark bloğunu oluşturmak için ters dönüştürülür (T-1). Tahmin bloğu, yeniden yapılandırılmış uF'n (u, filtrelenmemiş olduğunu gösterir) oluşturmak üzere D'n’e eklenir. Engelleme distorsiyonunun etkilerini azaltmak için filtre uygulanır ve yeniden oluşturulmuş referans blok bir dizi blok F'n’den (geri çatılmış çerçeve) üretilmiş olur [7].

2013’te ilan edilen HEVC standardı, yeniden yapılandırılmış videonun aynı kalitesiyle H.264/AVC’e kıyasla %35 ila 50 daha iyi sıkıştırma verimi elde etmektedir [8]. Şekil 1.3’te HEVC standardının genel kodlayıcı yapısı gösterilmektedir.

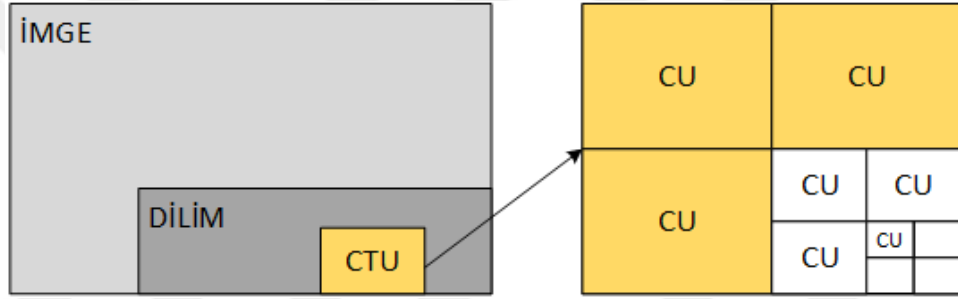


Şekil 1.3. HEVC standardının genel kodlayıcı yapısı [6]

Şekil 1.2’de H.264/AVC video kodlama standardı için bahsedilen kodlama akışı (çerçeve içi ve çerçeveler arası kestirim, dönüşüm, nicemleme, entropi kodlama) HEVC için de benzer şekildedir. Ancak, 4K veya 8K gibi yüksek çözünürlüklü

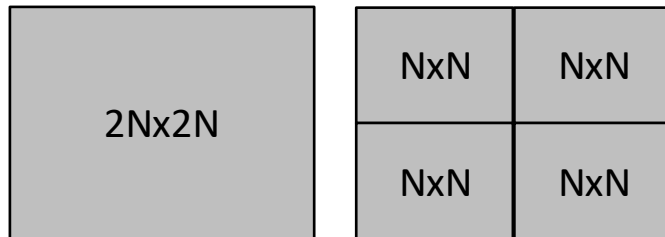
videoların daha verimli kodlanabilmesi için daha büyük blok boyutları ve daha esnek bir kodlama yapısı geliştirilmiştir. H.264/AVC standardında temel kodlama birimleri makro-blok olarak isimlendirilmekte ve boyutları 16×16 piksel olabilmektedir. HEVC standardı, ağaç kodlama birimi (CTU) olarak adlandırılan yeni blok bölümü konseptini sunar ve H.264/AVC’de 16×16 piksel boyutundaki MB’ye kıyasla boyutu 64×64 piksele kadar olabilir [8]. HEVC standardında CTU kodlamanın temel birimidir.

Bir CTU oldukça fazla bilinen dörtlü ağaç (quadtree) yapısına benzer bir şekilde kare şekilli kodlama birimlerine (CU) bölünür. Bir CU boyutu $2N \times 2N$ şeklinde ifade edilirse, N, 4, 8, 16 veya 32 olabilir. HEVC’deki bu hiyerarşik kodlama düzeni Şekil 1.4’te gösterilmiştir.



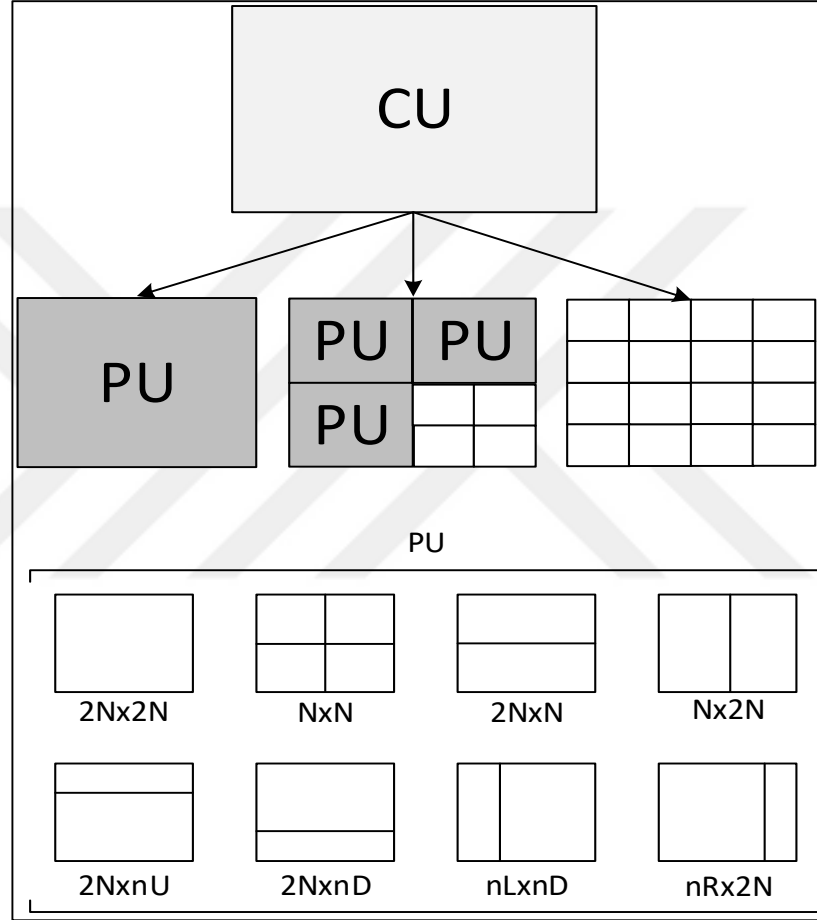
Şekil 1.4. HEVC’de bölümlenme yapıları

Çerçeve içi veya çerçeveler arası tahminlemenin gerçekleştirileceğine CU düzeyinde karar verilir. Çerçeve içi tahminlemede pikseller daha önceden kodlanmış ve geri çatılmış komşu blokların pikselleri kullanılarak tahmin edilir. Çerçeveler arası kodlanmış bir CU için iki adet tahmin birimi (PU) bölümlenme tipi mevcuttur. Şekil 1.5’te gösterildiği gibi N, 4, 8, 16 veya 32 değerleri için CU boyutu $2N \times 2N$ kabul edildiği takdirde, PU boyutu $2N \times 2N$ ve $N \times N$ olabilmektedir. HEVC’de çerçeve içi tahminleme yapılırken PU boyutu 4×4 ile 64×64 aralığında değişmektedir ve 35 adet tahminleme modu kullanılabilir [9].



Şekil 1.5. Çerçeve içi tahminleme için PU boyutları

Çerçevesel tahminlemede her $2N \times 2N$ boyutlu CU, simetrik ($2N \times 2N$, $N \times N$, $2N \times N$ ve $N \times 2N$) ve asimetrik ($2N \times nU$, $2N \times nD$, $nL \times 2N$, ve $nR \times 2N$) olarak Şekil 1.6'da gösterildiği gibi PU'lara ayrılabilir [8]. N , 4, 8, 16 veya 32 olabildiği ve n , $N/2$ 'ye eşit olduğu yerde; R, L, U ve D sırasıyla sağ, sol, yukarı ve aşağı bölümleri temsil eder. HK, PU düzeyinde gerçekleştirilir ve her bir PU kendi hareket vektörüne sahiptir [8].



Şekil 1.6. Çerçevesel tahminleme için kodlama ve tahmin birimi arasındaki ilişki

2. HAREKET KESTİRİMİ

H.264/AVC’de olduğu gibi HEVC’de hareket kestirimi(HK), ardışık video çerçevelerindeki zamansal artıklığı ortadan kaldıran video kodlamadaki en kritik süreçtir. Kodlayıcının hesapsal olarak en karmaşık görevini yerine getiren HK bloğu, kodlama süresinin %90’ından fazlasını harcamaktadır [10]. Hareket kestirimindeki işlem yükünü ve kaynak kullanımını azaltmak için literatürde düşük karmaşıklığa sahip birçok HK yöntemi geliştirilmiştir.

Blok temelli hareket kestirimi yönteminde, her bir çerçeve birbiri ile örtüşmeyen bloklara bölünür ve güncel çerçevedeki her bir blok, referans çerçevelerdeki aynı konum etrafında arama penceresi olarak adlandırılan geniş bir alanda aranır. Orijinal ve aday bloklar arasındaki benzerlik tespit edilir. En çok benzeyen blok konumunu bulmak için arama penceresindeki tüm aday konumların kontrol edildiği yaklaşım tam arama yaklaşımı olarak adlandırılır. B^t güncel bloğu oluşturan bit düzlemini, B^{t-1} referans bloğu oluşturan bit düzlemini temsil ettiği Denklem (2.1)’de, $N \times N$ boyutundaki bir bloğun hareket vektörü hesaplanmaktadır.

$$SAD(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |B^t(i,j) - B^{t-1}(i+m, j+n)|, \quad -s \leq m, n \leq s \quad (2.1)$$

$$HV = \min(SAD(m,n))$$

Denklem(2.1)’de (m,n) aday hareket vektörünü, s arama aralığını, i ve j blok içerisindeki ilgili pikselin konumunu temsil etmektedir. Burada aday vektörü, mutlak farklar toplamı (SAD) ölçütü ile hesaplanan en yüksek benzerliğe sahip bloğun hareket vektörüdür.

Tam arama temelli HK yöntemlerinde bütün aday konumlarının kontrol edilmesinden kaynaklanan hesapsal yükü ve donanım karmaşıklığını azaltmak için farklı yaklaşımlar geliştirilmiştir. Bunlardan biri arama penceresindeki tüm aday konumların yalnızca belirli bir alt kümesini kontrol ederek aday konum sayısının azaltılmasıdır.

Üç adımlı arama (TSS) [11], elmas arama [12] ve altıgen arama [13] bu yaklaşıma örnektir. [14]'de önerilen çalışmada ise her bir blok için önceden belirlenmiş bir arama aralığına bağlı olarak sınırlı aday konum sayısı kontrol edilmektedir. Bu çalışma uyarlanabilir arama aralığını belirlemeye dayalı yaklaşımlara örnek verilebilir.

Bir diğer yaklaşım ise uyumlama kriterinin hesaplanmasında kullanılan piksel sayısını azaltılmasıdır. Alt örnekleme örüntüsü yararlanarak piksel sayısının azaltıldığı bu gruptaki çalışmalara quarter (çeyrek) [15] ve 8-queen (sekiz-kraliçe) [10] ve yeniden yapılandırılabilir sınır [16] örnek verilebilir.

Tam arama yaklaşımının hesapsal yükünü azaltmak amacıyla arama yapılacak aday konum sayısını azaltmaya dayalı diğer bir yaklaşım grubunda ise belirli veya geriye kalan tüm aday konumlar için uyumlama kriterinin hesaplanmasını atlanmaktadır. Örneğin, [17]'de sunulan yaklaşım gibi ardışık eliminasyon algoritmasına dayanan yöntemler, daha düşük karmaşıklıkta uyumlama kriterinin alt sınırını hesaplar ve böylece bu aday konum için uyumlama kriterini hesaplamadan önce imkansız adayları atlar. Benzer bir şekilde, [18]'de önerilen çalışmada olduğu gibi erken sonlandırma yöntemleri, kısmi uyumlama sonucunu geçerli olan en düşük uyumlama hatasıyla karşılaştırarak belirli aday konumlar için uyumlama kriterinin hesaplanmasını atlamaktadır. Böylece, ilgili güncel blok için tam uyumlama kriterini hesaplamadan imkânsız aday konumlarını ortadan kaldırmak mümkün olabilmektedir.

2.1. Düşük Bit Gösterimi Temelli Hareket Kestirimi Yöntemleri

Düşük karmaşıklığa sahip olmak için geliştirilen diğer bir yaklaşım ise imgelerin 8 bit yerine daha düşük bit derinliğinde ifade edilmesidir. Düşük bit derinliği gösterimli yaklaşımların uyumlama sürecinde, geleneksel SAD yerine donanım maliyeti ve güç tüketimi açısından avantaj sağlayan özel veya (EXOR) işleminin kullanılması uyumlama ölçütünün hesaplanmasını hızlandırır ve işlem yükünü azaltır.

Literatürde imgeleri düşük bit derinliği ile ifade edebilmek için farklı HK algoritmaları kullanılmıştır. İkileştirme ve uyumlama aşamasında kullandıkları yöntemlere göre bilinen birkaç HK algoritması gruplandırılarak ana hatlarıyla aşağıda anlatılmıştır.

2.1.1. Bir bit dönüşümü temelli HK yöntemi

Uyumlama aşamasında kolaylıkla kullanılmak üzere, giriş imgeleri süzgeçlenerek ikili imgelere dönüştürülür. Süzgeçleme temelli ikilileştirme yaklaşımının benimsendiği [19]'da sunulan çalışmada pikselleri 1-bit derinliğinde ifade etmek için 17×17 çoklu bant geçiren süzgeç çoklu bant geçiren süzgeç kullanılmıştır. 1-bit dönüşümü (1BT) olarak adlandırılan bu yöntemde giriş imgelerinden ikili imgeleri elde ederken Denklem (2.2)'de ifade edilen eşitlik kullanılmıştır.

$$K_{1BT}(i,j) = \begin{cases} 1/25, & \text{if } i,j \in [1,4,8,12,16] \\ 0, & \text{diğer} \end{cases} \quad (2.2)$$

İkilileştirme sürecinde süzgeçlenmiş imge çerçevesi ile orijinal imge çerçevesi pikselleri arasında Denklem (2.3)'te ifade edildiği gibi bir karşılaştırma yapılmış ve bu karşılaştırma sonucuna göre ikili imge çerçevesi $B(i,j)$ elde edilmiştir.

$$B(i,j) = \begin{cases} 1, & \text{if } I(i,j) \geq I_F(i,j) \\ 0, & \text{diğer} \end{cases} \quad (2.3)$$

İkili hale dönüştürülmüş imge çerçeveleri arasında blok uyumlaması yapılırken uyumlama ölçütünün kolaylıkla gerçekleştirilebilir olması donanım açısından avantaj sağlamaktadır. [19]'da sunulan 1BT yönteminde kullanılan uyumlama ölçütü Denklem (2.4)'te gösterilmiştir. Uyumsuz nokta sayısı (Number of Non-Matching Points (NNMP)) ölçütünde, referans bloğun piksel değeri ile o anki aday bloğun piksel değeri arasında EXOR işlemi kullanarak eşleşmeyen nokta sayısı hesaplanır.

$$NNMP_{1BT}(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B^t(i,j) \oplus B^{t-1}(i+m,j+n) \quad (2.4)$$

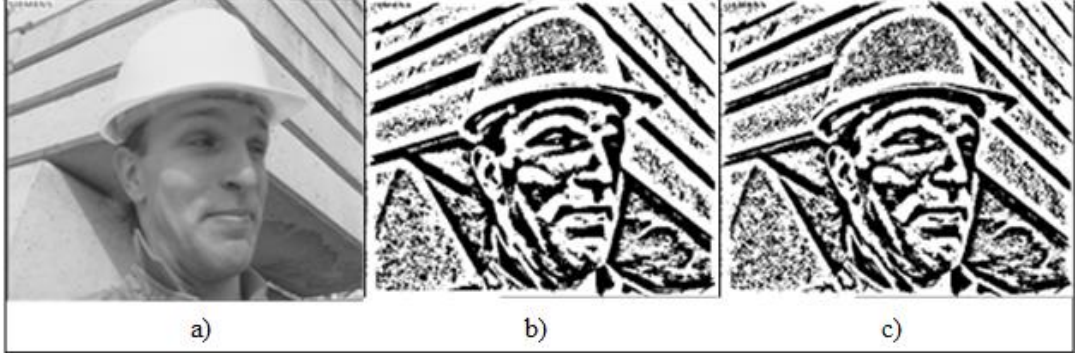
Denklem (2.4)'de i ve j blok içinde ilgili pikselin konumunu, m ve n aday hareket vektörünü, N blok boyutunu, B^t güncel bloğu oluşturan bit düzlemini, B^{t-1} arama penceresini oluşturan bit düzlemini, \oplus EXOR işlemi temsil etmektedir. Bütün aday (m, n) noktaları için NNMP ölçütü hesaplanır ve en küçük NNMP değerine sahip aday bloğun pozisyon bilgisi hareket vektörünü ifade etmektedir.

2.1.2. Çarpmasız bir bit dönüşümü temelli HK yöntemi

[19]'da önerilen yöntemde kullanılan çekirdek 25 tane sıfırdan farklı elemana sahiptir. Bu da imgeleri süzgeçlerken kayan noktalı çarpma işlemlerini gerektirir ve işlemsel karmaşıklığı önemli derecede artırır. [20]'de sunulan yöntemde 19×19 boyutundaki elmas şekilli çekirdek kullanılarak bu problemi ortadan kaldırmak istenmiştir. Bu çekirdek 16 tane sıfırdan farklı elemana sahiptir. Çekirdeğindeki sıfırdan farklı eleman sayısının ikinin kuvveti şeklinde ayarlandığı bu yöntemde süzgeçleme aşaması, kayan noktalı çarpma işlemleri kullanılmadan yapılabilmektedir. Çarpmasız bir bit dönüşümü olarak adlandırılan MF-1BT yönteminin çekirdek yapısı Denklem (2.5)'te gösterilmektedir.

$$K_{MF-1BT}(i,j) = \frac{1}{16} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.5)$$

İkili hale dönüştürülmüş giriş çerçevelerinin uyumlanması aşamasında [19]'da sunulan 1BT yönteminde olduğu gibi $NNMP_{1BT}$ ölçütü kullanılmaktadır.



Şekil 2.1. Foreman video dizisi örnek imgeleri a) Orijinal 8-bit derinlikli imge b)1BT yaklaşımı c) MF-1BT yaklaşımı [21]

Şekil 2.1’de Foreman video dizisinin beşinci çerçevesi için sırasıyla orijinal 8-bit derinlikli, 1BT, MF-1BT yöntemi ile elde edilen 1-bit derinlikli imgeler gösterilmiştir.

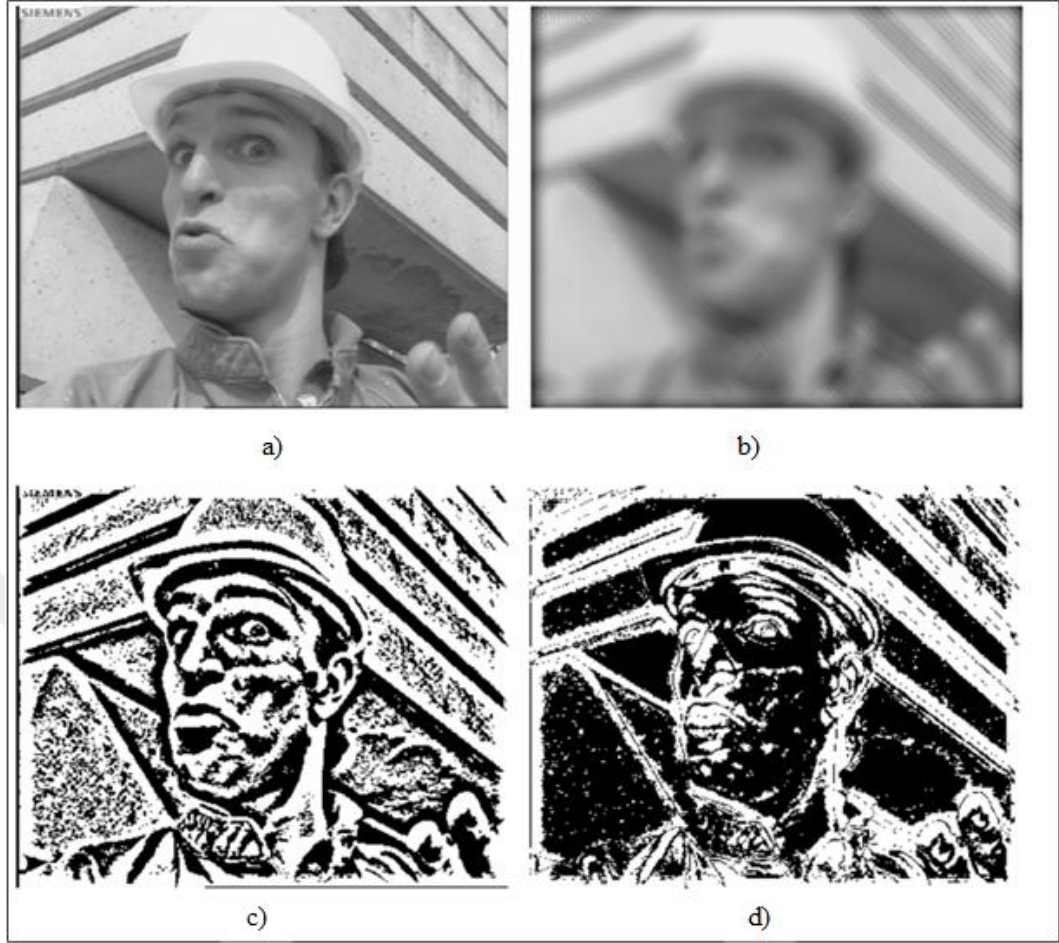
2.1.3. Kısıtlanmış bir bit dönüşümü temelli HK yöntemi

[22]’de önerilen kısıtlanmış bir-bit dönüşümü (C-1BT) temelli yöntemde iki bit düzlemi kullanılmaktadır. C-1BT yönteminde ilk bit düzlemi oluşturulurken MF-1BT yönteminde kullanılan çekirdek yapısı kullanılmıştır. İkinci bit düzlemi ise kısıt maskesi (CM) kullanılarak elde edilir. Denklem (2.6)’da ifade edilen kısıt maskesi orijinal ve süzgeçlenmiş çerçeve arasındaki mutlak farkı sabit bir eşik değeri D ile karşılaştırarak uyumlamaya katılacak pikselleri tespit etmektedir.

$$CM(i,j)=\begin{cases} 1, & \text{eğer } |I(i,j)-I(i,j)*K_{MF-1BT}|\geq D \\ 0, & \text{diğer} \end{cases} \quad (2.6)$$

Hesaplanan sıfır değerleri güvenilir olmayan piksellere karşılık gelir ve bu pikseller uyumlama aşamasına dahil edilmez. C-1BT yönteminde kullanılan kısıtlanmış uyumsuz nokta sayısı (CNNMP) uyumlama ölçütü Denklem (2.7)’de gösterilmektedir.

$$CNNMP(m,n)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\{CM^t(i,j)\|CM^{t-1}(i+m,j+n)\}\odot\{B^t(i,j)\oplus B^{t-1}(i+m,j+n)\} \quad (2.7)$$



Şekil 2.2. Foreman video dizisi örnek imgeleri a) 8-bit derinlik b) Filtrelenen imge c) MF-1BT yaklaşımıyla elde edilen 1-bit imge d) CM [22]

Şekil 2.2’de Foreman video dizisi örnek imge, Denklem (2.6)’daki çekirdek yapısı kullanılarak ile filtrelenen imge, MF-1BT kerneli ile elde edilen 1-bit derinlikli imge ve kısıt maskesi gösterilmektedir. Daha önce değinildiği üzere CM piksellerin eşige kabul edilebilir bir uzaklıkta olup olmadığına bakılarak oluşturulmaktadır ve ilgili piksellerin 1-bit uyumlamasında dikkate alınıp alınmayacağına karar verilmesi için kullanılmaktadır.

2.1.4. Yerel ikili örüntü temelli HK yöntemleri

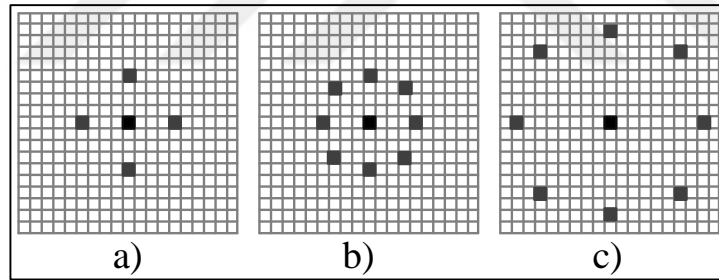
İlk olarak [23]’te önerilen yerel ikili örüntü (LBP) yöntemi yerel komşuluk değerlerine dayalı bir özellik çıkarma yöntemidir. Bu yöntem literatürde yüz tanıma [24], yaş tahmini [25], kolonoskopi görüntü sınıflaması [26] gibi birçok imge işleme uygulamasında kullanılmıştır. LBP yöntemi [27]’de sunulan çalışmada video dengeleme için ikilileştirme yaklaşımı olarak kullanılmıştır. [28]’de sunulan çalışmada

uyumlama aşamasında bir bit düzlemi kullanabilmek için 8 bitlik giriş imgeleri LBP-1BT olarak adlandırılan özgün ikilileştirme yöntemi ile 1 bit düzlemine dönüştürülmektedir. Bu çalışmada ikili imge elde edilirken kullanılan Denklem (2.8)'de gösterilmektedir. Denklemde $B(i,j)$ elde edilen ikili imgeyi, P komşu piksel sayısını, (i, j) ilgili merkez piksel konumunu belirtir. LBP-1BT yönteminin uyumlama ölçütü [19]'da kullanılan ölçüt ile aynıdır.

$$B(i,j) = \begin{cases} 1, & \text{if } \sum_{p=0}^{P-1} \text{sign}(I(p) - I(i,j)) \geq P/2 \\ 0, & \text{diğer} \end{cases} \quad (2.8)$$

$$\text{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{diğer} \end{cases}$$

Önerilen LBP temelli ikilileştirme yaklaşımında, giriş imgesindeki her bir piksel R yarıçapındaki bir çember üzerindeki eşit aralıklı P tane komşu piksel ile karşılaştırılır. Şekil 2.3'te LBP konfigürasyonları için örnekler gösterilmiştir.



Şekil 2.3.Örnek LBP konfigürasyonları LBP (P, R)
a) LBP(4, 4) b) LBP(8, 4) c) LBP(8, 8) [28]

LBP yönteminin hareket kestiriminde ikilileştirme yaklaşımı olarak önerildiği diğer bir çalışma ise [29]'da sunulmuştur. Daha iyi hareket kestirimi performansı ve daha düşük karmaşıklık sağlamak amacıyla geliştirilen LBP-2BT yönteminde uyumlama aşamasında iki bit düzlemi kullanılmıştır. Denklem (2.9)'da yöntemde kullanılan iki bit-düzleminin nasıl oluşturulduğu gösterilmektedir. Denklem (2.9)'da $B_1(i, j)$ ve $B_2(i, j)$ imgedeki birinci ve ikinci bit düzlemini, T sabit bir eşik değerini, $I(i, j)$ ve P ise sırasıyla (i, j) konumundaki ilgili piksel değeri ve pikselin komşularını ifade etmektedir.

$$\begin{aligned}
B_1(i,j) &= \begin{cases} 1, & \text{if } \sum_{p=0}^{P-1} \text{sign}(I(p) - I(i,j) - T) \geq P - 2 \\ 0, & \text{diğer} \end{cases} \\
B_2(i,j) &= \begin{cases} 0, & \text{if } \sum_{p=0}^{P-1} \text{sign}(I(p) - I(i,j) - T) = \{0 \text{ or } P\} \\ 1, & \text{diğer} \end{cases} \\
\text{sgn}(x) &= \begin{cases} 1, & x \geq 0 \\ 0, & \text{diğer} \end{cases}
\end{aligned} \tag{2.9}$$

Önerilen yaklaşımda LBP kullanarak giriş imgesi 2-bit uzayında temsil edildikten sonra, bir bloğun hareket vektörlerinin bulunması için Denklem (2.10)'daki modifiye uyumsuz nokta sayısı (Modified-Non-Matching Points (MNNMP)) isimli bir metrik oluşturulmuştur.

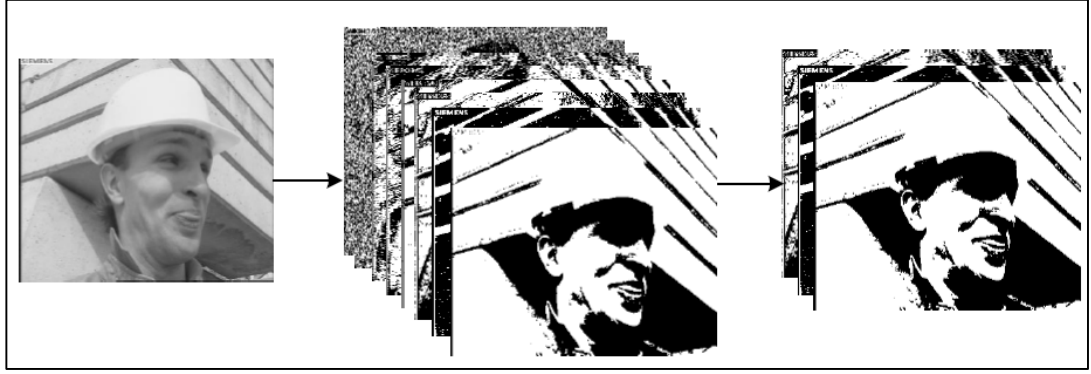
$$\text{MNNMP}(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{B_1^t(i,j) \oplus B_1^t(i+m, j+n)\} + \{B_2^t(i,j) \oplus B_2^t(i+m, j+n)\} \tag{2.10}$$

Denklem (2.10)'da $-s \leq m, n \leq s-1$, (m,n) aday hareket vektörlerini göstermektedir. N ve s blok boyutu ve arama aralığıdır. $B_1^t(i,j)$ ve $B_2^t(i,j)$ sırasıyla birinci ve ikinci bit düzlemlerini temsil etmektedir. Her bit düzlemi referans ve o anki çerçeveye için kendi içinde EXOR işlemi uygulanıp sonuçları toplanarak MNNMP değeri oluşturulmakta ve minimum MNNMP değerine göre hareket vektörleri elde edilmektedir.

2.1.5. Seçimli Gray kodlama temelli HK yöntemi

[30]'da sunulan çalışmada seçimli Gray kodlama temelli (SGC) 1-bit derinliğinde HK yöntemi önerilmiştir. Bu algorithmanda, ilk olarak, en değerliksiz bit düzlemleri düşürülerek 8 bit / piksel video çerçevesinden 3 en değerlikli bit düzlemi çıkarılır ve daha sonra bu 3 en değerlikli bit düzleminde ikili görüntü elde edilir. Şekil 2.4'te, 3 en değerlikli bit düzlemi elde etme süreci görülmektedir.

Gray kodlanmış piksel değerlerinden oluşan imgenin en değerlikli 3 bit düzleminde (g_5, g_6, g_7) sırayla seçme yapılmaktadır. Seçme işleminde kullanılan örüntü Şekil 2.5'te gösterildiği gibidir. Hem referans imgedeki bloklar için hem de aday bloklar için seçme işlemi yapılarak bir bit düzlemi elde edilir. Yöntemde kullanılan uyumlama ölçütü 1BT ve MF-1BT yöntemlerinde kullanılan standart NNMP ölçütüdür.



Şekil 2.4. 3 bitlik düzlemlerin elde edilme süreci

g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6
g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7
g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g7	g5

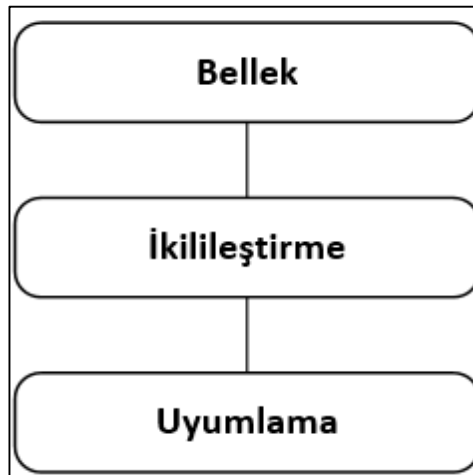
Şekil 2.5. SG temelli yöntemde bit düzlemlerinin seçilmesi

3. DÜŞÜK KARMAŞIKLIĞA SAHİP HK YAKLAŞIMLARI İÇİN ÖNERİLEN DONANIM MİMARİLERİ

3.1. 1BT, MF-1BT ve C-1BT HK Yaklaşımları İçin Önerilen Donanım Mimarileri

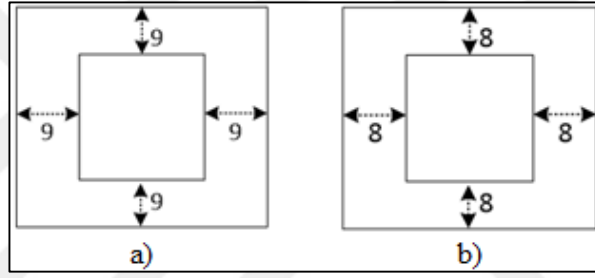
[31] ve [32]'de [20]'de önerilen MF-1BT yöntemi için, [34] ve [35]'de [22]'de önerilen C-1BT yöntemi için donanım mimarileri sunulmuştur. Literatürde düşük karmaşıklığa sahip HK yöntemleri için birçok donanım mimarisi önerilmiş olsa da bu mimarilerde uyumlama aşaması öncesinde ikili imgelerin hazır olduğu kabul edilmiş ve ikilileştirme aşaması göz ardı edilerek sadece uyumlama aşamaları gerçekleştirilmiştir. Video kodlayıcı mimarisinin tamamı düşünüldüğünde, en önemli aşama olan hareket kestiriminde ikilileştirme maliyetinin hesaba katılmaması uygun bir yaklaşım değildir.

Bu tez çalışması kapsamında, literatürdeki düşük karmaşıklığa sahip HK yöntemlerinden 1BT, MF-1BT ve C-1BT için süzgeçleme yapılarına dikkat çekilerek literatürde ilk defa ikilileştirme aşamasının da dahil edildiği HK donanım mimarileri sunulmuştur. Şekil 3.1'de HK donanım mimarilerinin temel blokları gösterilmiştir.



Şekil 3.1. İkilileştirme ve uyumlama aşaması birlikte ele alındığında temel HK donanım mimarisi

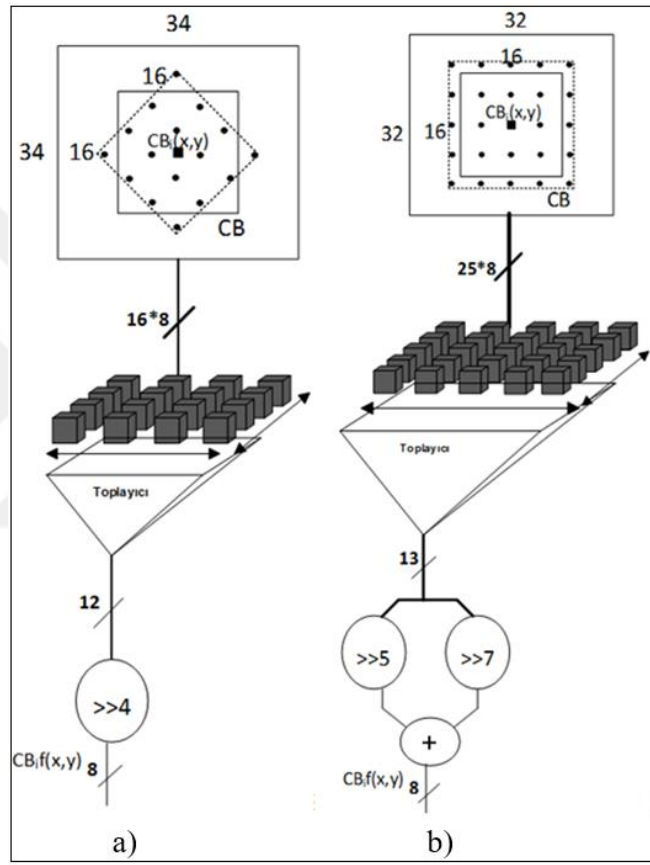
Süzgeçleme temelli ikilileştirme yaklaşımlarını kullanan 1BT, MF-1BT ve C-1BT yöntemlerinde giriş imgeleri süzgeçlenerek ikili imgelere dönüştürülür. Girişteki orijinal imge çerçevesi süzgeçlendikten sonra süzgeçlenmiş imge çerçevesi ve orijinal imge çerçevesi pikselleri arasında karşılaştırma yapılarak ikili imge çerçevesi elde edilir. Süzgeçleme aşamalarında önceki bölümlerde de bahsedildiği gibi farklı çekirdek yapıları kullanılmıştır. Bu yöntemlerde uyumlamada kullanılacak ikili imgeleri oluştururken 2-boyutlu çekirdeklerden faydalanılır. 1BT için 17×17 , MF-1BT ve C-1BT için 19×19 boyutunda çekirdekler kullanılır. Çekirdeklerin 2-boyutlu yapısı nedeniyle hafızada tutulması gereken blok boyutları süzgeçlenecek blok boyutlarından büyüktür. Diğer bir ifadeyle, $((\text{çekirdek uzunluğu}-1)/2)$ değeri kadar ilave piksel ikilileştirme aşamasında kullanılır. Şekil 3.2’de bu durum gösterilmiştir.



Şekil 3.2. Süzgeçleme aşamasında gerekli olan imge çerçevesi bölgeleri a) MF-1BT ve C-1BT yöntemleri için b) 1BT yöntemi için [36]

Şekil 3.2’de 1BT, MF-1BT ve C-1BT yöntemlerinde güncel blokta bir pikselin süzgeçlenmesi örnek alınarak süzgeç yapıları gösterilmiştir. MF-1BT ve C-1BT yöntemlerinin ikilileştirme aşamasında aynı çekirdek kullanıldığından süzgeç yapıları da aynıdır. Buna göre MF-1BT ve C-1BT yöntemlerinde 1 piksel süzgeçlenirken 16 tane toplama işlemi ve 1 tane kaydırma işlemi gerekmektedir. Şekil 3.3(a)’da gösterildiği gibi tasarlanan süzgeç yapısında 16 tane toplayıcı ve toplamın $1/16$ ile çarpma işlemini yerine getiren 1 tane 4-bitlik kaydırıcı kullanılmıştır. Şekil 3.3(b)’de 1BT yöntemi için tasarlanan süzgeç yapısında 25 tane toplayıcı gösterilmiştir. Bu yöntemde kullanılan çekirdeğin 25 tane sıfır olmayan elemanı olduğundan normalizasyon için 25 ’e bölme işlemi gerekmektedir. 25 değeri 2 ’nin kuvveti olmadığı için basit kaydırıcılar kullanılarak bu işlem gerçekleştirilememektedir. [36]’da sunulan çalışmada klasik 1BT yaklaşımında kuvvet indirgeme yöntemi kullanılarak kayan noktalı çarpma işlemlerine etkin bir çözüm getirilmiş ve tamsayı 1-bit dönüşümü (I-

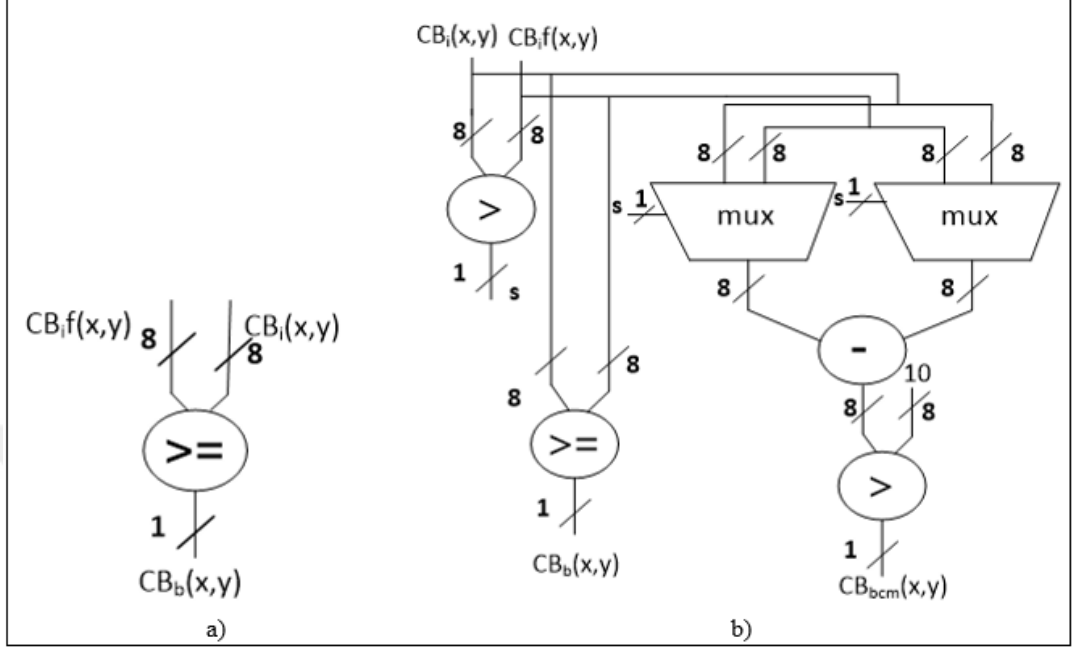
1BT) olarak adlandırılan özgün bir ikilileştirme yaklaşımı önerilmiştir. Buna göre, $1/25$ ile çarpmak yerine $(1/32 + 1/128)$ değeri ile yani $1/25.6$ ile çarpma işlemini gerçekleştiren 1 tane 7-bit ve 1 tane 5-bit kaydırma işlemi yapıp sonuçları 1 toplayıcı ile toplanmıştır. Böylece kayan noktalı çarpma işlemleri tamsayı forma getirilip işlem karmaşası ortadan kaldırılmıştır. Şekil 3.3'te gösterilen süzgeç mimarilerinde kesikli çizgiler ile ifade edilen geometrik şekiller merkezde süzgeçlenecek orijinal pikseli ve etrafında süzgeç çekirdeğini göstermektedir.



Şekil 3.3. Süzgeç donanım mimarileri a) MF-1BT ve C-1BT yöntemleri için, b) 1BT yöntemi için [36]

Şekil 3.3'teki süzgeç yapılarının çıkışında süzgeçlenmiş piksel değerleri elde edildikten sonra merkezdeki orijinal piksel değerleri ile karşılaştırma işlemi yapılmaktadır. Karşılaştırma sonucunda orijinal piksel değerinin ikili karşılığı elde edilmektedir ve bu aşama kullanılan hareket kestirimi yöntemine göre değişiklik göstermektedir. Şekil 3.4'te ikilileştirmenin son aşamasında kullanılan yapılar gösterilmiştir. Şekil 3.4(a)'da görüldüğü gibi MF-1BT ve 1BT yöntemleri için 1 tane

karşılaştırmacı kullanılarak bu aşama gerçekleştirilirken, C-1BT için daha karmaşık bir yapı kullanılarak ikili değere ulaşıldığı Şekil 3.4(b)'de gösterilmektedir.



Şekil 3.4. İkileştirme sürecinin son aşamasında kullanılan yapılar a) MF-1BT ve 1BT yöntemleri için, b) C-1BT yöntemi için [36]

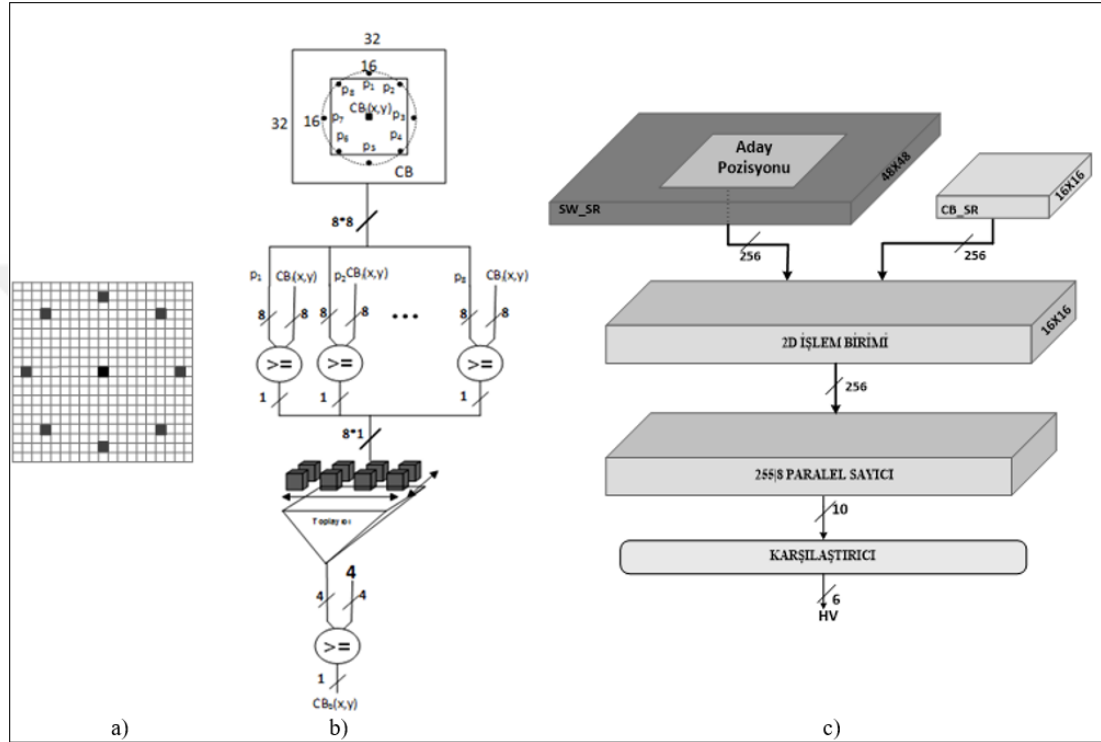
3.2. LBP-1BT ve LBP-2BT HK Yaklaşımları İçin Önerilen Donanım Mimarileri

3.2.1. 1-bit derinliğinde LBP donanımı (LBP-1BT)

Önceki bölümlerde bahsedildiği gibi yerel ikili örüntü yönteminin farklı kullanım alanları bulunmaktadır. Bu tez çalışmasında, [38]'de sunulan düşük karmaşıklığa sahip bir hareket kestirimi yöntemi olan LBP-1BT için hem ikileştirme hem de uyumlama aşamasını içeren bir donanım mimarisi önerilmiştir.

Temelde merkez ve komşu piksel değerleri arasında yapılan sıralı ikili karşılaştırmalara dayanan klasik LBP yönteminin [28]'de sunulan çalışmadaki kullanım amacı 8-bit derinliğindeki giriş imgelerini 1-bit derinliğinde ifade ederek ikileştirme sürecinin gerçekleştirilmesidir. Önerilen LBP temelli ikileştirme yaklaşımında, giriş imgesindeki her bir piksel R yarıçapındaki bir çember üzerindeki eşit aralıklı P tane komşu piksel ile karşılaştırılır. R yarıçapı kadar uzaklıktaki komşu pikseller ikileştirmeye katıldığından süzgeçlenecek blok boyutuna ilave pikseller eklenir. Bu tez çalışmasında Şekil 3.5(a)'daki konfigürasyon için R ve P değerleri 8

seçilerek 8 yarıçapındaki bir çember üzerindeki eşit aralıklı 8 tane komşu piksel ile merkez piksel karşılaştırılır. Karşılaştırmacı çıkışları toplanır ve $P/2$ değeri ile karşılaştırılır. Bu tez çalışmasında $P/2$ değeri 4 olmaktadır. Karşılaştırma sonucunda süzgeçlenen pikselin ikili değeri elde edilmektedir. Şekil 3.5(b)'de 16×16 boyutundaki güncel bloğun merkezindeki piksel için süzgeçleme yapıları gösterilmiştir.



Şekil 3.5. LBP yöntemi için donanım mimarisi a) Donanımı gerçekleştirilen LBP (8,8) konfigürasyonu, b) İkileştirme donanım mimarisi, c) Uyumlama donanım mimarisi [28]

Şekil 3.5(c)'de ikili imgeler elde edildikten sonraki uyumlama aşamasının donanım mimarisi gösterilmiştir. I-1BT, MF-1BT ve LBP-1BT yöntemlerinin hepsinde giriş imgeleri 1-bit derinliğinde ifade edilecek şekilde süzgeçlenip uyumlama aşamasına gönderilir. Bu sebeple bu aşamada aynı donanım mimarisi kullanılabilir. Arama penceresi ve güncel blok için kullanılan kaydedici dizileri [34]'te önerildiği gibi 3 ve 4 yönlü kaydırma kabiliyeti olan yaz-bozlardan oluşmaktadır. 2D işlem birimi dizisi, paralel sayıcı ve karşılaştırmacı blokları [30]'da sunulan mimariye benzerdir.

3.2.2. 2-bit derinliğinde LBP donanımı (LBP-2BT)

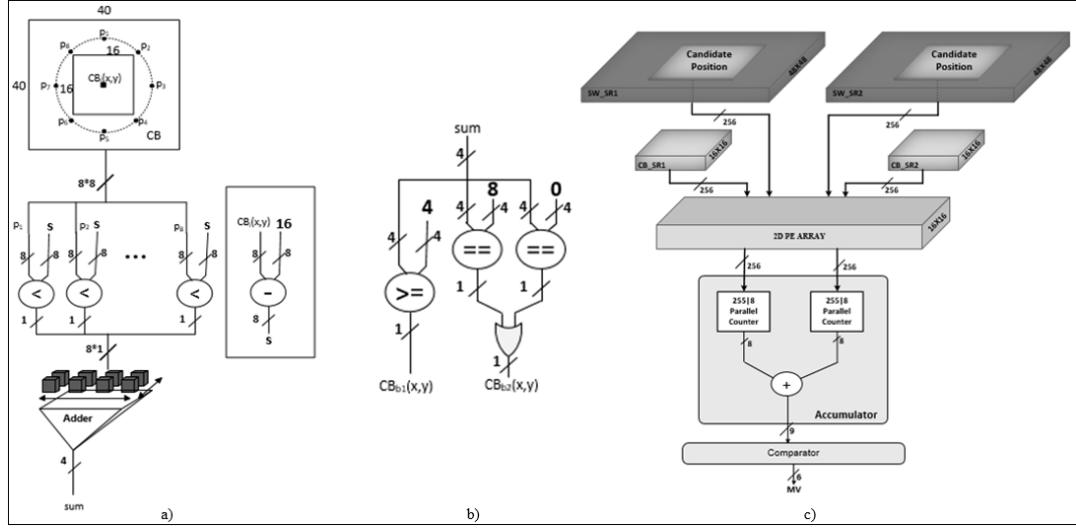
LBP-2BT yönteminde, 2-bit derinliğinde oluşturulan süzgeçlenmiş ikili imgeler uyumlama aşamasında kullanılmıştır. Bu yönteminin ikilileştirme yaklaşımı 1-bit derinliğinde çalışılan LBP-1BT yöntemindeki yaklaşıma çok benzerdir. [29]'da önerilen LBP-2BT yönteminin deneysel sonuçları incelendiğinde P değeri 8, R değeri 12 seçilerek LBP (8,12) konfigürasyonu için T değeri 16 seçildiğinde en iyi HK performansı elde edilmektedir.

Tez çalışmasında P değeri 8, R değeri 12 seçilerek LBP (8,12) konfigürasyonu için süzgeç yapısı tasarlanmıştır. Buna göre 12 yarıçapındaki çember üzerindeki eşit aralıklı 8 komşu piksel karşılaştırmaya dahil edilmiştir. R değeri 12 seçilerek 12 yarıçapındaki çemberin üzerindeki komşu piksellerin ikilileştirmeye katıldığı yöntemde çekirdek boyutu 25×25 olmaktadır.

1-bit derinliğindeki LBP-1BT yönteminden farklı olarak merkez piksel değerinden T değeri çıkartılır ve komşu pikseller bu çıkarma işlemi sonucu ile karşılaştırılır (Not: T değeri daha önce açıklandığı üzere 16 olarak seçilmiştir). Sonrasında karşılaştırmacı çıkışları toplanır ve toplam değeri elde edilir.

Şekil 3.6(a)'da anlatılan süzgeçleme aşamalarını 16×16 boyutundaki güncel bloğun merkezindeki 1 piksel için gerçekleştiren 1 adet çıkarma, 8 adet karşılaştırma ve 8 adet toplama bloğu gösterilmektedir.

8-bitlik piksellerden oluşan güncel bloğun süzgeçlenmiş 2 ayrı ikili karşılığını oluşturmak için toplam sonuç P, P/2 ve 0 değerleri ile karşılaştırılır. P/2 ile yapılan karşılaştırma sonucu birinci bit düzlemindeki ilgili değer oluşturulurken P ve 0 ile yapılan karşılaştırma sonuçları OR işleminden geçirilerek elde edilen 1-bitlik sonuç ikinci bit düzlemindeki ilgili değeri oluşturur. Şekil 3.6(b)'de toplama işleminden sonraki aşamalar gösterilmektedir.



Şekil 3.6. LBP-2BT yöntemi için donanım mimarisi a) Süzgeç donanım mimarisi b) İkileştirme son aşamasında kullanılan yapı, c) Uyumlama donanım mimarisi [37]

3.3. 1DF-2BT HK Yaklaşımı İçin Önerilen Donanım Mimarisi

Önceki bölümlerde anlatılan ve literatürde bulunan düşük bit gösterimine sahip filtreleme mantığına dayalı HK yöntemlerinin ikileştirme maliyetleri incelendiğinde bu yaklaşımların iddia edildiği gibi donanıma uygun olmadığı ortaya çıkarılmıştır. Literatürde bu sebeple 2-boyutlu süzgeçleme yerine 1-boyutlu dikey, yatay ve diyagonal süzgeçleme yapan ve orijinal çerçeveyi 1-bit ile temsil eden bir çalışma [38]'de gerçekleştirilmiştir. Bu çalışmada en iyi HK sonuçlarının diyagonal süzgeçleme yapılması durumunda elde edildiği üzerinde durulmaktadır. Bu tez çalışmasında bu yöntem LB-1BT diye isimlendirilmiştir. Tez kapsamında yapılan donanım çalışmalarında ise diyagonal filtreleme yapmanın donanımsal maliyetinin yüksek olduğu tespit edilmiştir. [21]'de ikileştirme işlemi sırasında daha düşük işlem yüküne sahip 1-boyutlu çekirdek kullanıp yatay süzgeçleme yapan ve toplamda 2-bit uzayı kullanan bir yöntem önerilmiş ve bu bir boyutlu süzgeçleme temelli iki-bit derinliğinde hareket kestirimi yöntemi 1DF-2BT olarak isimlendirilmiştir.

Önerilen yaklaşım 8-bitlik bir imgeyi iki-bit düzleminde temsil etmektedir. Birinci bit düzlemi pikselin en anlamlı bitine (most significant bit (MSB)) göre belirlenmektedir. Denklem (3.1)'de görüldüğü gibi piksel değeri 128'den büyük ise ilgili piksel 1 olarak aksi durumda 0 olarak belirlenmektedir. İkinci bit uzayı ise komşuluk ilişkilerini de

dikkate alarak 1-boyutlu bir çekirdek ile süzgeçleme yaparak Denklem (3.1)'deki gibi 1 veya 0'a karar vermektedir. Bu denklemdeki I_F bir-boyutlu çekirdekle giriş imgesinin süzgeçlenmiş halidir.

$$\begin{aligned} B_{1DF-2BT_1}(i,j) &= \begin{cases} 1, & I(i,j) \geq 128 \\ 0, & \text{diğer} \end{cases} \\ B_{1DF-2BT_2}(i,j) &= \begin{cases} 1, & \text{if } I(i,j) \geq I_F(i,j) \\ 0, & \text{diğer} \end{cases} \end{aligned} \quad (3.1)$$

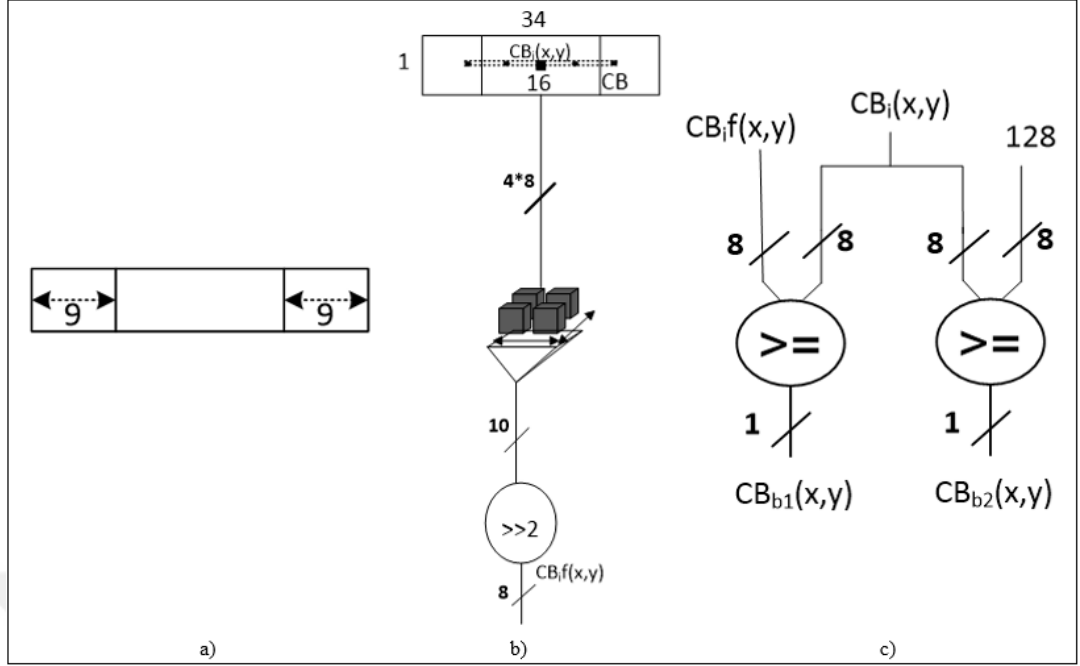
Burada en önemli avantaj kullanılan çekirdeğin bir boyutlu olmasıdır. Denklem (3.2)'de kullanılan 1-boyutlu çekirdek verilmektedir. Sadece 4 tane sıfırdan farklı değer bulunduran bu süzgeç ile toplama ve kaydırma işlemleri yaparak imgenin süzgeçlenmesi gerçekleştirilebilmektedir.

$$K_{1DF-2BT}(i,j) = \begin{cases} 1/4, & \text{if } i,j \in [1,6,14,19] \\ 0, & \text{diğer} \end{cases} \quad (3.2)$$

İkili imgeler elde edildikten sonra Denklem (3.3)'te denklemde verildiği gibi SMNNMP (Scaled modified non-matching points) isimli bir ölçüt oluşturulmuş ve hareket vektörü kararı buna göre verilmiştir. Burada 1.bit uzayı 2 ile ağırlıklandırılarak daha iyi HK başarımı elde edilmesi hedeflenmektedir.

$$\begin{aligned} SMNNMP(m,n) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{B_{1DF-2BT_1}^t(i,j) \oplus B_{1DF-2BT_1}^{t-1}(i+m,j+n)\} \times 2 + \\ &\{B_{1DF-2BT_2}^t(i,j) \oplus B_{1DF-2BT_2}^{t-1}(i+m,j+n)\} \end{aligned} \quad (3.3)$$

Bu tez kapsamında [21]'de önerilen 1DF-2BT temelli yöntemle ilişkin donanım mimarisi geliştirilmiştir. Süzgeçleme temelli HK yöntemlerinde süzgeçleme aşamasında kullanılan çekirdeklerin 2-boyutlu yapısı hafızada tutulması gereken blok boyutlarının süzgeçlenecek blok boyutlarından fazla olmasına sebep olmaktadır. Şekil 3.2'de de gösterildiği gibi, 2-boyutlu çekirdek kullanıldığında 4 yönde de ilave pikseller ikilileştirmeye katılmaktadır. Bunun yerine 1-boyutlu çekirdek kullanıldığında sadece 2 yönde artış olmaktadır. İşgal edilen yonga üzeri bellek alanından önemli derecede kazanç sağlanmaktadır.

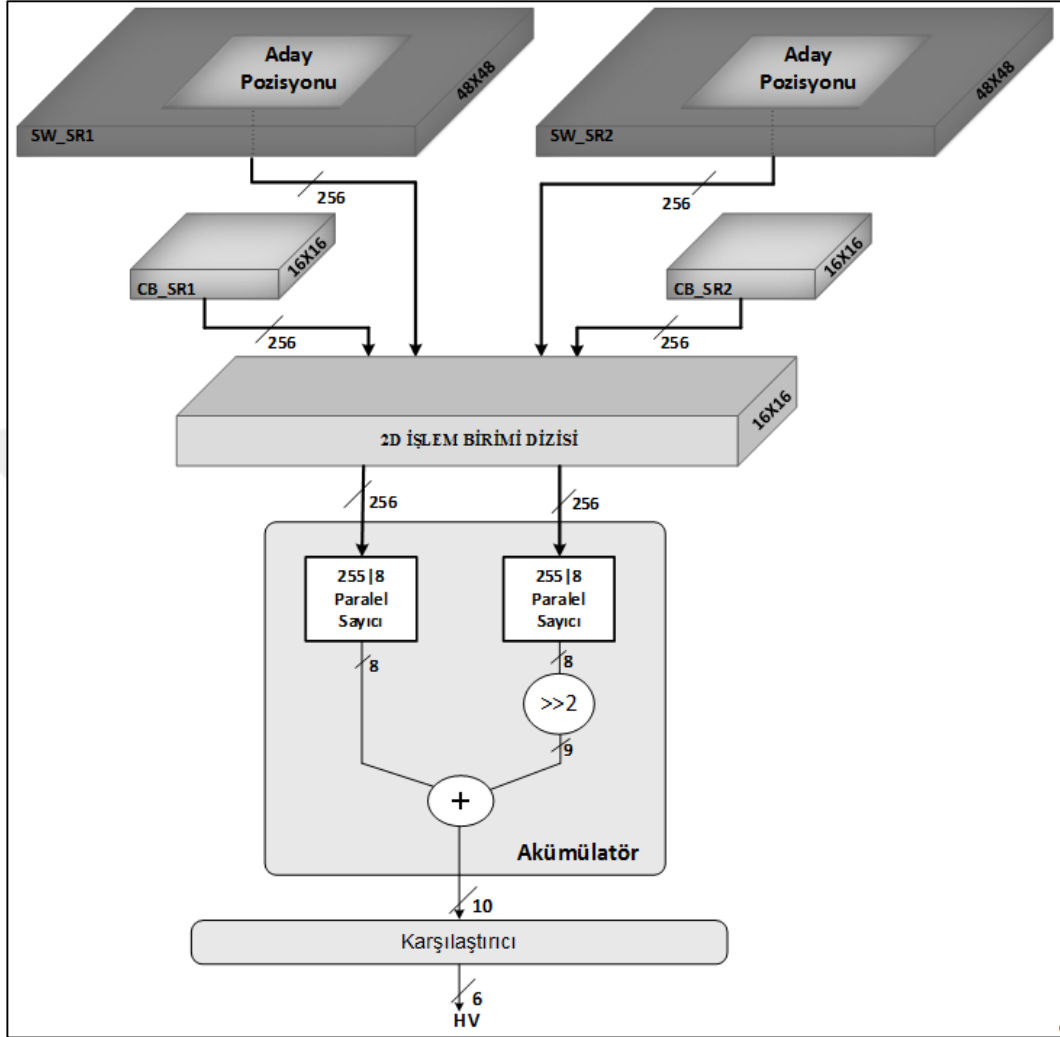


Şekil 3.7. 1DF-2BT yöntemi için ikilileştirme yapıları a) Süzgeçleme aşamasında gerekli olan imge çerçevesi bölgesi, b) Süzgeç donanım mimarisi, c) İkileştirme son aşamada kullanılan yapı [21]

Şekil 3.7’de 1DF-2BT yönteminin ikilileştirme aşamasında kullanılan yapılar gösterilmektedir. Şekil 3.7(a)’da 1×19 boyutundaki 1 boyutlu süzgecin sebep olduğu bellek boyutu artışı gösterilmiştir. Şekil 3.7(b)’de güncel blokta 1 pikselin süzgeçlenmesi ele alınarak süzgeç yapısı gösterilmektedir. Dikkat edilirse, sadece hafızada tutulan bellek boyutu değil aynı zamanda veriyolunu oluşturan donanım elemanlarının miktarı da azalmaktadır. Merkez pikselin filtrelenmiş değeri 4 adet toplayıcı ve 1 adet karşılaştırıcı kullanılarak elde edilmektedir. Şekil 3.3’te süzgeç yapıları gösterilen 1BT, MF-1BT ve C-1BT ile karşılaştırıldığında hem bellek alanı hem de veriyolu açısından önemli ölçüde kazanç sağlandığı açıkça görülmektedir. Son olarak süzgeçlenen pikselin ikili karşılığı Şekil 3.7(c)’de kullanılan yapı ile elde edilmektedir. 1BT ve MF-1BT ile karşılaştırıldığında yaklaşık 2 katı donanım elemanına sahip olduğu görülse de C-1BT’ye kıyasla artış ihmal edilebilecek derecede küçüktür.

İkilileştirme aşamasından sonra ikili imgelerin kullanıldığı uyumlama aşamasının özgün donanım mimarisi Şekil 3.8’de gösterilmektedir. Genel olarak uyumlama donanım mimarisi [34]’te önerildiği gibi 2-boyutlu çekirdek ile süzgeçleme yapan

yöntemlerin mimarilerine benzese de temel farklılık işlem birimi dizisi ile karşılaştırıcı blokları arasındaki veriyolundan kaynaklanmaktadır.



Şekil 3.8. 1DF-2BT yönteminin uyumlama donanım mimarisi

3.4. SGC-2BT HK Yaklaşımı İçin Önerilen Donanım Mimarisi

Gray kodlama temelli hareket kestirimi yöntemlerinin, sahip oldukları düşük ikilileştirme maliyeti sayesinde donanım gerçekleştirilmesi açısından avantajlı olduğu [30]'da sunulan çalışmada ortaya çıkarılmıştır. [30]'daki çalışmada seçimli Gray kodlama temelli 1-bit derinliğinde hareket kestirimi yöntemi önerilmiştir. Bu yöntem basit ikilileştirme işlemi gerçekleştirdiği için düşük işlem yüküne sahip hareket kestirimi yöntemleri içerisinde oldukça başarılıdır. [39]'de sunulan çalışmada ise [30]'daki yöntemin başarımını arttırmak için 2-bit derinliğine sahip [30]'daki gibi seçimli Gray kodlama temelli çalışan özgün bir hareket kestirimi yöntemi

geliştirilmiştir. Bu yöntem SGC-2BT olarak isimlendirilmektedir. Tez kapsamında, SGC-2BT yöntemi için ikilileştirme ve uyumlama aşamasını içeren donanım mimarisi geliştirilmiştir.

g5	g6	g7	g5	g6	g7	g5	g6	g5	g6	g5	g6	g5	g6	g5	g6	g7	g7	g7	g7	g7	g7	g7	g7
g6	g7	g5	g6	g7	g5	g6	g7	g6	g5	g6	g5	g6	g5	g6	g5	g7	g7	g7	g7	g7	g7	g7	g7
g7	g5	g6	g7	g5	g6	g7	g5	g5	g6	g5	g6	g5	g6	g5	g6	g7	g7	g7	g7	g7	g7	g7	g7
g5	g6	g7	g5	g6	g7	g5	g6	g6	g5	g6	g5	g6	g5	g6	g5	g7	g7	g7	g7	g7	g7	g7	g7
g6	g7	g5	g6	g7	g5	g6	g7	g5	g6	g5	g6	g5	g6	g5	g6	g7	g7	g7	g7	g7	g7	g7	g7
g7	g5	g6	g7	g5	g6	g7	g5	g6	g5	g6	g5	g6	g5	g6	g5	g7	g7	g7	g7	g7	g7	g7	g7
g5	g6	g7	g5	g6	g7	g5	g6	g5	g6	g5	g6	g5	g6	g5	g6	g7	g7	g7	g7	g7	g7	g7	g7
g6	g7	g5	g6	g7	g5	g6	g7	g6	g5	g6	g5	g6	g5	g6	g5	g7	g7	g7	g7	g7	g7	g7	g7
a)								b)								c)							

Şekil 3.9. 8×8 imge bloğu için bit seçme örüntüsü a) [30]'da gösterilen yöntemin bit seçme yapısı b) SGC-2BT yöntemi için 1.bit uzayı seçme örüntüsü c) SGC-2BT yöntemi için 2.bit uzayı [39]

[30]'daki yöntemde 1-bit düzlemini oluşturmak için Gray kodlanmış piksel değerlerinden oluşan en değerlikli 3 bit düzleminden (g_5, g_6, g_7) bitlerinden sırayla seçme yapılmaktadır. Bu şekilde oluşturulan 1-bitlik görüntü ile NNMP hesabı yapılarak hareket vektörleri tespit edilmektedir. Kullanılan bit-seçme yapısı Şekil 3.9(a)'da gösterilmektedir. [39]'da sunulan SGC-2BT yönteminde ise bu yaklaşıma benzer şekilde seçme yapılmaktadır. 1. Bit uzayı, g_5 ve g_6 bitleri sıra ile seçilerek oluşturulurken 2.bit uzayı ise g_7 bitinin tek başına kullanılması ile oluşturulmaktadır. Bu çalışmada kullanılan bit seçme yapısı 1.bit uzayı için Şekil 3.9(b)'de, 2. bit uzayı için Şekil 3.9(c)'de gösterilmektedir

İkilileştirme işlemi sonrasında, görüntülerin 2BT gösterimleri elde edilmekte ve uyumlama aşamasında oluşturulan bu 2-bit düzlemler kullanılmaktadır. SGC-2BT yaklaşımında kullanılan uyumlama ölçütü standart NNMP'den farklıdır. Bu çalışmada önerilen bu ölçüt $NNMP_{2BT}$ diye isimlendirilmiş olup aşağıdaki Denklem (3.4)'deki gibi hesaplanmaktadır.

$$\begin{aligned} \text{nnmp}_1(m,n) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B_1^t(i,j) \oplus B_1^{t-1}(i+m,j+n) \\ \text{nnmp}_2(m,n) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} B_2^t(i,j) \oplus B_2^{t-1}(i+m,j+n) \end{aligned} \quad (3.4)$$

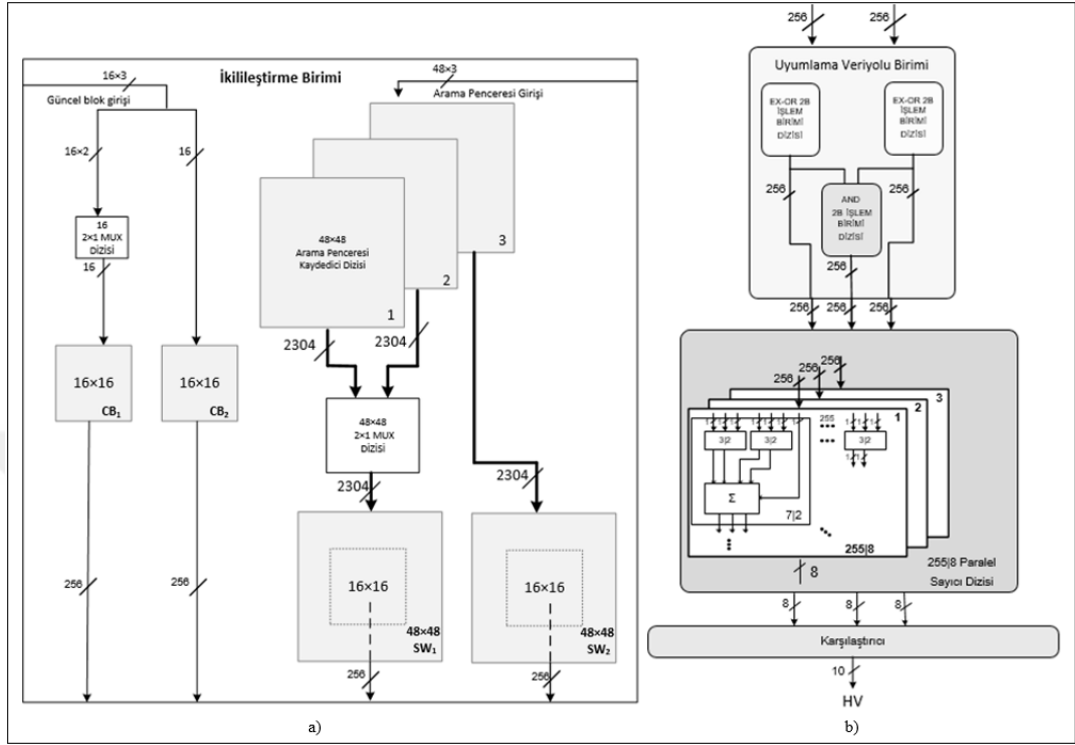
$$\text{nnmp}_3(m,n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \{B_1^t(i,j) \oplus B_1^{t-1}(i+m,j+n)\} \odot \{B_2^t(i,j) \oplus B_2^{t-1}(i+m,j+n)\}$$

$$\text{NNMP}_{2\text{BT}}(m,n) = \text{nnmp}_1(m,n) + \text{nnmp}_2(m,n) + \text{nnmp}_3(m,n)$$

Denklem (3.4)'de (i, j) blok içinde ilgili pikselin konumunu, (m, n) aday hareket vektörünü, N blok boyutunu, B_1^t güncel bloğu oluşturan 1.bit düzlemini, B_1^{t-1} arama penceresini oluşturan 1.bit düzlemini, B_2^t güncel bloğu oluşturan 2.bit düzlemini, B_2^{t-1} arama penceresini oluşturan 2.bit düzlemini temsil etmektedir. $\text{NNMP}_{2\text{BT}}$ 'yi kullanmanın amacı geleneksel NNMP'nin dinamik aralığını geliştirmektir. Önerilen bu ölçütün bir kısmı standart NNMP'ye benzer şekilde (nnmp_1 ve nnmp_2) hesaplanmaktadır. Bu kısımlara ilaveten, nnmp_3 kısmı, her bir bit düzlemi için geçerli blok ve arama penceresindeki ortak farklı noktaların etkisini arttırmak için hesaplanır. Bu nedenle, bu üç değer toplanarak $\text{NNMP}_{2\text{BT}}$ hesaplanmakta ve önerilen bu yeni uyumlama ölçütü ile standart NNMP'ye kıyasla daha yüksek dinamik aralık elde edilmektedir.

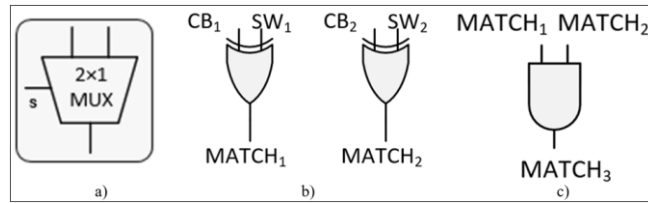
SGC-2BT yönteminde, uyumlama aşamasında kullanılmak üzere ikilileştirme aşaması sonucunda 2 bit düzlemi elde edilir. Şekil 3.10'da geliştirilen donanım mimarisi görülmektedir. Şekil 3.10(a)'da yöntemin ikilileştirme biriminin mimarisi, Şekil 3.10(b)'de uyumlama biriminin mimarisi görülmektedir. Şekil 3.10(a)'da görüldüğü gibi, arama penceresi ve güncel blok sırasıyla 3 ve 2 bit düzleminden oluşmuştur. Bu mimaride, arama penceresi ve güncel blok verileri, 3 ve 4 yönlü kaydırma kabiliyeti olan yaz-bozlardan oluşan kaydırmalı kaydedici dizilerinde saklanmaktadır. 48×48 boyutundaki arama penceresinin ilk iki bit düzlemini tek bit düzlemine düşürmek için

48×48 boyutunda 2×1 bilgi seçici dizisi kullanılmıştır. Kullanılan 2×1 bilgi seçici Şekil 3.11(a)'da gösterilmiştir.



Şekil 3.10. SGC-2BT için geliştirilen donanım mimarileri a) İkileştirme birimi b) Uyumlama birimi [39]

Yöntemde uygulanan seçme örüntüsüne bağlı olarak, 2×1 bilgi seçici girişindeki Gray kodlanmış 2 bittten birini seçmektedir. Böylelikle arama penceresinin 1. bit düzlemi oluşturulur. Arama penceresinin 2. bit düzlemi ise direkt olarak kaydırmalı kaydedici dizilerinin 3. bit düzleminden oluşturulur.



Şekil 3.11. SGC-2BT için veriyolu bileşenleri a) 2×1 bilgi seçici b) EXOR işlem elemanı c) Ve (AND) işlem elemanı [39]

Güncel blok verileri ise kaydedici dizilerinde saklanmadan önce ikiye ayrılır. İlk iki bit düzlemi verilerinden oluşan kısmı 16 tane 2×1 bilgi seçiciden geçirilerek seçme

örüntüsüne bağlı olarak kaydedici dizilerine yüklenirken 3. bit düzlemi verilerinden oluşan diğer kısım direkt aktarılır. İkileştirme aşaması sonucunda, seçme örüntüsüne bağlı olarak hem güncel blok hem de arama penceresi için iki bit düzlemi elde edilir ve bu düzlemler uyumlama aşamasında kullanılır.

İkileştirme aşamasından sonra, Denklem (3.4)'te verilen $NNMP_{2BT}$ uyumlama kriterine göre uyumlama aşaması gerçekleştirilmiştir. İlk olarak 48×48 arama penceresinin merkezindeki 16×16 boyutundaki referans blok ve güncel bloktan oluşan 256 bitlik veri uyumlama birimine giriş olarak uygulanır. Şekil 3.11(b)'de görüldüğü gibi, bu birimde 3 adet işlem birimi dizisi bulunur. Şekil 3.11(b)'de girişindeki 1 bitlik güncel blok ve referans blok verisini karşılaştıran EXOR kapısı gösterilmiştir. Şekil 3.11(c)'de VE (AND) kapısı göstermektedir. Bu işlem elemanı, 16×16 boyutundaki EXOR işlem elemanı dizilerinin çıkışındaki 256 bitlik veriler arasında mantıksal ve işlemini gerçekleştirir. Böylece uyumlama veriyolu biriminin çıkışında üç adet 256 bitlik veri elde edilir ve 3 paralel sayıcı dizisine giriş olarak verilir. Paralel sayıcı dizisi [30]'da sunulan dizi ile aynıdır, bu mimaride 3 adet kullanılmıştır. 3 NNMP hesaplandığında, karşılaştırmacı bloğunda toplanır ve en küçük $NNMP_{2BT}$ değerini veren referans bloğun konumu hareket vektörü olarak belirlenir.

4. BULGULAR VE TARTIŞMA

Bu bölümde tez kapsamında gerçekleştirilen donanım mimarileri iki alt başlık altında sonuçları karşılaştırılarak ayrıntılarıyla incelenmiştir. İlk olarak 1BT, MF-1BT, LBP-1BT ve LBP-2BT yaklaşımları için geliştirilen donanım mimarilerinin HK performansı, donanım maliyeti, güç ve enerji tüketimi sonuçları sırasıyla Tablo 4.1, Tablo 4.2 ve Tablo 4.3'te gösterilmiştir. İkinci kısımda ise 1DF-2BT ve SGC-2BT yaklaşımları için geliştirilen donanım mimarilerinin HK performansı incelemesi Tablo 4.4'te verilmiştir. Tablo 4.5'te 1DF-2BT yöntemi için donanım kaynak kullanımı incelemesi sunulmuştur. 1DF-2BT yönteminin güç ve enerji tüketimi sonuçları ise Tablo 4.6'da verilmiştir. SGC-2BT yaklaşımı için geliştirilen donanım mimarisinin donanım maliyeti ile güç ve enerji tüketimi sonuçları sırasıyla Tablo 4.7 ve Tablo 4.8'de gösterilmiştir.

4.1. 1BT, MF-1BT, LBP-1BT ve LBP-2BT yaklaşımları için geliştirilen donanım mimarilerinin incelenmesi

Literatürdeki birçok düşük bit gösterimi temelli HK yönteminin, HK performansı incelenirken açık çevrim yaklaşımı kullanılmıştır. Bu çalışmada da açık çevrim yaklaşımı uygulanarak, o anki imge çerçevesi kendisinden önceki çerçevelerden tahmin edildikten sonra tahmini ve orijinal çerçevelerin benzerliği PSNR değeri baz alınarak hareket kestirimi performansı açısından incelenmiştir. Tablo 4.1'de 6 farklı video dizisi için literatürdeki düşük karmaşıklığa sahip birkaç HK yöntemi için PSNR değerleri verilmiştir. Blok boyutu ve arama penceresi için sırasıyla 16 ve [-16,16] seçilmiştir. Doğru bir karşılaştırma yapılabilmesi adına geliştirilen tüm HK donanımları 28 nm (Xilinx Zynq 7000 - paket kodu: xc7vx690tffg1761-2) FPGA kullanılarak gerçekleştirilmiştir.

HK yöntemlerini sadece HK başarımı açısından değil aynı zamanda bu başarımı sağlarken harcadığı kaynak, güç ve enerji tüketimi açısından da incelemek gerekmektedir. Tablo 4.2'de önceki bölümlerde süzgeçleme yapıları ayrıntılı olarak

anlatılan I-1BT, MF-1BT, LBP-1BT ve LBP-2BT HK yöntemlerinin donanım maliyetleri verilmiştir. İkileştirme ve uyumlama maliyetlerinin ayrı ayrı verildiği bu tablodan anlaşılacağı üzere, ikileştirme maliyeti uyumlamaya göre çok daha fazladır. Bu da HK donanımı tasarlanırken ikileştirme aşamasını ihmal etmenin doğru bir kabul olmadığını açıkça göstermektedir. Ayrıca elde edilen sonuçlar hareket kestiriminin ikileştirme ve uyumlama aşaması bir bütün olarak ele alındığında, donanım maliyeti açısından MF-1BT'nin iddia edildiği gibi daha düşük işlem karmaşıklığı ve güç tüketimi sunmadığını göstermektedir. Şekil 4.1'den de anlaşılacağı üzere, MF-1BT için kullanılan çekirdek boyutu 1BT'nin çekirdeğinden büyük olduğundan ilave pikseller ile birlikte saklanması gereken blok boyutu da daha büyüktür. Bu da işgal edilen yonga üzeri bellek alanını arttırmaktadır ve iddia edilenin aksine daha fazla kaynak kullanımı ve daha fazla güç tüketimine sebep olmaktadır.

Tablo 4.1. 1BT, MF-1BT, LBP-1BT ve LBP-2BT yaklaşımları için HK performansları

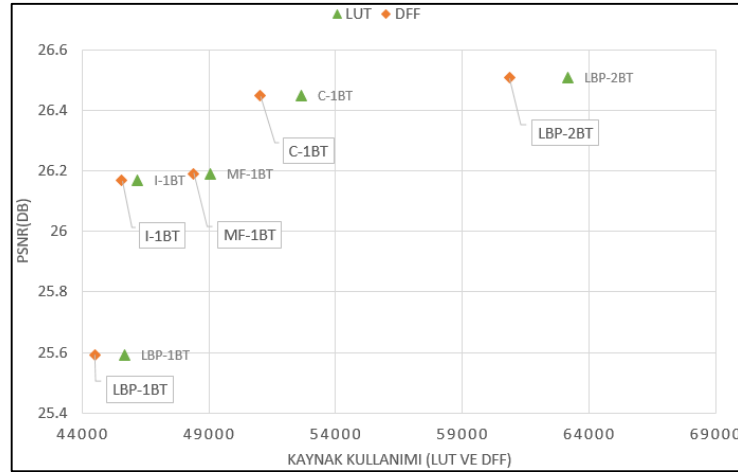
HK Yöntemi	Video Dizileri (Çerçeve Boyutu, Dizi Uzunluğu)						
	Football (352×240) 125 çerçeve	Foreman (352×288) 300 çerçeve	Flowergarden (352×240) 115 çerçeve	Coastguard (352×288) (300 çerçeve)	Tennis (352×240) 150 çerçeve	Mobile (352×240) 300 çerçeve	6 video dizisinin ortalaması
SAD (8-bit derinlik)	22,88	32,09	23,79	30,48	29,45	23,94	27,10
1BT [19]	21,83	30,32	23,31	29,83	28,11	23,61	26,17
MF-1BT [20]	21,81	30,38	23,26	29,88	28,18	23,63	26,19
2BT [40]	22,06	30,70	23,43	29,94	28,46	23,66	26,38
C-1BT [22]	22,10	30,86	23,38	29,98	28,71	23,69	26,45
I-1BT[36]	21,79	30,57	23,35	29,81	28,10	23,61	26,20
LBP [27]	21,40	29,20	23,07	29,72	27,43	23,43	25,59
LBP-1BT [28]	21,40	29,33	23,08	29,73	27,45	22,59	25,59
LBP-2BT[29]	22,18	30,80	23,52	30,11	28,73	23,72	26,51

Tablo 4.2. 1BT, MF-1BT, C-1BT, LBP-1BT ve LBP-2BT yaklaşımları için donanım kaynak kullanımı performansı

	HK Yöntemi	I-1BT [36]	MF-1BT [20]	C-1BT [22]	LBP-1BT [28]	LBP-2BT [37]
İkilileştirme	Çekirdek boyutu	17×17	19×19	19×19	17×17	25×25
	# saat çevrimi	896	896	896	896	896
	# LUT	42140	45006	45143	41628	55089
	# DFF	42636	45468	45537	41565	55019
Uyumlama	# saat çevrimi	1089	1089	1089	1089	1089
	# LUT	4047	4036	7495	4030	8070
	# DFF	2928	2928	5487	2928	5856
Toplam	# saat çevrimi	1985	1985	1985	1985	1985
	# LUT	46187	49042	52638	45658	63159
	# DFF	45564	48396	51024	44493	60875

Tablo 4.1’de LBP-1BT yönteminin HK performansı incelendiğinde 1BT temelli yöntemlere göre bir miktar düşük olduğu görülmektedir. Ancak Tablo 4.2’de LBP-1BT yönteminin ikilileştirme donanım maliyeti incelendiğinde diğer yöntemlere göre daha az kaynak kullanımına sahip olduğuna ve Tablo 4.3’te görüldüğü üzere daha az güç tükettiğine dikkat edilmelidir.

HK performansı ve donanım kaynak kullanımı sırası ile Tablo 4.1. ve Tablo 4.2’de verilmiş olan 5 farklı HK yaklaşımı için kaynak kullanımı ve PSNR değerleri arasındaki ilişki Şekil 4.1’de gösterilmiştir. Görüldüğü üzere, genel olarak daha iyi bir HK performansı sağlayabilen yaklaşımlarda kaynak kullanımı da artış göstermektedir. Ancak daha önceki bölümlerde de değinildiği üzere özellikle ikilileştirmeden kaynaklanan donanım maliyetleri tüm mimarinin kaynak kullanımına etki etmektedir. Örneğin, ikilileştirmede kullandığı daha büyük boyutlu çekirdek sebebiyle MF-1BT, I-1BT yaklaşımı ile benzer HK performansı sağlamasına rağmen daha çok kaynak kullanmaktadır. LBP-2BT çekirdek boyutu ve uyumlama aşamasında 2-bit düzlemi kullanması sebebiyle diğerlerine göre çok daha fazla kaynak kullanmaktadır. Bunun yanında sunduğu HK performansı da diğerlerinden yüksektir. Fazla kaynak kullanımının önemli bir kısıt oluşturmadığı daha iyi HK performansı istenen mimarilerde Şekil 4.1’de sunulan ilişkiye göre istenilen yaklaşım tercih edilebilir.



Şekil 4.1. Farklı HK yaklaşımları için PSNR değerleri ile kaynak kullanımı arasındaki ilişki

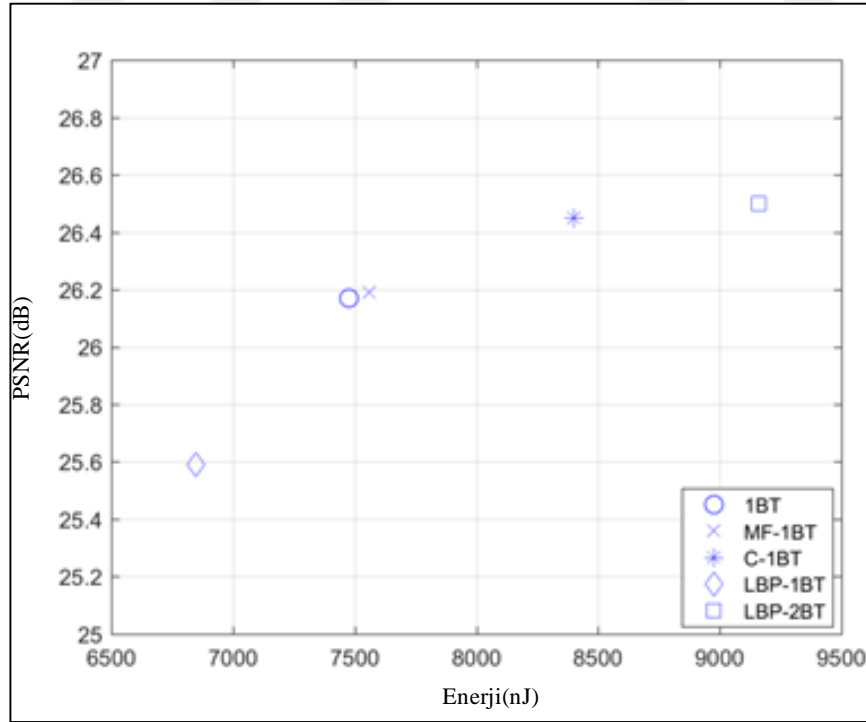
Elde taşınabilir mobil cihazlar gibi tüketici elektroniğindeki birçok cihaz düşünüldüğünde güç tüketimi çok önemli bir kriterdir. Bu çalışmada güç tüketimini

analiz edebilmek amacıyla Tablo 4.3'te verilen her bir yöntem için 9 bloğa göre güç analizleri gerçekleştirilmiştir. 3 farklı hareket karakteristiğine sahip hareket vektörü sonucu veren bloklar tercih edilerek daha doğru bir ortalama güç değerinin elde edilmesi istenmiştir. Saat işareti periyodunun 20 ns seçildiği ortalama güç analizi sonuçları ve enerji değerleri Tablo 4.3'te verilmiştir.

Tablo 4.3. 1BT, MF-1BT, LBP-1BT ve LBP-2BT yaklaşımları için güç ve enerji tüketimi sonuçları [37]

HK Yöntemi	Sinyal Gücü (mW)	Lojik Güç (mW)	Zaman (µs)	Enerji (nJ)
MF-1BT [20]	101	79	42	7560
I-1BT [36]	103	75	42	7476
C-1BT [22]	114	86	42	8400
LBP-1BT[28]	90	73	42	6846
LBP-2BT[37]	120	96	42,4	9158,4

Tablo 4.2 ve Tablo 4.3'te sunulan değerlere bakıldığında LBP-2BT ile hareket kestirimi yaklaşımının hem kapladığı alan hem de güç tüketimi diğer yöntemlere göre oldukça yüksektir. Daha büyük çekirdek boyutu ve uyumlama aşamasında 2 bit düzlemi kullanılıyor olmasından dolayı donanım maliyeti artmıştır.



Şekil 4.2. Farklı HK yaklaşımları için PSNR değerleri ve enerji tüketimi arasındaki ilişki [37]

HK yöntemini değerlendirirken sadece donanım maliyeti ve güç tüketimini değil sağladığı HK doğruluğunu da dikkate almak gerekir. Bu nedenle farklı HK yöntemleri için Tablo 4.1’de verilen PSNR değerleri ile Tablo 4.3’te verilen 3 farklı hareket karakteristiğine sahip 9 blok için ortalama enerji tüketimi değerleri arasındaki ilişki Şekil 4.2’de gösterilmiştir. Görüldüğü üzere LBP-2BT iyi bir enerji-PSNR dengesi sunmaktadır.

4.2. 1DF-2BT ve SGC-2BT yaklaşımları için geliştirilen donanım mimarilerinin incelenmesi

Tablo 4.1’de olduğu gibi, Tablo 4.4’te de 1DF-2BT ve SGC-2BT yaklaşımları için HK performansı incelenirken açık çevrim yaklaşımı kullanılmıştır. 6 farklı video dizisi için literatürdeki düşük karmaşıklığa sahip birkaç HK yöntemi için PSNR değerleri gösterilmiştir. Blok boyutu ve arama penceresi için sırasıyla 16 ve [-16, 16] seçilmiştir.

Tablo 4.4. 1DF-2BT ve SGC-2BT yaklaşımları için HK performansı incelemesi (dB) [39]

HK Yöntemleri	Video Dizileri (Çerçeve Boyutu, Dizi uzunluğu)						6 video dizisinin ortalaması
	Football (352×240) (125 çer.)	Foreman (352×288) (300 çer.)	Flowergarden (352×240) (115 çer.)	Coastguard (352×288) (300 çer.)	Tennis (352×240) (150 çer.)	Mobile (352×240) (300 çer.)	
SAD (8-bit derinlik)	22,88	32,09	23,79	30,48	29,45	23,94	27,11
1BT [19]	21,83	30,32	23,31	29,83	28,11	23,61	26,17
MF-1BT [20]	21,81	30,38	23,26	29,88	28,18	23,63	26,19
2BT [40]	22,06	30,70	23,43	29,94	28,46	23,66	26,38
C-1BT [22]	22,10	30,86	23,38	29,98	28,71	23,69	26,45
Erken Sonlandırılmalı C-1BT [41]	21,85	30,37	23,20	29,99	28,60	23,60	26,33
Uyarlamalı Arama Temelli C-1BT[42]	22,06	30,71	23,44	30,05	28,60	23,67	26,42
T-GCBPM [43] NTB=6	22,38	30,64	23,61	29,14	28,69	23,78	26,33
T-GCBPM [43] NTB=5	22,59	31,32	23,67	30,16	28,78	23,81	26,72
Iodifiye XOR(8-bit derinlik)[44]	22,64	31,91	23,65	30,40	29,25	23,80	26,94
SGC-1BT [30]	22,24	31,03	23,38	29,85	28,69	23,47	26,44
1DF-2BT [21]	22,35	30,86	23,55	29,97	28,30	23,75	26,46
SGC-2BT[39]	22,56	31,35	23,62	30,14	28,92	23,75	26,72

Süzgeçleme temelli HK yaklaşımlarından olan 1BT, MF-1BT, C-1BT, 2BT ile 1DF-2BT HK yaklaşımı karşılaştırıldığında tüm diziler için en iyi ortama PSNR değerini 1DF-2BT yaklaşımın verdiği görülmektedir. Ayrıca 1DF-2BT’nin ikilileştirme donanımı açısından değerlendirildiğinde de en az işlem yüküne sahip yöntemdir. Önceki bölümlerde ayrıntılı açıklanmış olan 1BT, MF-1BT, C-1BT donanım mimarileri ile 1DF-2BT donanım mimarisi kaynak kullanımı performansı açısından

Tablo 4.5'te karşılaştırılmıştır. Sonuçlar ikilileştirme, uyumlama ve HK'nin tamamı şeklinde ayrılarak ayrıntılı bir şekilde gösterilmektedir. Dikkat edilirse, önerilen 1DF-2BT yöntemi [30]'da önerilen seçimli Gray kodlama temelli HK yöntemi dışında en düşük ikilileştirme donanım maliyetine sahiptir. 1DF-2BT yönteminin uyumlama maliyeti ise diğer yöntemler ile benzerdir.

Tablo 4.5. 1DF-2BT yöntemi için donanım kaynak kullanımı incelemesi [21]

	HK Yöntemi	1BT [19]	MF-1BT [20]	C-1BT [22]	SGC-1BT [30]	1DF-2BT [21]
İkilileştirme	# saat çevrimi	896	896	896	Yok	896
	Çekirdek boyutu	17x17	19x19	19x19	Yok	1x19
	# LUT	42140	45006	45143	96	545
	# DFF	42636	45468	45537	Yok	1742
Uyumlama	# saat çevrimi	1089	1089	1089	1089	1089
	# LUT	4047	4036	7495	8574	8062
	# DFF	2928	2928	5487	7877	5856
	# uyumlamada kullanılan bitler	1	1	2	3	2
Toplam	# saat çevrimi	1985	1985	1985	1089	1985
	# LUT	46187	49042	52638	8670	8607
	# DFF	45564	48396	51024	7877	7598

Elektronik cihazların kullanımında önemli bir kısıt oluşturan güç tüketimini analiz edebilmek amacıyla karşılaştırılan süzgeçleme temelli ikilileştirme yaklaşımlarını kullanan HK yöntemleri için güç tüketimi analizleri gerçekleştirilmiştir. Daha doğru sonuç alabilmek için 3 farklı hareket karakteristiğinde 9 blok için saat işareti periyodu 20 ns seçilerek güç analizleri gerçekleştirilmiş ve ortalama güç ve enerji değerleri Tablo 4.6'da sunulmuştur. Sonuçlardan anlaşılacağı üzere, 1DF-2BT HK yöntemi en düşük güç tüketimine sahiptir.

Tablo 4.6. 1DF-2BT yöntemi için güç ve enerji tüketimi incelemesi [21]

HK Yöntemi	Sinyal Gücü (mW)	Lojik Güç (mW)	Zaman (µs)	Enerji (nJ)
1BT [19]	107	77	42	7728
MF-1BT [20]	104	82	42	7812
C-1BT [22]	117	87	42	8568
1DF-2BT [21]	13	13	40,2	1045,2

Tablo 4.4'te SGC-2BT yöntemi ile düşük karmaşıklığa sahip bilinen HK yöntemlerinin PSNR sonuçları verilmiştir. Tablodan görüldüğü gibi, SGC-2BT yöntemi, 1DF-2BT [21], 2BT [40], C-1BT [22], erken sonlandırılmalı C-1BT [41], uyarlamalı arama aralığı temelli C-1BT [42] ve T-GCBPM [43] gibi yöntemlerle karşılaştırıldığında uyumlama aşamasında 2-bit kullanılan hareket kestirimi

yöntemlerine göre çok daha iyi başarımlar vermektedir. Buna ek olarak, önerilen yöntem, 3 bitlik düzlem kullanan T-GCBPM [43] ile aynı performansa sahiptir. Ayrıca önerilen yaklaşım, yakın zamanda önerilen seçimli Gray kodlama [30] tabanlı hareket kestirimi yaklaşımı ile karşılaştırıldığında ise ortalama 0.3 dB PSNR kazancı sağlamaktadır. Bu sonuçlar SGC-2BT yönteminin düşük bit derinliğine sahip yöntemlerden çok daha başarılı sonuçlar verdiğini ortaya koymaktadır. [44]'te önerilen yöntem SGC-2BT yönteminden daha iyi HK performansı sunmaktadır. Ancak uyumlama ölçütü olarak modifiye edilmiş bir EXOR yapısı kullanan bu yöntem daha çok işlemsel yüke sahiptir. Çünkü SAD yaklaşımına benzer bir şekilde, uyumlama ölçütünü hesaplarken 8 bit derinliği kullanılmaktadır. Bu da donanım açısından dezavantaj oluşturmaktadır.

SGC-2BT yönteminin donanım mimarisi 28 nm (Xilinx Zynq 7000 – paket kodu: xc7a200tfbg676-2) FPGA kullanılarak gerçekleştirilmiştir. SGC-2BT yöntemi için geliştirilen donanım mimarisinde 12283 LUT ve 8743 DFF kullanılmıştır.

[41]'de sunulan donanımın maksimum frekansının yüksek olmasının sebebi uyarlamalı arama aralığı yaklaşımını kullanmasıdır.

Seçimli Gray kodlama yaklaşımı [30]'da sunulan SGC-1BT yönteminde de kullanılmıştır. Ancak önerilen yöntemde uyumlama süreci 2 bit düzlemi ile gerçekleştirilirken [30]'da 1 bit düzlemi kullanılmıştır. Dolayısıyla önerilen yöntemin daha fazla LUT ve DFF kullanması beklenen bir durumdur.

Tablo 4.7'de geliştirilen donanımın hareket kestirimi performansı diğer düşük karmaşıklığa sahip yöntemlerin donanımları ile ayrıntılı olarak karşılaştırılmıştır.

SGC-2BT donanımında kullanılan tüm kaynaklardan bellek için kullanılan miktar çıkartıldığında geriye kalan LUT sayısı yaklaşık olarak 1500 ve DFF sayısı yaklaşık olarak 2200 olmaktadır. Bu nedenle, [39]'da önerilen çalışma [41]'de sunulan çalışma ile uyumlama aşamasındaki kaynak kullanımı açısından karşılaştırılabilir niteliktedir. Tablo 4.7'den anlaşılacağı üzere, SGC-2BT donanım mimarisinin saniyede işleyebileceği çerçeve sayısı [21, 44]'teki mimarilerden daha fazladır.

Tablo 4.7. SGC-2BT yöntemi için donanım kaynak kullanımı incelemesi [39]

	SGC-2BT [39]	SGC-1BT [30]	Erken Sonlandırılmalı C-1BT [41]	Modifiye XOR [44]	1DF-2BT [21]
İkilileştirme aşaması	Var	Var	Yok	Yok	Var
İkilileştirme bit derinliği	3	3	NA	8	8
Uyumlama bit derinliği	2	1	2	8	2
Yonga üzeri bellek alanı	0	0	9632 bit	8192 bit	0
Alan	12283 LUT 8743 DFF (10737 LUT 7427 DFF bellek için)	8747 LUT 7877 DFF (8086 LUT 7677 DFF bellek için)	5035 LUT 4100 DFF	6157 LUT NA (0 LUT 0 DFF bellek için)	8607 LUT 7598 DFF (8062 LUT 5856 DFF bellek için)
Dinamik güç	171mW @50MHz 222mW @100MHz	111mW @50MHz 137mW @100MHz	NA 202mW @100MHz	190mW @50MHz NA	26,3mW @50MHz 52,22mW @100MHz
Maksimum frekans	309 MHz	327 MHz	265 MHz	293 MHz	209,24 MHz
Performans 720p	78,81 fps	83,41 fps	NA	19,85 fps	29,24 fps
Performans 1080p	35,03 fps	37,07 fps	NA	8,82 fps	13,01 fps
Performans 4K	8,75 fps	9,26 fps	> 30 fps	2,20 fps	3,25 fps
Teknoloji	FPGA 28 nm	FPGA 28 nm	FPGA 65 nm	FPGA 40 nm	FPGA 28 nm
Arama aralığı	[-16 16]	[-16 16]	[-16 16]	[-16 16]	[-16 16]
Arama metodu	Tam arama	Tam arama	Tam arama /erken sonlandırma	Tam arama	Tam arama
Tarama örüntüsü	Spiral tarama	Spiral tarama	NA	Raster arama	Spiral tarama
Desteklenen makroblok boyutu	16x16	16x16	16x16 (4 MB paralel olarak)	4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16	16x16
# referans çerçeve	1	1	1	1	1

Ayrıca önerilen mimari [21, 41, 44]'teki mimarilere göre daha yüksek maksimum frekansa sahiptir. Bununla birlikte, SGC-2BT yönteminin PSNR performansı [44]

dışındaki çalışmalardan daha iyidir. Ancak, [44]'te sunulan çalışmanın bir piksel için 8 bit kullandığı ve bunun donanımsal açıdan işlemsel yük getirdiği göz ardı edilmemelidir. Tablo 4.7'de görüldüğü üzere [44]'te sunulan çalışma daha az donanım maliyetine sahip olmasına ve yaklaşık olarak benzer güç tüketmesine rağmen [39]'da sunulan çalışmadan saniyede işlenen çerçeve sayısı açısından 4 kat daha yavaştır.

[30]'da uyumlama aşaması için sadece bir bit düzlemi kullanıldığından daha az güç tüketmesi doğaldır. [39]'da önerilen donanım [41]'deki donanıma kıyasla daha fazla güç tüketiyor gibi gözükse de bu mimaride ikilileştirme donanımı bulunmamaktadır. [44]'teki donanımın güç tüketimi değerleri [39]'da önerilen donanımdan daha yüksektir. Seçimli Gray kodlama temelli hareket kestirimi yaklaşımları, ikilileştirme aşamasında kullanılan seçme örüntüsünden kaynaklı olarak daha fazla anahtarlama aktivitesi gerektirir. Bu nedenle güç tüketimi daha fazla olabilmektedir. Sağladığı diğer avantajların yanında anahtarlama aktivitesinden kaynaklanan güç tüketimi bir dezavantaj sayılabilir. Tablo 4.8'de SGC-2BT donanım mimarisinin 20ns ve 10ns saat işareti periyodunda, 3 farklı hareket karakteristiğine sahip 9 blok için dinamik güç tüketimi değerleri verilmiştir.

Tablo 4.8. SGC-2BT güç analizi sonuçları [39]

3 farklı hareket karakteristiği	Dinamik Güç Tüketimi (mW)
	20ns/10ns
$5 \geq h_v_x/h_v_{yz} \geq 0$	SGC-2BT [39]
	170/219
	182/242
$10 \geq h_v_x/h_v_{yz} > 5$	160/202
	179/238
	160/200
$ h_v_x/h_v_{yz} > 10$	181/242
	165/211
	164/208
Ortalama	182/243
	171/222

5. SONUÇLAR VE ÖNERİLER

Literatürde bilinen düşük bit gösterimi temelli hareket kestirimi yaklaşımları için ikili imgelerin hazır olduğu kabul edilmiş ve sadece uyumlama aşamasını içeren donanım mimarileri önerilmiştir. Bu tez kapsamında hareket kestirimi yaklaşımlarına ait uyumlama aşamasına ek olarak ikilileştirme aşamasının donanım mimarileri gerçekleştirilmiştir.

Bilinen düşük karmaşıklığa sahip hareket kestirimi yaklaşımları, hareket kestirimi doğruluğu, ikilileştirme ve uyumlama donanım maliyeti, güç ve enerji tüketimi açısından kapsamlı bir şekilde karşılaştırılmıştır. Elde edilen sonuçlardan anlaşılacağı üzere ikilileştirme donanım maliyeti uyumlamaya göre çok daha fazladır. Video kodlayıcının tamamı dikkate alındığında ikilileştirme aşamasının donanım maliyetini ihmal etmenin doğru bir kabul olmadığı ortaya çıkmaktadır.

Önerilen donanım mimarileri arasında doğru bir karşılaştırma yapılabilmesi amacıyla tüm mimariler 28nm (Xilinx Zynq 7000 - paket kodu: xc7vx690tffg1761-2) FPGA teknolojisi için sentezlenmektedir. Önerilen mimarileri kodlamak için Verilog HDL kullanılmaktadır. Önerilen donanım mimarileri, H.264/AVC ve HEVC hareket kestirimi donanım mimarilerine kolaylıkla entegre edilebilir ve sunduğu düşük donanım maliyeti ve güç tüketimi ile mobil cihazlardaki uygulamalar için de uygundur.

KAYNAKLAR

- [1] Aslam M., Celebi A., Efficient Hardware Architecture for Selective Gray Coded Bit Plane based Low Complexity Motion, *Gazi University Journal of Science*, DOI: 10.1109/TCE.2016.7448566.
- [2] Milicevic Z., Bojkovic Z., HEVC vs. H.264/AVC Through Performance and Complexity Comparison, *Proceedings of the 13th International Conference on Applications of Computer Engineering (ACE'14)*, China, September, 2014.
- [3] Sullivan G. J., Jens-Rainer O., Woo-Jin H., Thomas W., Overview of the High Efficiency Video Coding (HEVC) Standard, *IEEE Transactions On Circuits And Systems For Video Technology*, 2012, **22**(12), 1649-1668.
- [4] Richardson I. E., *The H.264 Advanced Video Compression Standard*, 2nd ed., John Wiley & Sons, UK, 2010.
- [5] ITU-T H.263, Video Coding for Low-Bit Rate Communications, *ITU Telecommunication Standardization Sector*, Switzerland, 1998.
- [6] Sze V., Budagavi M., Sullivan G. J., High Efficiency Video Coding (HEVC), *Springer International Publishing*, Switzerland, 2014.
- [7] Malvar H., Hallapuro A., Karczewicz M., Kerofsky L., Low-Complexity Transform and Quantization in H. 264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology.*, 2003, **13**(7), 598-603.
- [8] JVT of ISO/IEC and ITU-T, Recommendation ITU-T H.265 (ISO/IEC 23008-2:2013), *Joint Video Team(JVT) of ITU Telecommunication Standardization Sector (ITU-T) and International Organization for Standardization (ISO)*, Geneva, 2013.
- [9] Lainema J., Bossen F., Han W. J., Min J., Ugur K., Intra Coding of the HEVC Standard, *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, **22**(12), 1792-1801.
- [10] Wang C. N., Yang S. W., Liu C. M., Chiang T., A Hierarchical N-queen Decimation Lattice and Hardware Architecture for Motion Estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, 2004, **14**(4), 429-440.
- [11] Koga T., Linuma K., Hirano A., Lijima Y., Ishiguro T., Motion Compensated Interframe Coding for Video Conferencing, *Proceedings of National Telecommunication Conference*, New Orleans, LA, December 1981.

- [12] Zhu S., Ma K. K., A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation, *IEEE Transactions on Image Processing*, 2000, **9**(2), 287-290.
- [13] Zhu C. C., Lin X., Chau L.P., Hexagon based Search Pattern for Fast Block Motion Estimation, *IEEE Transactions on Circuits and Systems for Video Technology*, 2002, **12**(5), 349-355.
- [14] Lee J., Choi M., Cho Y., Kim J., Cho W. K, Fast H.264/AVC Motion Estimation Algorithm using Adaptive Search Range, *Proceedings of the 12th International Symposium on Integrated Circuits (ISIC '09)*, Singapore, 14-16 December 2009.
- [15] Bierling M., Displacement Estimation by Hierarchical Block Matching, *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, San Jose, CA, USA, October 1998.
- [16] Saha A., Mukherjee J., Sural S., New Pixel-Decimation Patterns for Block Matching in Motion Estimation, *Signal Process-Image Communication.*, 2008, **23**(10), 725-738.
- [17] Li W., Salari E., Successive Elimination Algorithm for Motion Estimation, *IEEE Transactions on Image Processing*, 1995, **4**(1), 105-107.
- [18] Yang L., Yu K., Li J., Li S., An Effective Variable Block-Size Early Termination Algorithm for H.264 Video Coding, *IEEE Transactions on Circuits and Systems for Video Technology*, 2005, **15**(6), 784-788.
- [19] Natarajan B., Bhaskaran V., Konstantinides K., Low-Complexity Block-based Motion Estimation via One-Bit Transforms, *IEEE Transaction on Circuits and Systems for Video Technology*, 1997, **7**(4), 702-706.
- [20] Ertürk S., Multiplication-Free One-Bit Transform for Low-Complexity Block-based Motion Estimation, *IEEE Signal Processing Letters*, 2007, **14**(2), 109-112.
- [21] Taşyapı Çelebi A., Yavuz S., Çelebi A., Urhan O., One-Dimensional Filtering based Two-Bit Transform and Its Efficient Hardware Architecture for Fast Motion Estimation, *IEEE Transaction on Circuits and Systems for Video Technology*, 2017, **63**(4), 377-385.
- [22] Urhan O., Ertürk S., Constrained One-Bit Transform for Low-Complexity Block Motion Estimation, *IEEE Transaction on Circuits and Systems for Video Technology*, 2007, **17**(4), 478-482.
- [23] Ojala T., Pietikainen M., Harwood D., A Comparative Study of Texture Measures with Classification based on Feature Distributions, *Pattern Recognition Systems*, 1996, **29**(1), 51-59.

- [24] Ahonen T., Hadid A., Pietikainen M., Face Recognition with Local Binary Patterns, *Lecture Notes in Computer Science*, 2004, **3021**(1), 469-481.
- [25] Ylioinas J., Hadid A., Hong X., Pietikainen M., Age Estimation using Local Binary Pattern Kernel Density Estimate, *Lecture Notes in Computer Science*, 2013, **8156**(1), 141–150.
- [26] Manivannan S., Wang R., Trucco E., Extended Gaussian-Filtered Local Binary Patterns for Colonoscopy Image Classification, *International Conference on Computer Vision (ICCV-2013)*.
- [27] Kır B., Kurt M., Urhan O., Local Binary Pattern based Fast Digital Image Stabilization, *IEEE Signal Processing Letters*, 2015, **22**(3), 341-345.
- [28] Yavuz S., Taşyapı Çelebi A., Çelebi A., Urhan O., Local Binary Pattern Method and Its Hardware Architecture for Low-Complexity Motion Estimation, *Proceedings of 25th Signal Processing and Communication Applications Conf. (SIU)*, Antalya, Turkey. DOI: 10.1109/SIU.2017.7960443.
- [29] Taşyapı Çelebi A., Two-Bit Transform using Local Binary Pattern Method for Low-Complexity Block Motion Estimation, *Turkish Journal Of Electrical Engineering & Computer Sciences.*, DOI: 10.3906/elk-1704-198.
- [30] Yavuz S., Çelebi A., Aslam M., Urhan, O., Selective Gray-Coded Bit-Plane based Low Complexity Motion Estimation and Its Hardware Architecture, *IEEE Transactions on Consumer Electronics*, 2016, **62**(1), 76-84.
- [31] Akin A., Sayilar G., Hamzaoglu I., High Performance Hardware Architectures for One Bit Transform based Single and Multiple Reference Frame Motion Estimation, *IEEE Transaction on Consumer Electronics*, 2010, **56**(2), 1144–1152.
- [32] Chatterjee S., Chakrabarti I., Low Power VLSI Architectures for One Bit Transformation based Fast Motion Estimation, *IEEE Transaction on Consumer Electronics*, 2011, **56**(4) 2652–2660.
- [33] Çelebi A., Lee H. J., Erturk S., Bit Plane Matching based Variable Block Size Motion Estimation Method and Its Hardware Architecture, *IEEE Transaction on Consumer Electronics*, 2010, **56**(3), 1625-1633.
- [34] Çelebi A., Urhan O., High Performance Hardware Architecture for Constrained One-Bit Transform based Motion Estimation, *19th European Signal Processing Conference (EUSIPCO 2011)*, Barcelona, Spain, 29 August-2 September 2011.
- [35] Çelebi A., Urhan O., Hamzaoğlu I., Ertürk S., Efficient Hardware Implementations of Low Bit Depth Motion Estimation Algorithms, *IEEE Signal Processing Letters*, 2009, **16**(6), 513-516.

- [36] Yavuz S., Taşyapı Çelebi A., Çelebi A., Urhan O., Integer 1-bit Transform Method and Its Hardware Architecture for Low-Complexity Block-based Motion Estimation, *Proc. of 25th Signal Processing and Communication Applications Conference. (SIU)*, DOI: 10.1109/SIU.2017.7960295.
- [37] Yavuz S., Çelebi A., Taşyapı Çelebi A., Urhan O., Hardware Implementation of Local Binary Pattern based Two-Bit Transform Motion Estimation, *World Academy of Science, Engineering and Technology International Journal of Electrical and Computer Engineering*, 2018, **12**(1), 19-23.
- [38] Chew L. W., Ang L., Seng K. P., Low-Complexity Line-based Motion Estimation Algorithm, *Computational Intelligence and Security, International Conference*, China, December 2007.
- [39] Taşyapı Çelebi A., Yavuz S., Çelebi A., Urhan O., Selective Gray-Coded Bit-Plane based Two-Bit Transform and Its Efficient Hardware Architecture for Low-Complexity Motion Estimation, *IEEE Transaction on Consumer Electronics*, 2018, **64**(3), 259 - 266.
- [40] Ertürk A., Ertürk S., Two-Bit Transform for Binary Block Motion Estimation, *IEEE Transaction Circuit System Video Technology*, 2005, **15**(7), 938- 946.
- [41] Akın A., Hamzaoğlu I., A High Performance Hardware for Early Terminated C-1BT based Motion Estimation, *Design and Technology of Integrated Systems in Nanoscale Era (DTIS) Conference*, Istanbul, Turkey, 12-14 April 2016.
- [42] Urhan O., Constrained One-Bit Transform based Fast Block Motion Estimation using Adaptive Search Range, *IEEE Transaction on Consumer Electronics*, 2010, **56**(3), 1868–1871.
- [43] Çelebi A., Akbulut O., Urhan O., Ertürk S., Truncated Gray-Coded Bit-Plane Matching based Motion Estimation and Its Hardware Architecture, *IEEE Transaction on Consumer Electronics*, 2009, **55**(3), 1530-1536.
- [44] AlQaralleh E.A., Abu-Sharkh O.M.F., Low-Complexity Motion Estimation Design using Modified XOR Function, *Multimed Tools Applications*, 2016, **75**(24), 16809-16834.

KİŞİSEL YAYIN VE ESERLER

- [1] **Yavuz S.**, Çelebi A., Taşyapı Çelebi A., Urhan O., Hardware Implementation of Local Binary Pattern based Two-Bit Transform Motion Estimation, *World Academy of Science, Engineering and Technology International Journal of Electrical and Computer Engineering*, 2018, **12**(1), 259-266.
- [2] Taşyapı Çelebi A., **Yavuz S.**, Çelebi A., Urhan O., Selective Gray-Coded Bit-Plane based Two-Bit Transform and Its Efficient Hardware Architecture for Low-Complexity Motion Estimation, *IEEE Transaction on Consumer Electronics*, 2018, **64**(3), 259 - 266.
- [3] **Yavuz S.**, Taşyapı Çelebi A., Çelebi A., Urhan O., Integer 1-bit Transform Method and Its Hardware Architecture for Low-Complexity Block-based Motion Estimation, *Proceedings. of 25th Signal Processing and Communication Applications Conf. (SIU)*, DOI: 10.1109/SIU.2017.7960295.
- [4] Taşyapı Çelebi A., **Yavuz S.**, Çelebi A., Urhan O., One-Dimensional Filtering based Two-Bit Transform and Its Efficient Hardware Architecture for Fast Motion Estimation, *IEEE Transaction Consumer Electronics*, 2017, **63**(4), 377-385.
- [5] **Yavuz S.**, Taşyapı Çelebi A., Çelebi A., Urhan O., Local Binary Pattern Method and Its Hardware Architecture for Low-Complexity Motion Estimation, *Proceedings. of 25th Signal Processing and Communication Applications Conf. (SIU)*, DOI: 10.1109/SIU.2017.7960443.
- [6] **Yavuz S.**, Çelebi A., Aslam M., Urhan O., Selective Gray-Coded Bit-Plane based Low Complexity Motion Estimation and Its Hardware Architecture, *IEEE Transactions on Consumer Electronics*, 2016, **62**(1), 76-84.

ÖZGEÇMİŞ

Seda ÇETİN, Şişli İstanbul’da doğdu. Lise öğrenimini Burak Bora Anadolu Lisesi’nde tamamladı. Kocaeli Üniversitesi Elektronik Haberleşme Mühendisliği alanında lisans derecesini 2016 yılında aldı. 2016-2018 yılları arasında Kocaeli Üniversitesi Tümlleşik Sistemler Araştırma Laboratuarında yüksek lisans bursiyeri olarak çalışmıştır. 2018 yılından bu yana TÜBİTAK UEKAE’de çalışmaktadır. Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü’nde yüksek lisans eğitimine 2016 yılında başlamıştır. Mevcut araştırma ilgi alanları arasında video kodlama ve FPGA uygulamaları bulunmaktadır.

